NO. 49 JUNE 1982

U.S./Canada Edition: $2.50
International Edition: $2.95
United Kingdom Edition: £1.80

# MICRO™

## THE 6502/6809 JOURNAL

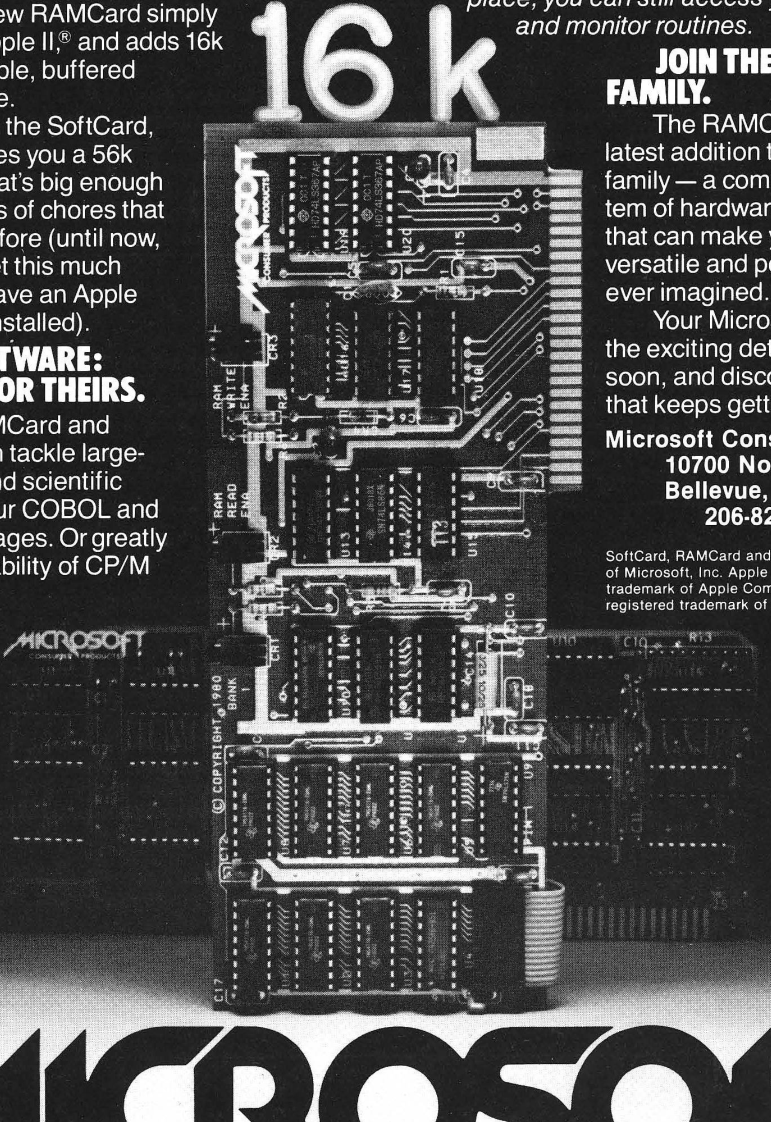
## Applications Feature

Low-Cost Digitizer for Apple

Face Synthesizer for PET

Microcomputer Interfacing: FORTH *vs.* BASIC

# WHY THE MICROSOFT RAMCARD™ MAKES OUR SOFTCARD™ AN EVEN BETTER IDEA.



Memory — you never seem to have quite enough of it.

But if you're one of the thousands of Apple owners using the SoftCard, there's an economical new way to expand your memory dramatically.

## 16K ON A PLUG-IN CARD.

Microsoft's new RAMCard simply plugs into your Apple II,® and adds 16k bytes of dependable, buffered read/write storage.

Together with the SoftCard, the RAMCard gives you a 56k CP/M® system that's big enough to take on all kinds of chores that would never fit before (until now, the only way to get this much memory was to have an Apple Language Card installed).

## GREAT SOFTWARE: YOURS, OURS, OR THEIRS.

With the RAMCard and SoftCard, you can tackle large-scale business and scientific computing with our COBOL and FORTRAN languages. Or greatly increase the capability of CP/M applications like the Peachtree Software accounting systems. VisiCalc™ and other Apple software packages can take advantage of RAMCard too.

And RAMCard gives you the extra capacity to develop advanced programs of your own, using the SoftCard and CP/M. *Even with the RAMCard in place, you can still access your ROM BASIC and monitor routines.*

## JOIN THE SOFTCARD FAMILY.

The RAMCard is just the latest addition to the SoftCard family — a comprehensive system of hardware and software that can make your Apple more versatile and powerful than you ever imagined.

Your Microsoft dealer has all the exciting details. Visit him soon, and discover a great idea that keeps getting better.

**Microsoft Consumer Products**
**10700 Northup Way**
**Bellevue, WA 98004**
**206-828-8080**

SoftCard, RAMCard and Microsoft are trademarks of Microsoft, Inc. Apple II is a registered trademark of Apple Computer Inc. Z-80 is a registered trademark of Zilog, Inc. CP/M is a registered trademark of Digital Research Inc. VisiCalc is a registered trademark of Personal Software, Inc. Microsoft Consumer Products is a division of Microsoft, Inc.

MICROSOFT™

# 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 ¼'' 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 SS50 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 ¼" 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 MICROWARES'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 SS50 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.

**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.

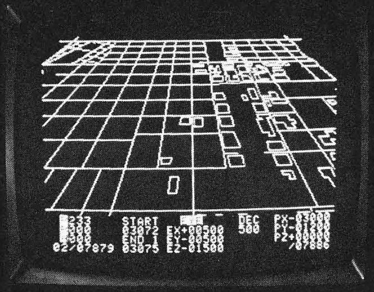
FLEX AND Uniflex are trademarks of Technical Systems Consultants Inc. OS-9 is a trademark of Microware Inc. See their ads for other **GIMIX** compatible software

**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**

# from SubLOGIC... quality software for your Apple II.



```
 233     START   ▀▀▀▀ -  DEC  PX-03000
 500     03072 EX+00500  500  P1-01500
 500     END 1 EY-00500       P2+00000
02/07879 03075 EZ-01500        /07000
```

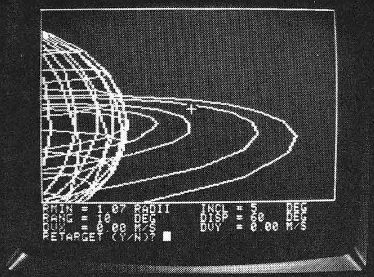
A2-3D1

**GRAPHICS FAMILY** – State-of-the-art 3D graphics. Define 2D or 3D wire-frame objects in any size and orientation, from any perspective. Variable field of view, color or hi-res (280 x 192) line generation, object instance nesting, and independent object manipulation. **Graphics Editor** lets you add 3D text to your scene, superimpose 2D text labels in upper- or lower-case, and record your entire presentation for playback. BASIC interface included to aid development of your own control programs. DOS 3.3, 48K, 3 disks, **$119.85**.

---

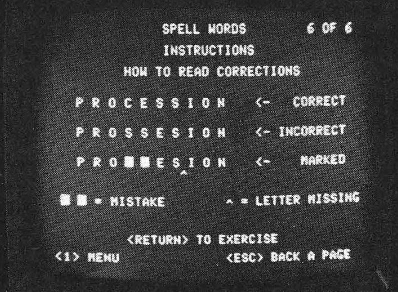
A2-3DA

**SATURN NAVIGATOR** – A hi-res 3D adventure simulation of a space flight from earth to Saturn. Maneuver your ship into orbit around the ringed planet, rendezvous with the Saturn space station. Available as a complete package or as an adjunct to the A2-3D1 graphics package. Applesoft 48K disk. A2-3DA **$34.95**. A2-3D/A (requires A2-3D1) **$24.95**.



```
RMIN = 1.07 RADII     INCL = 5   DEG
RANG = 10   DEG       DISP = 60  DEG
DVX  = 0.00 M/S       DVY  = 0.00 M/S
RETARGET (Y/N)? ■
```

---



```
          SPELL WORDS         6 OF 6
          INSTRUCTIONS
       HOW TO READ CORRECTIONS

  P R O C E S S I O N   <-   CORRECT

  P R O S S E S I O N   <-  INCORRECT

  P R O ■ ■ E S I O N   <-    MARKED
          ^
■ ■ = MISTAKE         ^ = LETTER MISSING

          <RETURN> TO EXERCISE
<1> MENU              <ESC> BACK A PAGE
```

A2-ED1

**WHOLE BRAIN SPELLING**

A colorful program that helps develop internal visualization skills for improving spelling. Available in the following word-list categories:

| | |
|---|---|
| MAIN (8 – adult) | SECRETARIAL |
| SCIENTIFIC | FAIRY TALE |
| MEDICAL | A CHILD'S GARDEN OF WORDS (preschool – 8) |

DOS 3.3, Applesoft 48K disk, **$34.95**.

---

## *See them today at your dealer . . .*

or for direct orders add $1.50 and specify UPS or first class mail. For the A2-3D1 Graphics Family add $3.00 for UPS or $6.50 for first class delivery. Illinois residents add 5% sales tax. VISA and MasterCard accepted. Descriptive brochures of most products listed here are available on request.

"Apple" is the registered trademark of Apple Computer Inc.

**subLOGIC**

**Communications Corp.**
**713 Edgebrook Drive**
**Champaign, IL 61820**
**(217) 359-8482**
**Telex: 206995**

NO. 49 JUNE 1982

# MICRO™

THE 6502/6809 JOURNAL

**STAFF**

President/Editor-in-Chief
ROBERT M. TRIPP

Publisher
MARY GRACE SMITH

Senior Editor
LAURENCE KEPPLE

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 HENSDILL

Comptroller
DONNA M. TRIPP

Bookkeeper
KAY COLLINS

---

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

---

Subscription/Dealer Inquiries
(617)256-5515

## I/O ENHANCEMENTS

## APPLICATIONS

## PROGRAMMING LANGUAGES

## MACHINE-LANGUAGE AIDS

## DEPARTMENTS

# COMMUNICATE!

## "The Professional" Series from SDS

*NEW Apple II terminal software*

- **Z-Term "The Professional"™** by Bill Blue, for Apple CP/M*
- **P-Term "The Professional"™** by Joel Kunin and Bill Blue, for Apple Pascal**
- **ASCII Express "The Professional"™** by Mark Robbins and Bill Blue, for Apple DOS**

## Businessmen

**Q.** Do you have difficulty operating your printer when connected to a time-sharing computer? Are files you're trying to download too large for your system buffer? Does your host computer lose data when you send files to it?

**A.** "The Professionals" incorporate printer ring buffers which allow slower printers to accept data at their own rates. Very large files are easily received by periodically saving the buffer to disk. Unlike some software which can lose data during disk saves, "The Professionals" not only direct the host to stop, but actually wait for it to respond before performing the save. After a successful save, the host is automatically directed to continue. This process may be repeated indefinitely. Lost data during send is virtually eliminated by the widest variety of send options available in any communications software. "The Professionals" ensure fast, reliable data transfer of any valuable business information.

## Authors

**Q.** Does your line of work involve sending written material to others? Are you a program author who would like to send work in progress to a partner or client and know that it arrived intact? What would the ability to instantly send material or programs to anyone at any time be worth to you?

**A.** "The Professionals" provide the ideal way to send your articles, manuscripts, reports, programs and technical documents to another computer with phone line access. Now you can work WHEREVER you want, and be assured that your data is sent to its destination quickly and error-free. In fact, compared to the fastest mail services, "The Professionals" offer immediate delivery and will save you the purchase price in just a few uses.

## Students

**Q.** Are you bothered by limited access to your school's existing terminals? Would you like to be able to do your school assignments at home at your own convenience?

**A.** "The Professionals" allow you to access virtually any dial-up school or college computer system over standard telephone lines. This means no more waiting in line for an available terminal or hassles with malfunctioning school equipment. You can even prepare term papers or reports while off-line and send the completed work to the school computer for final printing. Best of all, you can work from home at the times most convenient for you.

## Time Share Users

**Q.** Are you tired of wasting time and money sending or receiving files with inadequate, poorly designed software? Do you find yourself manually performing the same lengthy log-in procedures over and over again? Would you like to automate these procedures for yourself and others?

**A.** "The Professionals" allow you to send files which have been prepared in advance. They may then be transferred at any time, as quickly as possible — even to several different systems. No time is wasted reviewing information while on line; data may be captured by your computer or printer (or both) to be evaluated later at your convenience. These features assure minimum on-line time and therefore minimum on-line cost.

"The Professionals" introduce macros that are more sophisticated than anything previously seen in communications software. These "hand-shaking" macros allow you to perform complete multi-stage log-on sequences automatically; all you do is specify the system to be called. This eliminates sign-on errors and greatly simplifies operation of the entire system, not only for you, but for other less skilled operators.

## Bulletin Boards

**Q.** Would you like to be able to take advantage of the information featured on local bulletin boards and information services such as The Source, CompuServe, Dow Jones, and others?

**A.** "The Professionals" open the world of modem communication networks to you. There are already thousands of these systems and networks in use nationwide. "The Professionals" provide an ideal way of accessing these systems. All 80 column boards, external terminals (even the 40 column screen), and currently available communications devices are fully supported, including the Hayes Micromodem II and Novation Apple CAT. All standard baud rates — 110, 300, 1200 and others — are fully supported; BAUDOT too, if your computer is equipped with the Apple CAT modem.

## Clubs

**Q.** Are there other Apple owners with whom you would like to exchange programs or files, but have been unable to do so because of limitations imposed by the software you now use?

**A.** Any two Apples equipped with "The Professionals" can transfer ANY type or size file with complete error checking and correction. All of "The Professional" packages are fully conversant with each other and operate almost identically. For the first time ever, you can transfer compatible files to an operating system different from yours — error free!

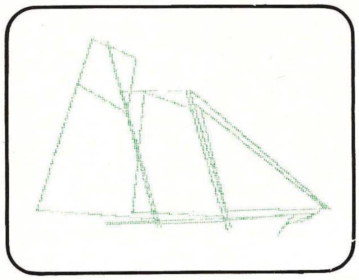
**"The Professional" Series** - Excellence in Apple Communications Software

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.

## About the Cover



The yachts on our cover this month are sailing off British Columbia in the race for the World Championship. Advanced graphics capabilities make computers an excellent tool for sail designers, allowing them to simulate the performance of a proposed design and make adjustments that optimize performance.

Our cover graphic shows the sail and spar plan of the yacht *America*, which won the Royal Yacht Squadron Cup from the British in 1851. The *America*'s design stunned the yachting world of the time and completely outclassed the competition. Computers have since increased the pace of creative innovation.

Cover photo: Brian Dowley Photography
26 Peverell St.
Dorchester, MA 02125
617/288-1530

**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 |

# MICRO™

## Editorial

It was exactly one year ago that MICRO ceased to be exclusively about the 6502 microprocessor and its large family of products. I wrote a four-part series entitled "It's Time to Stop Dreaming," which examined the new 6809 microprocessor from the point of view of a 6502 devotee. Since then I have become personally much more involved with the 6809, by choice, and the 6809 world has expanded and matured in many ways.

On the personal side, MICRO is just one of my activities. I also have a company, The Computerist, Inc., which for many years has been actively developing, manufacturing, and selling products for the 6502, especially expansion boards for the AIM, SYM, and KIM microcomputers. About the time we decided to cover the 6809 in MICRO, we had also decided to have a 6809 as an option on our new Flexi Plus multipurpose expansion board which was in the final stages of development. Well, the 'option' turned out to be the most significant part of the new product, and since last June I have been very actively involved with the 6809. I am no longer the 6809 'novice' I was then. I strongly urge all programming-oriented readers to become acquainted with this device. It has many features which make it a dream to use.

Of greater significance to the MICRO reader is the 6502/6809 explosion. There are several strong points at which the 6502 and 6809 interact. There are now at least two companies making devices to plug into your Apple II to permit it to run a 6809: The Mill by Stellation II and the Excel-9 by ESD Labs. Both processors are standard equipment for the Commodore SuperPET. The Computerist FOCUS system can run with either processor. And the Synertek SYM-1 has a simple 6809 conversion board.

But, exciting as some of these products are, the real significance lies in the software. Almost every 6809 system made will run either the Technical Systems Consultants' FLEX, or the Microware Systems Corporation's OS-9, or both. These two operating systems provide the 'glue' which holds the 6809 world together. Unlike the 6502 world which developed a totally different operating system for each product, the 6809 world has generally accepted these two systems as the starting point for development. The Mill already offers OS-9 and the Excel-9 runs FLEX. Versions of these are available now (or very soon) for the TRS Color Computer. The result is that there is a large population of software which will run on virtually any 6809 system. If you are a purchaser of software, this means that there are more high quality packages available than would be the case if each machine were different. If you are a developer of software for sale, imagine your new product having this diverse population of computers and users as a market.

If you become convinced that it is worth upgrading your existing equipment or even investing in a total 6809-based system, what about all of the software you have already written in 6502 assembler? At least one company, Frank Hogg Laboratory, is already offering a program to translate 6502 code into 6809 code, and others will probably follow. Once you have your 6809 going, if you ever require 6502 code again, there are several 6809-based cross assemblers available which will support the 6502.

If you have written the bulk of your software in BASIC, Pascal, FORTH, COBOL, or C, then life is even easier. All of these languages are fully supported in the 6809 world, and generally in more sophisticated packages than their 6502 brethren (reflecting in part on the superiority of the 6809 in writing position-independent code, using multiple stacks, 16-bit index registers, and so forth).

I believe that inherent superiorities of the 6809, the systematic development of the general operating systems, and the overall quality of the hardware and software that is being offered will make the 6809 a very significant device in your microcomputing future — whether you buy one or not! Therefore, MICRO will continue to provide you substantial information about the 6809 and the family of products which are developing around it.

Robert M. Tripp

# MICRO™

## New Publications

Mike Rowe
New Publications
34 Chelmsford Street
P.O. Box 6502
Chelmsford, MA 01824

---

**Word Processors and Information Processing,** by Dan Poynter. Para Publishing (P.O. Box 4232, Santa Barbara, CA 93103), 1982, 172 pages, 5½ × 8½ inches, paperback.
ISBN 0-915516-31-4 $11.95

This book is designed to aid you in purchasing equipment, products, and services. It will help you determine what your word processing requirements are. Includes a product directory and a resource section.

CONTENTS: The Word Processing Dilemma; What Is Word Processing And How Can It Help Me?; The Parts of the Word Processor: The Choices; Word Processor Functions and Features; Let The Buyer Compare How To Buy A Word Processor; The Price and Other Costs; Appendix: For More Information; Glossary Of Word Processing Terms; Index; Colophon.

**BASIC Programs for Scientists and Engineers,** by Alan R. Miller. Sybex (2344 Sixth Street, Berkeley, CA 94710), 1981, 318 pages, 7 × 9 inches, paperback.
ISBN: 089588-073-3 $14.95

A library of BASIC programs encountered in science and engineering applications. Each program is explained in detail.

CONTENTS: Preface; Introduction; A Note on Typography; Evaluation of a BASIC Interpreter or Compiler; Mean and Standard Deviation; Vector and Matrix Operations; Simultaneous Solution of Linear Equations; Development of a Curve-Fitting Program; Sorting; General Least-Squares Curve Fitting; Solution of Equations by Newton's Method; Numerical Integration; Nonlinear Curve-Fitting Equations; Advanced Applications: The Normal Curve, the Gaussian Error Function, The Gamma Function, and the Bessel Function; Appendix A: Reserved Words and Functions; Appendix B: Summary of BASIC; Bibliography; Index.

**From Chips to Systems: An Introduction to Microprocessors,** by Rodnay Zaks. Sybex (2344 Sixth Street, Berkeley, CA 94710), 1981, 551 pages, 7 × 9 inches, paperback.
ISBN: 0-89588-063 $14.95

A history of microprocessors, including a discussion of its support components and design. No preliminary knowledge of microprocessors needed.

CONTENTS: Preface; Introduction; Fundamental Concepts; Internal Operation of a Microprocessor; System Components; Comparative Microprocessor Evaluation; System Interconnect; Microprocessor Applications; Interfacing Techniques; Microprocessor Programming; Assembly and High-Level Programming; System Development; The Future; Appendices; Index.

---

# Silicon Office: A Review

### by Jim Strasma

*Editor's Note: Because of the size, complexity, and significance of "Silicon Office," we feel it cannot be covered adequately in our normal "Reviews in Brief" format.*

| | |
|---|---|
| Product Name: | **Silicon Office** |
| Equip. req'd: | CBM 8096 and 8000 or higher series CBM disk drive. IEEE modem and CBM or ASCII printer optional. |
| Price: | $999.00 |
| Manufacturer: | Bristol Software Factory<br>P.O. Box 14<br>Horley, Surrey<br>England |

*Silicon Office* is the most significant program yet written for any microcomputer. It claims to handle the daily computer needs of a small business with a single program, and very nearly does! Nothing remotely comparable is available on competing microcomputers. It may well become the Software Sensation of 1982, and a vast boost to Commodore sales. Already it is outselling all other business programs in Europe, where it has been available for six months. This is in spite of a price tag there of $1,600.

What is so special about *Silicon Office*? For one thing, it's the largest single machine-language program ever written for a microcomputer. It occupies 54K bytes, representing 27000 lines of source code. (Triple the size of VisiCalc.) In return for taking nearly all available memory space, *Silicon Office* gives complete freedom in using both drives of the disk unit.

Briefly, *Silicon Office* does three things: 1) it is a database language — the only one available on a microcomputer; 2) it is a word processor — one with amazingly useful features; 3) it is a communicating terminal. It is this multi-functionality which makes it worth nearly any price. Compared to the cost of say, *Jinsam 8.2*, plus *Word Pro 4 Plus*, and some sort of terminal program, it is a bargain.

Gestalt psychotherapists teach that the whole is greater than the sum of its parts. This is certainly true of *Silicon Office*. It is vastly more useful than having three separate programs with similar functions. Each of its modes interacts with the others in very useful ways. For instance, the report generator is the weakest link in most database packages I have used. In *Silicon Office*, the report generator is the word processor, which has full access to the entire database at all times.

The first function of *Silicon Office* is database management. In this role, it is visibly the replacement for *OZZ*, the company's previously best-known program. (*OZZ* is probably best known for its role in helping rescuers during the MGM Grand Hotel fire. It was used there to keep track of thousands of guests.) *OZZ* had its faults, primary among them limited record and file size, single sort key, and fragile non-standard file structure.

*Silicon Office* has overcome these weaknesses. It uses standard Commodore relative files for data, and opens and closes the disk file each time a record is read. This makes it nearly impossible to lose the entire file. To get around the file size limitations of Commodore's relative records and the 8050 disk unit, *Silicon Office* allows each record to consist of up to six separate relative records, allowing an effective record size of 6*252 characters.

Most users, however, won't need records that large, due to the program's ability to link up to six totally independent files in meaningful ways. It is what is called a Relational Database. This makes it arguably the only true database manager for Commodore computers. To my knowledge, only *MDBS* and *DBase II* for the S-100 market, and *DB Master* for the Apple share this capability.

Why do you need a relational database? Let me share an example from my work. As pastor of a 370-member church, I have 300 families to track, each with one to nine members. I have tried over 20 database packages, and every one required total duplication of all information on each family in the records of each family member. They further required that every record have exactly the same fields as every other record. This makes no sense for a church. A weekly record of attendance and contributions isn't needed for a student a thousand miles from home. And it is a horrible waste of time and disk space to copy the same street address into the records of ma, pa, and each of six kids. Imagine the fun when they move across town....

A relational database gets around this problem by defining different kinds of records for different situations, and allowing them to be used together. With *Silicon Office*, I now have three basic record formats: one for each local family, a second for each active member, and a third for distant families.

Each family record contains a reference to the matching individual records, and vice versa. Likewise, the distant records include a number that allows them to be merged in with the family records for all-church mailings. To expand the concept further, I could add a further distinction between adult and child records. Redundant information, such as the town name is supplied from a separate, disk-storable, variable memory with *Silicon Office*.

File structures are easy to design, by drawing them on the video screen. Commodore-recommended forms editing standards are followed. Unlike competing programs, *Silicon Office* allows range checks and validation on each field, and can be designed to allow or prevent duplicate field data in a file's primary sort key.

Like at least one competing package, CMD's *The Manager*, it also has arithmetic functions, although more of them and far better handled than in its competitor. Among these, the most interesting is DATE, which includes the ability to calculate days between dates, and already knows that 1982 isn't a Leap Year. But *Silicon Office* goes one crucial step further, a step that makes it truly a new language. It allows branching, the all-important IF statement. (It is only the lack of this, in my opinion, which keeps VisiCalc from being a language.) *Silicon Office* allows up to a hundred database commands, including IF and GOTO, to be preprogrammed in a simple English-like syntax. These commands can be saved as a file, and also linked to almost any effective program length.

As if that weren't enough, *Silicon Office* also includes true variables, 64 numeric and 26 string. They may have names of up to 16 significant characters, and numeric precision of 14 digits. It is even possible to program in user input prompts for untrained operators. The responses can be stored as variables.

Some of *Silicon Office*'s other database features are: the ability to have tables in memory for data lookup; subsorting from one field to another when matches are found while sorting; the ability to automatically create records for a new file from the record data of other files; searches for records with field data within preset bounds; and wild-card searches for arbitrary strings of characters, either within particular data fields or throughout the entire file. This last feature proceeds at the previously unheard-of (in Commodore circles at any rate) speed of 250 full-length records per minute.

Perhaps by now you have forgotten that this package also includes a word processor? Don't! It is not just another word processor. This one extends the state of the art, very obviously building on the better features of its competitors *Word Pro 4 Plus* and *Wordcraft 80.*

One of my chief complaints about previously available word processors has been the difficulty (or impossibility) of linking text files too large to fit in memory at once. I can't use recent versions of *Word Pro*, for example, to write sermons, simply because they aren't able to hold the full text of a 20-minute message at once, and linking is too much trouble. But in *Silicon Office*, the entire disk may be treated as part of the text area. You pass from one file to another, up to a maximum of 150-180 printed pages, by simply scanning from "page" to "page" of memory. There is sufficient buffering of text that the disk access time is not noticeable.

Like *Wordcraft 80, Silicon Office* is based on the idea that "What you see is what you get." Formatted output can be wider than the standard 80-column screen up to a maximum of 125 columns; both include horizontal scrolling like that in VisiCalc. Text is formatted immediately on the screen. However, unlike *Wordcraft 80*, this formatting is done immediately. There is no chance for the typist to get ahead of the word processor, or to have to wait until what was just typed appears on the screen. Inserts and deletes are done at full-speed. Whole chapters may be quickly inserted in the middle of existing text files.

Like the other programs mentioned, *Silicon Office* justifies text. However, it justifies *within* words, not merely between them as on the others, leading to an extremely smooth appearance of finished text. Even more important, it is easily able to handle true multicolumn documents, in a single pass through the printer. This makes it the only Commodore-compatible word processor suitable for newsletter formatting. Naturally, it also works well with nearly any ASCII daisy wheel printer, and can send any needed ASCII sequence.

Another amazing feature of *Silicon Office* (that has people in Europe trading in dedicated word processors) is its ability to perform truly complex calculations of text data and format the results, *at print time*. And we haven't forgotten about the database manager, have we? The word processor can use any variable or field data anywhere in the currently accessible data files. It can even pause while changing data diskettes. And if the printer fouls up, it can restart from the top of any page.

Both the data manager and the word processor handle "find" and "change" tasks more intelligently than competing programs. Matches are made without regard to the case, upper or lower, of the text being searched. But if a change is made, the new text is made to conform to the case of the replaced text, even in words containing both upper and lower case text.

The communications capability requires a *Silicon Office* at both ends of the line. But within that limitation, it is possible to do almost anything from a remote terminal that could be done locally, including sending every kind of file used by *Silicon Office.*

Officially it is for the Commodore 4010 modem, which is now hard to get. However, any RS232 modem should work, if connected to the IEEE-488 port *via* a bi-directional interface, such as the TNW 2000.

The sellers are obviously making great efforts at support. Already available is an application library that includes the equivalent of five megabytes of data file examples, including: inventory, payroll, general ledger, job control, and so on. There is also a support Hotline, a newsletter of user applications, and a videocassette training course. There is also, thankfully, a two year guarantee on the program, including needed upgrades for such predicted events as Commodore's forthcoming DOS 3.0 and hard disks. And the manuals are just perfect. The training guide teaches all the essentials in one or two hours, and the HELP screens within the program reference a complete and well-organized reference manual. There is also a complete programmer's reference section on file structures, telling how to access them from other programs.

What don't I like about *Silicon Office*? I can sum it up in one word — protection. With 4½ man years of programming effort in the program, I certainly appreciate why the Bristol Software Factory doesn't want the user giving away their program. However, they only supply one spare diskette, and have gone to great pains to be sure no one copies it. Thus, if I trash a diskette, I'll have to be extra careful with the spare for the week or so it takes to mail in the dead diskette and get back a replacement. Also, I can't make any changes to handle special needs.

Having watched the throes of foreign DOS disk protection on the Apple II for the past two years, I don't welcome this first appearance of it on the Commodore disk drives. Fortunately, once *Silicon Office* is loaded, its diskette may be safely filed away until the next day (assuming nothing resets the computer meanwhile.) On that basis, with normal care, a diskette should last a long time.

Since I'm quibbling, I may as well mention that data files are stored in screen codes and need conversion for use in other programs. Fortunately, the needed fixes are included in the manual.

Do I recommend *Silicon Office*? You'd better believe I do! It is the single best program I have ever seen!

*For more information contact Associated Marketing Systems, Inc., 55 Central Dr., Farmingdale, NY 11735.*

---

MICRO™

# Disk to Tape Backup Utility

### by Richard Merten

**In this article the author demonstrates ways to save money on archives by using cassette tape to back up and restore every sector of a disk.**

**Disk to Tape**
requires:

Apple II or Apple II Plus
48K, DOS 3.2 or 3.3
Integer BASIC in language card or motherboard

This program for making backup tapes is easy to use and will run with either 13- or 16-sector diskettes. It is short — less than one page of machine language and about 50 lines of Integer BASIC. It also contains error checking routines that alert you to problems from both the disk and the cassette. An entire 16-sector disk can be sent to tape in about 15 minutes. As many as four full disks can be stored on a single C-60 cassette, or six on a C-90.

To transfer a disk to cassette, first run the Backup program from a diskette using the same DOS as the disk that you are going to duplicate. (This is important because the backup program uses the RWTS routines in the Apple to read and write information to the disk.)

The BASIC portion loads the machine language routine then asks if you want to save or restore a disk. For two-drive systems you can put the disk in either drive 1 or 2, then turn on the tape recorder and press return. If you are saving a DOS 3.3 disk no problems should occur, but if you are saving a DOS 3.2 disk to tape, there will probably be at least two sectors on track 2 that cannot be read.

If a sector cannot be read, the track and sector that DOS is unable to read will be printed to the screen. These unreadable sectors are filled with zeros before they are sent to the tape. This

**Listing 1**

```
 10 REM  BACKUP DISK TO TAPE UTILITY
 20 REM             BY
 30 REM         R.C. MERTEN
 40 REM
 50 REM           7/28/81
 60 REM
 70 LOMEM:24576
 80 D$="": REM  CONTROL D
 90 PRINT D$;"BLOAD TAPBAK.OBJ"
100 REM  GET ADDRESS OF SECTOR NUMBER AND LENGTH FROM MACHINE LANGUAGE
110 TOTAPE=256* PEEK (2049)+ PEEK (2048):FRMTAPE=256* PEEK (2051)+ PEEK
    (2050)
120 SECLEN=256* PEEK (2053)+ PEEK (2052):SECNUM=256* PEEK (2055)+ PEEK
    (2054)
130 REM  PEEK AT WHICH VERSION DOS (IN 48K MACHINE)
140 DOS= PEEK (-19522): IF DOS=3 OR DOS=2 THEN 160
150 CALL -936: VTAB 13: PRINT "ERROR!! EITHER NOT A 48K MACHINE OR PAGE3
        HAS BEEN ALTERED": PRINT : PRINT "REBOOT DOS AND TRY AGAIN": END
160 IF DOS=3 THEN 190
170 REM  IF DOS VERSION 3.2 POKE IN NEW SECTOR NUMBER AND SECTOR LENGTH
180 POKE SECNUM+1,12: POKE SECLEN+1,80
190 DRIVE=2:TRACK=3:ERR=6: DIM A$(10)
200 CALL -936: VTAB 4: PRINT "       DISK TO TAPE BACKUP UTILITY"
210 PRINT : PRINT "                BY R.C. MERTEN"
220 PRINT : PRINT "                  FOR DOS 3.";DOS
230 VTAB 14: PRINT "THIS UTILITY WILL SAVE OR RESTORE AN": PRINT : PRINT
    "ENTIRE DISK TO TAPE, DOS AND ALL!"
240 VTAB 20
250 PRINT : PRINT "IT TAKES ABOUT 15 MINUTES TO COMPLETE": PRINT : PRINT
    "PRESS RETURN TO CONTINUE"
260 INPUT "",A$
270 CALL -936
280 VTAB 3: PRINT "       DOS 3.";DOS;" TAPE BACKUP UTILITY "
290 VTAB 6: PRINT "          1) SAVE TO TAPE"
300 PRINT : PRINT "          2) RESTORE TO DISK"
310 PRINT : PRINT : INPUT "          WHICH ",W
320 IF W<1 OR W>2 THEN 270
330 VTAB 11: INPUT "       ENTER DRIVE # (1 OR 2) ",DR: IF DR<1 OR DR>
    2 THEN 330
340 POKE DRIVE,DR
350 VTAB 16: PRINT "PUT DISK IN DRIVE #";DR;" START TAPE RECORDER":PRINT
    "AND PRESS RETURN"
360 INPUT "",A$
370 VTAB 15: CALL -958
380 IF W=2 THEN 510
390 ERRT=0
400 FOR X=0 TO 34 STEP 5: VTAB 13: PRINT "      NOW BACKING UP TRACKS "
    ;X;" TO ";X+4
410 POKE TRACK,X
420 CALL -958
430 VTAB 15: TAB 9
440 POKE 50,63: PRINT "BAD TRACK & SECTOR LIST": POKE 50,255
450 PRINT "T$   S$"
460 CALL TOTAPE
470 ERRT=ERRT+ PEEK (ERR): NEXT X
480 PRINT : PRINT "      ";ERRT;" ERRORS! WANT TO BACKUP ANOTHER?"
490 INPUT "",A$: IF A$="Y" THEN 270

500 END
510 FOR X=0 TO 34 STEP 5: VTAB 15: PRINT "      NOW RESTORING TRACKS "
    ;X;" TO ";X+4
520 POKE TRACK,X
530 CALL FRMTAPE
540 IF PEEK (ERR)=0 THEN NEXT X
550 IF PEEK (ERR)=128 THEN GOTO 590
```

*(Continued)*

presents no problem when the disk is reconstructed since they are not used during the booting process.

A read error from the disk will cause an error indication to appear on the screen. However, a read error from cassette is considered fatal and causes the program to bomb gracefully, asking if you would like to try again.

## How it Works

Information is moved from tape to disk and back again five tracks at a time, a total of seven times through the loop to accommodate all 35 tracks. The program keeps you posted on its progress with beeps and other updates along the way, and totals any errors at the end.

The BASIC portion of the program is primarily concerned with prompting the user, and general housekeeping activities. The assembly language portion does most of the work, so the detailed explanation will start there.

Two CALLs are possible from BASIC. A series of subroutines called TOTAPE moves information from disk to tape. Another series called FRMTAPE moves data from the cassette onto the disk. In either case the 6502 registers are first saved for the return trip to BASIC. Next, the page zero addresses starting at $10 are moved into a temporary buffer starting at $910. This is done because the RWTS routine changes page zero data that makes it impossible to correctly return to the BASIC caller.

Next the location of the DOS IOB routine is located with a JSR to location $3E3. On return the IOB address is in the accumulator and Y registers. This address is saved at IOBADD and used to index information in and out of the IOB.

After some registers have been set up, the program loops through the GETRAX and NXTRAK routines until five full tracks have been read into (or written out of) a buffer area that begins at $1000. If at any time the processor returns from the RWTS routine with its carry flag set, it will then jump to the ERROR routine. After all five tracks have been moved, the page zero values are restored and the program returns to BASIC for updating, then back into the loop to get five more tracks.

## The Error Routine

If an error occurs while reading from the disk, the ERROR routine increments an error counter, then fills with

**Listing 1** *(Continued)*

```
560 PRINT : INPUT "          REBUILD ANOTHER DISK? ",A$
570 IF A$="Y" THEN 270
580 END
590 PRINT : PRINT
600 PRINT "        ****  TAPE READ ERROR  ****"
610 PRINT : INPUT "            WANT TO TRY AGAIN? ",A$: IF A$="Y"

620 END
```

**Listing 2**

```
                              ORG   $800
                              OBJ   $6000
               *
               ******************************
               *                            *
               *       DISK TO TAPE         *
               *          BACKUP            *
               *                            *
               *            BY              *
               *          R.C.M.            *
               *          7/13/81           *
               *                            *
               ******************************
               *
               IOBADD    EQU   $00
               DRIVE     EQU   $02
               TRACK     EQU   $03
               SECTOR    EQU   $04
               COMAND    EQU   $05
               ERR1      EQU   $06
               COUNT     EQU   $07
               READCH    EQU   $08
               BUFLO     EQU   $09
               BUFHI     EQU   $0A
               CH        EQU   $24
               A1L       EQU   $3C
               A1H       EQU   $3D
               A2L       EQU   $3E
               A2H       EQU   $3F
               RWTS      EQU   $3D9
               FNDIOB    EQU   $3E3
               TEMP      EQU   $900
               SAVE      EQU   $FF4A
               RESTORE   EQU   $FF3F
               WRITE     EQU   $FECD
               READ      EQU   $FEFD
               PRBLNK    EQU   $F948
               PRBYTE    EQU   $FDDA
               CROUT     EQU   $FD8E
               *
0800: 08 08                DA    TOTAPE.
0802: 21 08                DA    FRMTAPE.
0804: B5 08                DA    SECLEN
0806: 74 08                DA    SECNUM
               *
0808: 20 4A FF TOTAPE    JSR   SAVE.
080B: 20 D7 08           JSR   SAV1.
080E: 20 4B 08           JSR   SETUP.
0811: A9 01              LDA   #$01    * SET READ COMMAND
0813: 85 05              STA   COMAND.
0815: 20 7A 08           JSR   GETRAX.
0818: 20 A9 08           JSR   SETTAP.
081B: 20 CD FE           JSR   WRITE.
081E: 4C E2 08           JMP   REST1.
               *
0821: 20 4A FF FRMTAPE   JSR   SAVE.
0824: 20 D7 08           JSR   SAV1.
0827: 20 4B 08           JSR   SETUP.
082A: A9 02              LDA   #$02    * SET WRITE MODE
082C: 85 05              STA   COMAND.
082E: 20 A9 08           JSR   SETTAP.
0831: A5 24              LDA   CH.
0833: 85 08              STA   READCH.
0835: 20 FD FE           JSR   READ.
0838: A5 24              LDA   CH.
083A: C5 08              CMP   READCH.
083C: F0 07              BEQ   OK.
083E: A9 80              LDA   #$80
0840: 85 06              STA   ERR1.
0842: 4C E2 08           JMP   REST1.
0845: 20 7A 08 OK        JSR   GETRAX.
0848: 4C E2 08 RTS       JMP   REST1.
               *
084B: 20 E3 03 SETUP     JSR   FNDIOB * GET IOB ADDRESS
```

**Listing 2** *(Continued)*

```
084E: 84 00             STY  IOBADD * AND SAVE AT IOBADD
0850: 85 01             STA  IOBADD+1
0852: A9 05             LDA  #$05   * SET TRACK COUNT TO 5
0854: 85 07             STA  COUNT.
0856: A9 10             LDA  #$10   * SET BUFFER TO $1000
0858: 85 0A             STA  BUFHI.
085A: A5 02             LDA  DRIVE  * GET DRIVE #
085C: A0 02             LDY  #$02
085E: 91 00             STA  (IOBADD),Y
0860: A9 00             LDA  #$00
0862: 85 04             STA  SECTOR * SET SECTOR 0
0864: 85 06             STA  ERR1   * ZERO ERR BYTE
0866: 85 09             STA  BUFLO.
0868: C8                INY
0869: 91 00             STA  (IOBADD),Y * SET VOLUME TO 0
086B: A0 08             LDY  #$08
086D: 91 00             STA  (IOBADD),Y * SET LOW BUFFER ADDRESS
086F: 60       RTS1     RTS
               *
0870: E6 0A    LOOP1    INC  BUFHI  * NEXT MEMORY PAGE
0872: A5 04             LDA  SECTOR * NEXT SECTOR
0874: C9 0F    SECNUM   CMP  #$0F
0876: F0 24             BEQ  NXTRAK * DONE WITH THIS TRACK
0878: E6 04             INC  SECTOR * TO NEXT SECTOR
087A: A5 03    GETRAX   LDA  TRACK.
087C: A0 04             LDY  #$04   * SET TRACK TO READ
087E: 91 00             STA  (IOBADD),Y
0880: C8                INY
0881: A5 04             LDA  SECTOR * SECTOR TO READ
0883: 91 00             STA  (IOBADD),Y
0885: A5 05             LDA  COMAND * READ OR WRITE
0887: A0 0C             LDY  #$0C
0889: 91 00             STA  (IOBADD),Y
088B: A5 0A             LDA  BUFHI  * SET MEMORY PAGE
088D: A0 09             LDY  #$09
088F: 91 00             STA  (IOBADD),Y
0891: 20 E3 03          JSR  FNDIOB * ADDRESS OF IOB
0894: 20 D9 03          JSR  RWTS   * READ T/S
0897: B0 21             BCS  ERROR  * IF CARRY SET THEN ERROR
0899: 4C 70 08          JMP  LOOP1  * GET NEXT SECTOR
               *
089C: C6 07    NXTRAK   DEC  COUNT  * 5 TRACKS PER LOAD
089E: F0 CF             BEQ  RTS1   * IF DONE
08A0: E6 03             INC  TRACK.
08A2: A9 00             LDA  #$00   * RESET TO SECTOR ZERO
08A4: 85 04             STA  SECTOR.
08A6: 4C 7A 08          JMP  GETRAX.
               *
08A9: A9 00    SETTAP   LDA  #$00   * SET BOUNDRY FOR TAPE MOVE
08AB: 85 3C             STA  A1L.
08AD: A9 FF             LDA  #$FF
08AF: 85 3E             STA  A2L.
08B1: A9 10             LDA  #$10
08B3: 85 3D             STA  A1H.
08B5: A9 5F    SECLEN   LDA  #$5F
08B7: 85 3F             STA  A2H.
08B9: 60                RTS
               *
08BA: E6 06    ERROR    INC  ERR1   * INCREMENT COUNTER
08BC: A9 00             LDA  #$00   * FILL SECTOR WITH ZEROS
08BE: A8                TAY
08BF: 91 09    ERLOOP   STA  (BUFLO),Y
08C1: C8                INY
08C2: D0 FB             BNE  ERLOOP.
08C4: A5 03             LDA  TRACK  * PRINT BAD TRACK & SECTOR
08C6: 20 DA FD          JSR  PRBYTE.
08C9: 20 48 F9          JSR  PRBLNK.
08CC: A5 04             LDA  SECTOR.
08CE: 20 DA FD          JSR  PRBYTE.
08D1: 20 8E FD          JSR  CROUT.
08D4: 4C 70 08          JMP  LOOP1.
               *
08D7: A2 10    SAV1     LDX  #$10
08D9: B5 00    SAVLOP   LDA  $00,X
08DB: 9D 00 09          STA  TEMP,X
08DE: E8                INX
08DF: D0 F8             BNE  SAVLOP.
08E1: 60                RTS
               *
08E2: A2 10    REST1    LDX  #$10
08E4: BD 00 09 RESTLOP  LDA  TEMP,X
08E7: 95 00             STA  $00,X
08E9: E8                INX
08EA: D0 F8             BNE  RESTLOP.
08EC: 20 3F FF          JSR  RESTORE.
08EF: 60                RTS  *      RETURN TO CALLER
```

zeros that page in the buffer that didn't receive any data. Finally it prints the faulty track and sector location to the screen and returns to read the next sector.

## The BASIC Program

A routine which checks out the system to see which DOS is in effect is at the beginning of the BASIC program. If it is 3.2, values are POKEd into two locations of the assembly routine. These routines define the read buffer boundries and the number of sectors per track to read. If the RWTS pointers on page three are missing, an error message is encountered and the program tells you to reboot the disk before running the program again.

The BASIC portion of this program is straightforward with one exception. Line 70 contains the illegal command LOMEM:. Before entering any part of the BASIC program, first type NEW to kill any existing program. As the first line of BASIC, type "70 LIST 24576". Next type CALL −151 to get to the monitor. When in the monitor, type CA CB and RETURN to find the pointers for the start of the program (low byte first). On a 48K machine this will be $95FB.

Print $95FBL to get a list of the tokenized program. About four numbers from the start, you should find a $74 character. Change this to $11 and then re-enter BASIC with a control C. Now LIST the program and you should see "70 LOMEM:24576". Now type in the rest of the BASIC program and save it to disk.

## Save the Machine Code

After assembling the machine code, save it on disk with the command "BSAVE TAPBAK.OBJ,A$800,L$F0". This routine will be loaded from line 90 of the integer routine when it is first run.

## Caution

If you are using the DOS 3.3 and have 3.2 disks that have been modified with one of the universal boot routines, you may experience difficulty in using this backup routine. There are just too many sectors in the modified DOS that cannot be read by the 3.2 RWTS routine. My suggestion is to first MUFFIN all the programs onto a 3.3 disk and then make the backup for that disk. They can later be DEMUFFINED to another universal 3.2 disk if desired.

---

Richard Merten may be contacted at 12307 Oak St., Omaha, NE 68144.

HDE inc.

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 924
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

# AIM User Function Dispatcher

### by Joel Swank

**Overcome the AIM's limitation of three user-definable keys with "Function Dispatcher." Up to 85 different commands may be defined. Four samples are provided.**

**Function Dispatcher**
requires:
AIM 65

The AIM 65 monitor reserves three keys for user-written commands. Pressing one of these keys causes a jump to a vector in RAM page one. You can then put a jump to your routine in this vector. It is convenient to be able to execute a program with a single keystroke. But the programs available for the AIM far outnumber the available keys. And most of the AIM disk systems use one or two of these keys. In addition, relatively inexpensive 16K and 32K RAM boards are now available that allow plenty of space to keep many frequently used utilities resident in memory. Some RAM boards even have the ability to write-protect sections of memory, making them perfect for containing user extensions of the AIM monitor. The AIM needs a way to execute many more than three user commands; User Function Dispatcher fills this need.

The dispatcher expands one of the AIM user function keys to allow execution of nearly as many routines as there are characters on the AIM keyboard. The limit is actually 85 because the command table is limited to 255 bytes. Listing 1 includes the dispatcher and four sample commands executed through the F3 key. The command names may be multiple characters, but only the first character is entered to execute the command. For example, to execute the clear command you press the F3 key and the 'C' key; the dispatcher then echoes 'CLEAR'. This feature not only gives visual confirmation that you are executing the proper command, but also makes the commands easier to remember.

You may add your own routines to the dispatcher by adding entries to the command table. Each entry in the table is composed of three or more bytes. The first byte is the character that must be entered to execute the command. The rest of the command name follows. The last character of the name must have the high bit of the byte set to one. The name may be as long as you like, but the entire table is limited to 255 bytes. Following this is the two-byte address of the start of the routine in normal low-byte, high-byte order. The table must end with a $00 byte. The routine should end with a jump to the monitor at address $E1A1.

The four commands included are VIEW, CLEAR, ECHO and TEST. VIEW displays a section of memory in hex and ASCII. It requests that the starting and ending addresses of the memory be displayed with the standard 'FROM=' and 'TO=' prompts. It then uses the 'OUT=' prompt to allow you to direct the display to any system device. If the output device is tape, VIEW sends a double CR and a control-Z to end the display and closes the file. This allows the output to be read back by either the AIM editor or BASIC. VIEW calls AIM subroutine RCHEK at the end of each line to allow you to stop or cancel the display.

CLEAR, a routine to clear the AIM editor buffer, allows you to delete all text in the editor buffer without reinitializing the editor.

ECHO is a routine that executes from the AIM DILINK vector. ECHO sends a copy of AIM keyboard/display I/O to the TTY port. This allows you to work from the AIM keyboard/display but get a copy of everything on your TTY or CRT. The ECHO command is used to toggle this feature on and off. ECHO responds with the standard 'ON' and 'OFF' messages. You may use ECHO even if you do not have a CRT or TTY to slow the display by setting the terminal speed to a low baud rate. Page 9-32 of the *AIM Users Guide* explains how to set the terminal speed. ECHO has no effect when in TTY mode. Because of the way ECHO is implemented, it will not execute from ROM or write-protected RAM.

TEST is a dummy command to allow easy testing of new routines. The address in the table for TEST is the monitor return, $E1A1. To use TEST, replace this address with the address of the routine to be tested. Below is a sample run of the four commands.

**Sample Run**

```
<↑>VIEW FROM=E000 TO=E022 OUT=
E000  46524F4D  FROM
E004  BD544FBD   TO
E008  202A2A2A   ***
E00C  2A205053  * PS
E010  20414120   AA
E014  58582059  XX Y
E018  592053D3  Y S
E01C  4D4F5245  MORE
E020  BF4F4EA0   ON
<↑>CLEAR
<↑>TEST
<↑>ECHO ON
<↑>ECHO OFF
<
```

The author may be contacted at 25730 Beach Dr., Rockaway, OR 97136.

**Listing 1**

```
               ;*
               ;* AIM-65 USER FUNCTION DISPATCHER
               ;*
               ;EXPANDS THE F3 KEY TO DISPATCH UP TO 85 USER COMMANDS
               ;
               ;ZERO PAGE
               ;
               CMDSAV    EPZ $25             ;COMMAND SAVE AREA
               STRL      EPZ $26             ;START OF LINE
               STRH      EPZ $27
               BOTLN     EPZ $E1             ;BOTTOM OF TEXT
               TEXT      EPZ $E3             ;START OF TEXT BUFFER
               ;
               ;AIM SUBROUTINES
               ;
               FROM      EQU $E7A3           ;INPUT FROM ADDRESS
               TO        EQU $E7A7           ;INPUT TO ADDRESS
               RCHEK     EQU $E907           ;CHECK FOR INTERRUPT
               CRLF      EQU $E9F0           ;SEND CR & LF
               NUMA      EQU $EA46           ;SEND ACCUM IN HEX
               OUTALL    EQU $E9BC           ;SEND ACCUM TO AOD
               WHEREO    EQU $E871           ;OPEN OUTPUT DEVICE
               DU12      EQU $E511           ;CLOSE TAPE FILE
               READ      EQU $E93C           ;INPUT FROM KBD WITHOUT ECHO
               CKER00    EQU $E394           ;DISPLAY 'ERROR'
               COMIN     EQU $E1A1           ;RETURN TO AIM
               OUTPUT    EQU $E97A           ;DISPLAY CHAR IN ACCUM
               BRK3      EQU $E6F1           ;DISPLAY 'OFF'
               BRK4      EQU $E6FA           ;DISPLAY 'ON'
               DILINK    EQU $A406           ;DISPLAY VECTOR
               OUTTTY    EQU $EEA8           ;OUTPUT TO TTY
               ;
               ;AIM RAM
               ;
               CURAD     EQU $A41C
               OUTFLG    EQU $A413
               ;
               ;EQUATES
               ;
               CTLZ      EPZ $1A
               ;
                         ORG $112

               ;
0112 4C 00 0E              JMP UFD             ;INIT USER F3 VECTOR
0115           ;
0E00                       ORG $E00
0E00
0E00           ;
0E00 20 3C E9  UFD         JSR READ            ;GET COMMAND CHARACTER
0E03 A2 00                 LDX #$00            ;CLEAR INDEX
0E05 85 25                 STA CMDSAV          ;SAVE COMMAND
0E07 BD 42 0F  NXTENT      LDA CMDTBL,X        ;COMMAND FROM TABLE
0E0A F0 30                 BEQ NOFOUN          ;QUIT ON NULL
0E0C 29 7F                 AND #$7F            ;CLEAR HIGH BIT
0E0E C5 25                 CMP CMDSAV          ;MATCH?
0E10 F0 0D                 BEQ EXECMD          ;YES, GO DISPATCH
0E12           ;
0E12           ;SKIP TO NEXT COMMAND
0E12           ;
0E12 BD 42 0F  SKLUP       LDA CMDTBL,X        ;GET A CHARACTER
0E15 30 03                 BMI SKIPAD          ;QUIT IF HIGH BIT IS 1
0E17 E8                    INX                 ;NEXT CHARACTER
0E18 D0 F8                 BNE SKLUP
0E1A E8        SKIPAD      INX                 ;SKIP ADDRESS
0E1B E8                    INX
0E1C E8                    INX
0E1D D0 E8                 BNE NXTENT
0E1F           ;
0E1F BD 42 0F  EXECMD      LDA CMDTBL,X        ;GET A COMMAND CHARACTER
0E22 48                    PHA
0E23 20 7A E9              JSR OUTPUT          ;ECHO CHARACTER
0E26 68                    PLA
0E27 30 03                 BMI GETADD          ;QUIT IF HIGH BIT ON
0E29 E8                    INX
0E2A D0 F3                 BNE EXECMD          ;NEXT CHARACTER
0E2C E8        GETADD      INX
0E2D BD 42 0F              LDA CMDTBL,X        ;MOVE ADDRESS OF ROUTINE
0E30 8D 1C A4,             STA CURAD           ;TO CURAD
0E33 BD 43 0F              LDA CMDTBL+1,X
0E36 8D 1D A4              STA CURAD+1
0E39 6C 1C A4              JMP (CURAD)         ;JUMP TO IT
0E3C           ;
0E3C           ;UNKNOWN COMMANDS
0E3C           ;
0E3C 20 94 E3  NOFOUN      JSR CKER00          ;DISPLAY 'ERROR'
0E3F 4C A1 E1              JMP COMIN           ;RETURN TO AIM
0E42           ;
0E42           ;*
0E42           ;* VIEW--
```

**Listing 1** *(Continued)*

```
0E42           ;*  DISPLAY A FORMATTED DUMP OF MEMORY
0E42           ;*
0E42           ;
0E42           ;CALLING FORMAT
0E42           ;
0E42           ;  <^>VIEW FROM= TO= OUT=
0E42           ;
0E42 20 E7 0E  VIEW        JSR BLANK           ;SEND A SPACE
0E45 20 A3 E7              JSR FROM            ;GET FROM ADDRESS
0E48 B0 F8                 BCS VIEW
0E4A AD 1C A4              LDA CURAD           ;SAVE ADDRESS
0E4D 85 26                 STA STRL
0E4F AD 1D A4              LDA CURAD+1
0E52 85 27                 STA STRH
0E54 20 A7 E7  VTL         JSR TO              ;GET END ADDRESS
0E57 B0 FB                 BCS VTL
0E59 20 71 E8              JSR WHEREO          ;GET OUTPUT DEVICE
0E5C 4C 65 0E              JMP FIRST
0E5F 20 07 E9  MORE        JSR RCHEK           ;CHECK FOR INTERRUPT
0E62 20 F0 E9              JSR CRLF            ;NEW LINE
0E65 A5 27     FIRST       LDA STRH            ;PRINT ADDRESS
0E67 20 46 EA              JSR NUMA
0E6A A5 26                 LDA STRL
0E6C 20 46 EA              JSR NUMA
0E6F 20 E4 0E              JSR BLANK2          ;PRINT BLANKS
0E72 A0 00                 LDY #$00
0E74 B1 26     BYLUP       LDA (STRL),Y        ;GET A CHARACTER
0E76 20 46 EA              JSR NUMA            ;PRINT IT IN HEX
0E79 C8                    INY
0E7A C0 04                 CPY #$04            ;IN GROUPS OF 4
0E7C D0 F6                 BNE BYLUP
0E7E 20 E4 0E              JSR BLANK2
0E81 A0 00                 LDY #$00            ;NOW IN ASCII
0E83 B1 26     ASLUP       LDA (STRL),Y        ;GET A CHARACTER
0E85 30 04                 BMI PBL             ;SKIP NON ASCII
0E87 C9 20                 CMP #$20            ;AND CONTROL CHARACTERS
0E89 B0 02                 BCS PTO
0E8B A9 20     PBL         LDA '               ;SUBSTITUTE BLANK
0E8D 20 BC E9  PTO         JSR OUTALL          ;PRINT IT
0E90 C8                    INY
0E91 C0 04                 CPY #$04
0E93 D0 EE                 BNE ASLUP           ;DO 4 TIMES
0E95 18                    CLC
0E96 A5 26                 LDA STRL            ;ADD 4 TO POINTER
0E98 69 04                 ADC #$04
0E9A 85 26                 STA STRL
0E9C 90 02                 BCC PCY
0E9E E6 27                 INC STRH
0EA0 A5 27     PCY         LDA STRH            ;CHECK FOR END
0EA2 CD 1D A4              CMP CURAD+1         ;GET HIGH BYTE OF POINTER
0EA5 90 B8                 BCC MORE
0EA7 D0 09                 BNE DFIN            ;QUIT IF GREATER
0EA9 A5 26                 LDA STRL            ;EQUAL, CHECK LOW
0EAB CD 1C A4              CMP CURAD
0EAE F0 AF                 BEQ MORE            ;CONTINUE IF LESS THAN OR =
0EB0 90 AD                 BCC MORE
0EB2 AD 13 A4  DFIN        LDA OUTFLG          ;USER OUTPUT?
0EB5 C9 50     PRCHEK      CMP 'P              ;PRINTER?
0EB7 D0 06                 BNE TAPCK           ;NOPE
0EB9 20 F0 E9              JSR CRLF            ;PRINT LAST LINE
0EBC 4C A1 E1              JMP COMIN
0EBF           ;
0EBF C9 54     TAPCK       CMP 'T              ;TAPE?
0EC1 D0 09                 BNE UCHK            ;NOPE
0EC3 20 D6 0E              JSR ENDUO           ;SEND EOF CHARS
0EC6 20 11 E5              JSR DU12            ;YES, CLOSE IT
0EC9 4C A1 E1              JMP COMIN
0ECC           ;
0ECC C9 55     UCHK        CMP 'U              ;USER?
0ECE D0 03                 BNE OUT             ;NO
0ED0 20 D6 0E              JSR ENDUO           ;YES, SEND EOF CHARS
0ED3 4C A1 E1  OUT         JMP COMIN           ;RETURN TO AIM
0ED6           ;
0ED6           ;ENDUO: SEND EOF CHARACTERS TO OUTPUT DEVICE
0ED6           ;
0ED6 20 F0 E9  ENDUO       JSR CRLF            ;END LINE
0ED9 A9 1A                 LDA #CTLZ           ;SEND A CONTROL 'Z'
0EDB 20 BC E9              JSR OUTALL
0EDE 20 F0 E9              JSR CRLF            ;AND A COUPLE OF CR'S
0EE1 4C F0 E9              JMP CRLF
0EE4           ;
0EE4 20 E7 0E  BLANK2      JSR BLANK
0EE7 A9 20     BLANK       LDA #$20
0EE9 4C BC E9              JMP OUTALL
0EEC           ;
0EEC           ;*
0EEC           ;* CLEAR--
0EEC           ;*   CLEAR THE EDIT BUFFER
0EEC           ;*
0EEC           ;
0EEC           ;CALLING FORMAT:
```

**Listing 1** *(Continued)*

```
0EEC               ;  <^>CLEAR
0EEC               ;
0EEC A5 E3         CLEAR    LDA TEXT          ;COPY BUFFER START ADD
0EEE 85 E1                  STA BOTLN         ;TO TEXT END ADD
0EF0 A5 E4                  LDA TEXT+1
0EF2 85 E2                  STA BOTLN+1
0EF4 A9 00                  LDA #$00          ;FLAG END OF TEXT
0EF6 A8                     TAY
0EF7 91 E3                  STA (TEXT),Y
0EF9 4C A1 E1               JMP COMIN         ;RETURN TO AIM
0EFC               ;
0EFC               ;*
0EFC               ;* ECHO TO TTY ROUTINE
0EFC               ;*

0EFC               ;
0EFC               ;CALLING FORMAT--
0EFC               ;
0EFC               ;  <^>ECHO ON
0EFC               ;  <^>ECHO OFF
0EFC               ;
0EFC 48            ECHO     PHA
0EFD 29 7F                  AND #$7F          ;CLEAR HIGH BIT
0EFF C9 0D                  CMP #$0D          ;CR?
0F01 D0 0A                  BNE NOLF          ;NO, SKIP IT
0F03 20 A8 EE               JSR OUTTTY        ;YES, SEND IT
0F06 A9 0A                  LDA #$0A          ;SEND LF
0F08 20 A8 EE               JSR OUTTTY
0F0B 68                     PLA
0F0C 60                     RTS
0F0D 68            NOLF     PLA
0F0E 4C A8 EE               JMP OUTTTY        ;DISPLAY CHAR
0F11               ;
0F11               ;ROUTINE TO TOGGLE ECHO ON AND OFF
0F11               ;
0F11 20 E7 0E      TOGG     JSR BLANK         ;SEND A SPACE
0F14 AD 07 A4               LDA DILINK+1      ;SAVE CURRENT CONTENTS
                                               IF DILINK
0F17 48                     PHA
0F18 AD 06 A4               LDA DILINK
0F1B 48                     PHA
0F1C AD 40 0F               LDA SAVE          ;MOVE CONTENTS OF
                                              SAVE AREA

0F1F 8D 06 A4               STA DILINK        ; TO DILINK
0F22 AD 41 0F               LDA SAVE+1
0F25 8D 07 A4               STA DILINK+1
0F28 68                     PLA
0F29 8D 40 0F               STA SAVE          ;RESTORE PREVIOUS
                                              CONTENTS OF DILINK
0F2C 68                     PLA               ; INTO THE SAVE AREA
0F2D 8D 41 0F               STA SAVE+1
0F30 C9 0E                  CMP /ECHO         ;DID WE SAVE ECHO ADD?
0F32 D0 06                  BNE ON            ;NO, DISPLAY 'ON'
0F34 20 F1 E6               JSR BRK3          ;YES, DISPLAY 'OFF'
0F37 4C A1 E1               JMP COMIN
0F3A 20 FA E6      ON       JSR BRK4          ;DISPLAY 'ON'
0F3D 4C A1 E1               JMP COMIN
0F40               ;
0F40 FC 0E         SAVE     ADR ECHO          ;MAKES ECHO NON-ROMMABLE
0F42               ;*
0F42               ;* COMMAND TABLE
0F42               ;*
0F42 56 49 45      CMDTBL   ASC 'VIE'         ;VIEW
0F45 D7                     BYT $D7           ;'W' OR $80
0F46 42 0E                  ADR VIEW
0F48 43 4C 45               ASC 'CLEA'        ;CLEAR
0F4B 41
0F4C D2                     BYT $D2           ;'R' OR $80
0F4D EC 0E                  ADR CLEAR
0F4F 45 43 48               ASC 'ECH'         ;ECHO
0F52 CF                     BYT $CF           ;'O' OR $80
0F53 11 0F                  ADR TOGG
0F55 54 45 53               ASC 'TES'         ;TEST
0F58 D4                     BYT $D4           ;'T' OR $80
0F59 A1 E1                  ADR COMIN
0F5B               ;

0F5B               ;ADD MORE COMMANDS HERE
0F5B               ;
0F5B 00                     BYT $00           ;MUST END WITH ZERO
0F5C               ;
0F5C                        END
```

# 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/II + , 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**

**APLUS.** The original Structured Basic enhancement for Applesoft Basic. APLUS 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 APLUS does the rest. APLUS 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. APLUS 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 •

# Additional Output Ports for KIM

by S. Henning

**This hardware modification allows your KIM to support seven additional output ports.**

Ralph Tenny described, in the May 1980 issue of MICRO, a technique to increase the number of usable I/O ports on the KIM-1 without adding additional hardware. The KIM-1 features 15 user accessible I/O lines. It has an additional 15 lines which are dedicated to the built-in keyboard and LED display. Tenny described how to use them as additional *input* lines.

My requirement was different. I needed additional *output* lines. I used seven lines to attach an ASCII keyboard — three more for Lew Edwards' Ziptape, plus a memory management unit.

So, when I wanted to add an output printer, I needed an additional seven output lines. Rather than adding an additional 6530 or 6522, I decided to use the approach described by Ralph Tenny.

Inspection of figure 3.5 in the KIM-1 user manual indicated a problem. The important parts are reproduced in figure 1. Port A of the 6530 nicely provides the required seven lines; they are even accessible at the application connector. However, they are already loaded with one standard TTL load, so even adding a 74LSxx circuit would exceed the maximum permissible load.

This is the solution: U17 and U26 in figure 1 go only to resistors R26 through R32, which are 82 Ohm each. An additional 74LSxx load can be added to the output of U17 and U26 without overloading the circuit. Figure 2 shows the location of these resistors on the KIM-1. They are readily accessible, and soldering seven wires to them is done easily.

These are the required connections:

| Port | Connection to |
|---|---|
| PA0 | R26, top side |
| PA1 | R27, left side |
| PA2 | R28, left side |
| PA3 | R29, left side |
| PA4 | R30, left side |
| PA5 | R31, top side |
| PA6 | R32, top side |

The subroutine in listing 1 outputs one byte to the new port.

Because of U17 and U26, the output byte will be inverted. Re-inverting it by hardware or by software is up to the user.

**Listing 1**

```
A9  3F       LDAIM  3F
8D  43  17   STA    PBDD
A9  15       LDAIM  15
8D  42  17   STA    SBD    shut off LED
A9  7F       LDAIM  7F     initialize
8D  41  17   STA    PADD   port A to output
AD  xx  xx   LDA    xx     get byte
8D  40  17   STA    SAD    output byte
A9  00       LDAIM  00     initialize
8D  41  17   STA    PADD   port A to input
60           RTS
```

The author may be contacted at Howard Johnson's Motor Lodge, 290 Tarrytown Road, Elmsford, NY 10523.

**Figure 1**

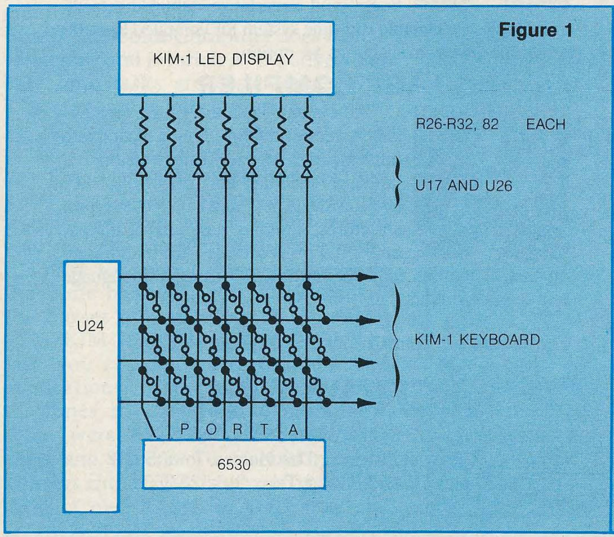
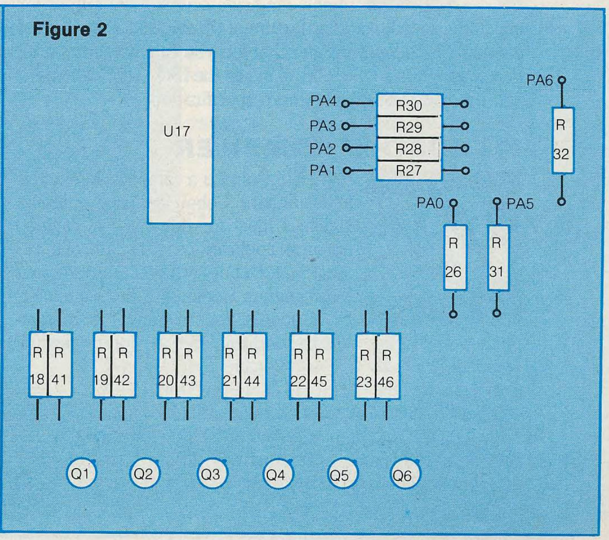
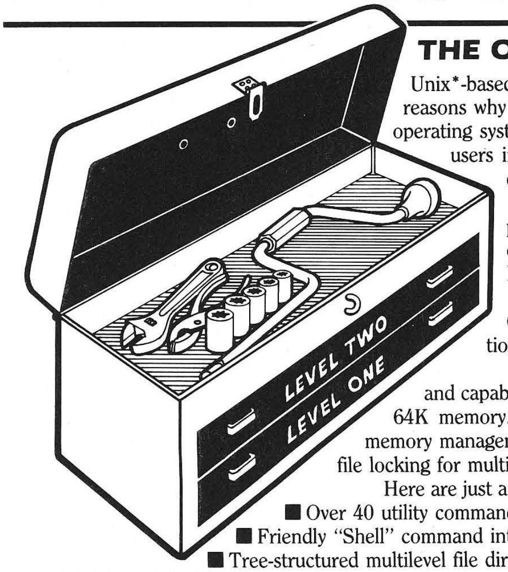


**Figure 2**



# Build performance into your system using the OS-9™ toolbox!



## THE OS-9 OPERATING SYSTEM

Unix*-based, multitasking, modular, and versatile: these key features are some of the reasons why more 6809 computer manufacturers have selected OS-9 as their standard operating system than any other O.S. And OS-9 has been put to work by thousands of users in almost every conceivable computer application in business, science, industry, education, and government.

Your operating system should not be a barrier between you and your computer! OS-9 is very friendly and easy to use. Due to its modular structure it's easy to customize, plus its comprehensive documentation shows you exactly how to interface it to just about any I/O device.

OS-9's advanced features unleash the performance potential of almost any 6809 computer—large or small. In many respects the OS-9/6809 combination is more powerful than many *minicomputers!*

There are two basic versions of OS-9. Both have the same basic features and capabilities. OS-9 Level One runs on small to medium sized systems having up to 64K memory. The Level Two version runs on medium to large size systems having memory management hardware and up to 1 megabyte of memory, and includes record and file locking for multiuser database applications.

Here are just a few reasons why you should insist on OS-9 for your microcomputer system.

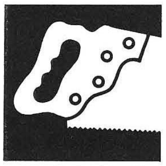
- Over 40 utility commands
- Friendly "Shell" command interpreter
- Tree-structured multilevel file directories
- Full timesharing support with log-in and file security
- Fast, secure random and sequential access files
- Comprehensive English language error messages
- Compact real-time multitasking executive
- Hardware or software memory management
- Device independent interrupt-driven I/O
- Fully ROMable for small control systems
- Standard versions available from manufacturers of most popular 6809 computers

---

## OS-9 PASCAL™ LANGUAGE COMPILER

The OS-9 Pascal language compiler is the most complete and versatile Pascal available for the 6809. OS-9 Pascal has the unusual capability of generating P-code for interpretive execution while debugging OR highly optimized 6809 assembly language source code output for maximum speed. Another feature of OS-9 Pascal is its "virtual memory" P-code interpreter that lets you run incredibly large Pascal programs. OS-9 Pascal meets the ISO 7185.1 Standard and the complete Wirth/Jensen specification.

## CIS COBOL™ COMPILER



6809 CIS Cobol is a compact, interactive and standard Cobol language compiler which is ideal for the most demanding business applications. Standard features are: ISAM, Debug, ACCEPT/DISPLAY, and Interprogram Communications modules. CIS Cobol is *the* preeminent microcomputer Cobol in the industry, and the OS-9 version retains full compatibility with CP/M applications software, CIS Cobol meets the ANSI 1974 Level One COBOL standard and is GSA certified. Also available is Micro Focus' FORMS 2, an optional automatic program generator that lets you interactively design screen-oriented applications with ease.

## BASIC09™ STRUCTURED BASIC INTERACTIVE COMPILER

Basic09 is the fastest and most comprehensive full Basic language available for the 6809. It combines standard Basic with the best features of Pascal. It is a unique interactive compiler that combines compiler speed, interpreter friendliness, and superlative debugging facilities. RunB, a ROMable run-time system for compiled Basic09 programs is now available as an option.

## C LANGUAGE COMPILER

C—the systems language of the future—is here today on OS-9. This is a complete implementation of the Unix Version 7 C language including INT, CHAR, SIGNED, UNSIGNED, FLOAT and LONG data types, structures, unions, standard C library, and a full preprocessor with macro definitions. Generates fully reentrant 6809 assembly language source code output.

For information contact your computer supplier, or

**MICROWARE**®

Microware Systems Corporation
5835 Grand Avenue, Des Moines, Iowa 50312
Phone 515-279-8844 • Telex 910-520-2535

---

*Unix is a trademark of Bell Laboratories. CIS Cobol is a trademark of Micro Focus, Inc. OS-9 and Basic09 are trademarks of Microware and Motorola, Inc.

# Add a VIA and Speech Synthesizer to the Color Computer

*by William C. Clements, Jr.*

**Add two user-accessible, 8-bit bidirectional I/O ports to your color computer and interface an inexpensive speech synthesizer.**

Requires:

TRS-80 Color Computer

The Radio Shack Color Computer is quite a machine for the money, as recent reviews in several magazines have indicated. Moreover, it's an easy and inexpensive way to become acquainted with the MC6809 microprocessor.

Since the Color Computer is still relatively new, the Color devotee must devise many of his own modifications and software. More often than not, he is left to gaze longingly at dozens of articles in the "just what I need except it's for another computer" category.

I began microcomputing on a well-expanded KIM system, and have several KIM-driven peripherals that would work well with the Color Computer. I had also become used to KIM's parallel ports with individually programmable I/O lines and the hardware timer. The Color Computer does have a serial port and joystick A/D converters, but some tasks are easier done with TTL-compatible bidirectional I/O ports.

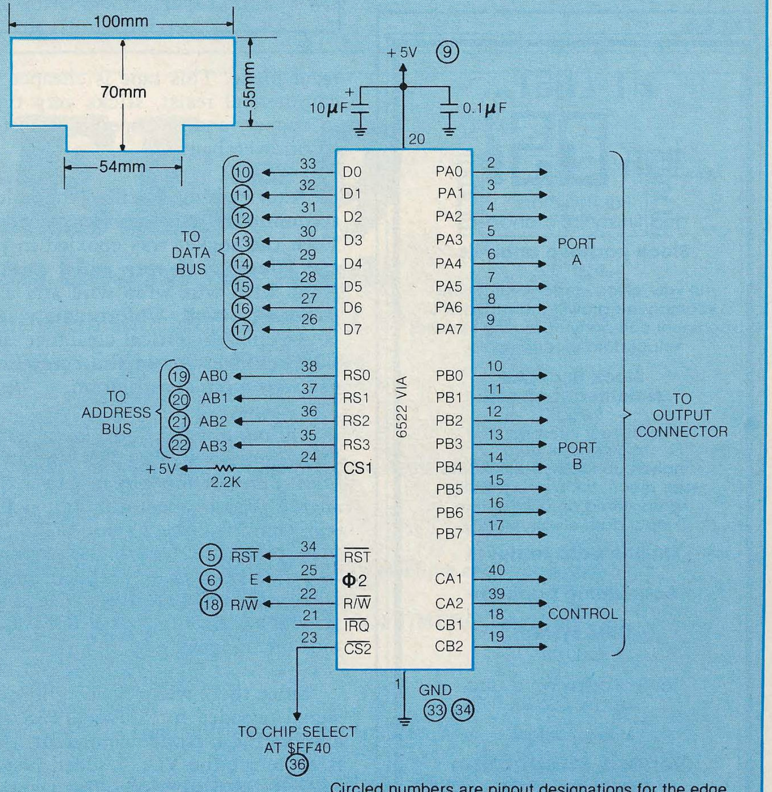
Fortunately, the cartridge connector provides access to data and address busses and all important control signals, as well as to a few handy features like decoded address selects. It's easy to expand the machine with the same hardware you would add to the busses of a member of the AIM/SYM/KIM (ASK) computer family. In fact, you could adapt many published applications for the single-board 6502 machines to the Color Computer if there were 6522- or 6530-style I/O lines and timers. You would need to convert their 6502 driver software into either Color BASIC or 6809 machine code. Then devices such as parallel-data printers, the MTU music synthesis D/A unit, the Optimal Technology Co. EPROM programmer, and the Sweet-Talker speech synthesizer sold by Micromint, Inc., could be used with the Color Computer.

This article shows you how to connect a 6522 Versatile Interface Adapter (VIA) to the Color Computer through its cartridge connector. BASIC driver routines and interface connections for the Sweet-Talker illustrate a simple application for the new I/O ports.

## The VIA Interface Board

The VIA is well known to ASK computer users, thanks to Dr. DeJong's

***Figure 1:* Circuit Diagram and Dimensions for the VIA Board**



Circled numbers are pinout designations for the edge connector in the Color Computer's cartridge slot. The non-circled numbers are I.C. pinouts.

MICRO articles [1, 2] and his excellent book [3], and by its use in the SYM and AIM machines. Adding the VIA to the Color Computer provides two 8-bit parallel I/O ports with individually programmable bits, two interval timers with several different modes of use, and parallel-serial data interconversion (if you need more capability than the Color Computer's serial port can provide). These items, plus the serial output, the cassette interface, and the two joystick A/D channels already on the machine, give you quite a versatile system.

Figure 1 is a simple schematic diagram of the 6522 board. You will have to provide edge fingers to plug into the female edge-connector in the cartridge compartment. I took a piece of un-etched double-sided circuit board stock, cut it to the approximate dimensions of the printed circuit board inside the Tandy cartridges (again see figure 1), and laid down 40 strips of 1.5 mm wide tape, 20 per side with 0.1 inch spacing, on the edge. A couple of wider strips at the board sides were put down to make +5V and ground busses. I made my own resist tape by slitting a piece of Scotch Filament tape with a razor blade. This tape is cheaper than commercial resist, sticks very tightly to copper, and is impervious to ferric chloride etchant.

**Listing 1**

```
1 DATA57,9,31,62
2 DATA9,18,31,62
3 DATA41,52,52,43,62
4 DATA25,51,24,49,43,62
5 DATA25,51,12,37,27,54,42,58,62
6 DATA31,37,44,25,39,20,62
8 A=65344
10 DIMZ(100)
20 POKEA+3,255:POKEA+2,6
30 P=63:GOSUB1000
40 FORN=1TO35:READZ(N):NEXTN
50 N=N-1
60 FORI=1TON:P=Z(I):GOSUB1000
70 NEXTI:STOP
1000 POKEA,0:POKEA+1,P:POKEA,2
1010 IF (PEEK(A) AND 1) THEN RETURN ELSE 1010
```

**Listing 2**

```
1 'VIA AT $FF40
5 A=65344:N=0
10 DIMZ(100)
15 'PORT A = OUTPUTS: PB0 = INPUT, PB1,PB2 = OUTPUTS
20 POKEA+3,255:POKEA+2,6
25 'SEND STOP-PHONEME & SILENCE SYNTHESIZER
30 P=63:GOSUB1000
33 'INPUT PHONEME CODES
36 'WHEN READY TO HEAR THEM, INPUT ANY NEGATIVE NO.
40 N=N+1:INPUTZ(N):IFZ(N)=>0 THEN 40
50 N=N-1
55 'SEND PHONEME CODES OUT TO SYNTHESIZER
60 FORI=1TON:P=Z(I):GOSUB1000
70 NEXTI:N=0:GOTO30
999 'TAKE ENABLE AND STB LOW
1000 POKEA,0
1005 'PLACE PHONEME CODE ONTO PORT A
1010 POKEA+1,P
1015 'TAKE STB HIGH AGAIN
1020 POKEA,2
1025 'RETURN WHEN A/R GOES HIGH
1030 IF (PEEK(A) AND 1) THEN RETURN ELSE 1030
```

After etching, I drilled the holes to accommodate a low-profile 6522 socket. Then the socket was epoxied in place and the circuit was wired point-to-point with wire-wrap wire and a fine soldering pencil. Unfortunately, there is not enough vertical clearance inside the cartridge slot of the computer to use wire-wrap construction.

On my unit, the port lines, control lines, and the +5V, +12V, and ground lines were all brought out to a right-angle DB-25 connector mounted at the rear center of the circuit board. You may prefer to use a less expensive 24-pin DIP socket or a ribbon cable instead. I didn't plan to use timer-generated interrupts, so I left the IRQ pin unconnected.

Since there was a handy chip-select signal decoded for $FF40-$FF5F available on the cartridge connector, I used it to select the VIA. I added just one 6522, so no further decoding was done. This address space will hold two VIAs if you want to add some logic to distinguish between the addresses $FF40 and $FF50. In my case, register 0 of the VIA is addressed at $FF40 with address image at $FF50, register 1 at $FF41 with image at $FF51, and so forth.

For protection and insulation, the circuit board was enclosed in a homemade plastic box about the size of a Tandy cartridge. A slot was cut to allow the contact fingers to protrude through the front of the box. I also cut a hole for the DB-25 output connector in the rear.

## Now Make It Talk

As a simple and entertaining example of using the 6522 to interface a non-Radio Shack peripheral, let's hook up the Sweet-Talker speech synthesizer recently described in a construction article by Ciarcia[4] and also marketed as an assembled unit by Micromint, Inc. This device, based on the Votrax SC-01 synthesizer chip, is simple to build and use. Speech is generated by cascading phonemes together; there are 64 generated by the SC-01. These phonemes are called out by placing a 6-bit phoneme code onto the Sweet-Talker's data lines (P0-P5) and strobing the synthesizer. Two pitch-control lines, I1 and I2, are available to vary phoneme inflection; their use is optional.

The data lines (P0-P5) and inflection bits (I1 and I2) together comprise an 8-bit speech-configuration word that is provided in my setup through Port A of the VIA, P0-P5 being driven by PA0-PA5 and I1, I2 by PA6 and PA7. In addition, three control lines must be interfaced. The synthesizer signal A/R requires an input line and is connected to PB0. The STB and enable signals of the Sweet-Talker are driven as outputs from the computer and are connected to PB1 and PB2, respectively. The power requirements are +5 and +12 volts at current levels easily handled by the Color Computer's power supply.

The Sweet-Talker synthesizer must be driven as follows (see reference 1 for detailed explanation and exact definition of the control signals): initialize the synthesizer by driving STB and enable both low. Place 6-bit phoneme code onto P0-P5 and 2-bit inflection code (if used) on I1 and I2. Latch data into unit by applying rising edge of positive-going pulse to STB. Data must be stable 450 ns prior to rising edge of STB, and STB must have been low at least 100 μs before it returns high. About 500 ns after rising edge of STB, A/R goes low, indicating that the synthesizer chip is busy. When A/R is found again to be high, the chip is ready to accept another phoneme code.

These functions can easily be implemented using a short BASIC or machine-language program. To find out how the synthesizer sounds, attach it to the VIA as indicated above, then enter and run the program in listing 1.

The program in listing 2 lets you experiment with the synthesizer by inputting a series of phoneme codes and then listening to the speech they produce. With the phoneme table in Ciarcia's article and this program, you can generate just about any speech you want. Don't forget to convert the hex codes given in the article into decimal before entering them.

In this article, I have attempted to show how easy it is to connect parallel-mode peripherals to the Color Computer, and to suggest that Color Computer users can now make use of the extensive literature covering 6502-based computer peripherals. If I have lessened even a little the frustration of the Color owner in seeing so many goodies out there for other machines, then I will rate this effort a success.

## Bibliography

1. De Jong, M.L., "6522 Timing and Counting Techniques," MICRO 17, pp. 27-39.
2. De Jong, M.L., "AIM 6522 Based Frequency Counter," in *The Best of MICRO*, MICRO INK, Inc., Chelmsford, MA, 1980, pp. 10-11.
3. De Jong, M.L., *Programming and Interfacing the 6502, With Experiments*, Howard W. Sams & Co., Inc., Indianapolis, IN, 1980.
4. Ciarcia, Steve, "Build An Unlimited Vocabulary Speech Synthesizer," *BYTE* 6, No. 9, pp. 38-50.

---

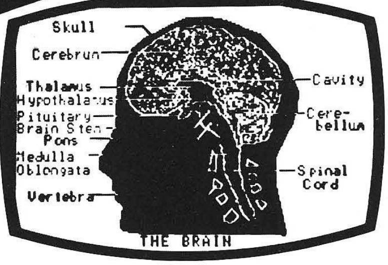
W.C. Clements, Jr. is professor of chemical engineering at the University of Alabama. He designed and built several 6502- and 6801-based machines and has developed software for them. Presently, he is designing peripherals and programs for his department's seven color computers. Write to him at the University of Alabama, P.O. Box 2662, University, AL 35486.

MICRO

# 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**

MICRO™

# The Single Life

By Brad Rinehart

## More About Disk BASIC

Last month we discussed the sequential and snapshot file structures. This month I want to teach you the ins and outs of RANDOM files.

The definition of a random file denotes that it takes the same amount of time to access the last record in the file as it takes to access the first record. To access the last record of a sequential file, all preceding records must first be read.

To date, the most widely used method for determining where a record is located in a random file is to define a fixed record length (in bytes). Then you index the number of bytes times the record number into the file.

HDE disk BASIC defines this record length as 128 bytes, which is the length of one sector. The size of the random file is limited only by the amount of disk space. On an 8'' floppy this means about 1975 records.

In HDE Disk BASIC, random files may be either open ended or reserved. If the file is open ended, it may continue to grow until it runs out of disk space. By reserving the file with the RSV command, a unique area of the disk, X number of sectors, may be assigned to the file. I prefer to reserve the file. In figure 1 I illustrate three random files. The first contains ten records, the second eight records, and the third three records.

Let's assume we opened FILE 1, wrote ten records, and then closed it. Next we opened FILE 2, reserved it for eight records, then opened FILE 3 and reserved it for three records. If we try to write record nine to FILE 2 or record four to FILE 3, BASIC will return an error because we tried to write past the end of the reserved area. However, since we did not reserve FILE 1, we could open it and write records 11 and 12 to the file. The shaded area shows where the data would be put. As you can see, we would overwrite information in FILE 2 which could cause some drastic problems. Therefore, reserve the file for the maximum number of records you expect to write.

To review the procedure thus far, we have opened, reserved, and closed files. The instructions for this procedure would look like:

```
10 FILES3
20 OPEN "R", 1, "FILE 1"
30 RSV1, 50
40 CLOSE 1
```

where the FILES command declares that we expect a maximum of three files to be OPEN at one time. This reserves space in memory for three 256-byte file buffers. FILES must be declared *before* any variables are defined.

The OPEN command tells BASIC we wish to access FILE 1. In addition, the "R" denotes that the file is to be opened for RANDOM access. The "1" in the OPEN command tells BASIC we wish to use file buffer 1. HDE BASIC allows up to 32 files to be open *at the same time.* While a file buffer is OPEN, it will access the file named in the OPEN statement. However, if the file is CLOSEd, it may be reopened and used to access another named file, as in:

```
50 CLOSE 1
60 OPEN "R", 1, "FILE 2"
```

Since we declared three files, all three may be open at the same time.

The statements:

```
100 OPEN "R", 1, "FILE 1"
110 OPEN "R", 2, "FILE 2"
120 OPEN "R", 3, "FILE 3"
```

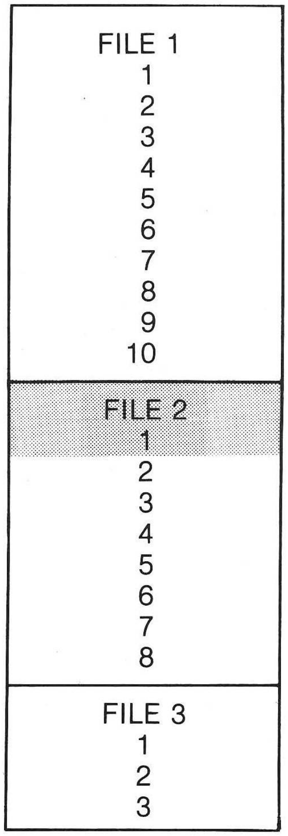
will open all three files and allow access to any one or all three of them. Several files may be CLOSEd simultaneously by the command CLOSE 1, 2, 3.

I mentioned that random files should be reserved. The reserve or RSV command also provides for initializing an entire file to a pre-defined value. For example the statement:

```
RSV1, 50, CHR$(32)
```

will reserve file number one for fifty records and also write a space, CHR$(32), in every byte in the file. Any one-byte character may be specified as the fill character.

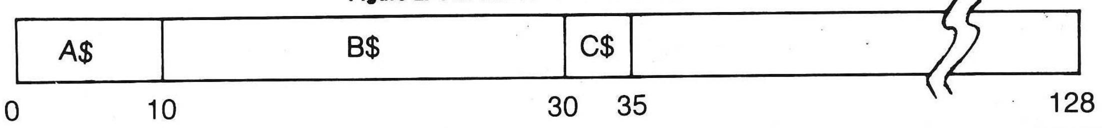
***Figure 1:* Random Access Disk Files**



After a random file has been opened, the next step is to read and write information from and to it. I stated that HDE BASIC assigns a fixed record length of 128 bytes to each record in the file. This does not mean that every piece of data will use 128 bytes on the disk. A random file record is designed to be fielded, or broken down into one or more smaller fields. A field may consist of as little as one byte, or as many as 128 bytes of information. Fields may overlap, meaning that FIELD A$ may consist of all or part of FIELD B$, FIELD C$, FIELD D$, etc.

Referring to figure 2, note fields A$, B$, and C$. Field A$ has a length of 10 bytes, B$ a length of 20 bytes, and C$ a length of five bytes. The FIELD command is used to define the particular

**Figure 2: One Random Access Record**



fields and their lengths. Figure 2's fields were created by the statement:

```
FIELD 1, 10 AS A$, 20 AS B$, 5 AS C$
```

where the '1' following the field command defines which file buffer we wish to field.

If we execute a second FIELD statement such as:

```
FIELD 1, 16 AS D$
```

we end up with the record depicted in figure 3. Note that the field D$ contains all of field A$ and part of field B$. However, the previously defined fields, A$, B$, and C$, remain as originally defined. Hence, the fields overlap.

Any string variable may be used in the FIELD statement. This includes string array variables as in A$(X). Hence, every byte of a record may be fielded as an array element. The code might look like:

```
100 DIM A$ (128)
110 FOR X=1 TO 128
120 FIELD 1, X-  AS D$, 1 AS A$(X)
130 NEXT
```

where the string variable D$ is used as a filler.

Some explanation of the operation of the FIELD statement is in order here. BASIC knows where variables are located by means of a variable pointer or table. The FIELD statement simply fills the proper address and variable length into this table. When the statement 'FIELD 1' is encountered, BASIC puts the address of the beginning of FILE BUFFER 1 into a temporary location. Each time the 'AS' statement is encountered, the address in this temporary location is stored into the proper variable pointer and then the temporary location is incremented by the value *preceding* the 'AS' statement. Hence, if the address of FILE BUFFER 1 is $DE00, the statement:

```
FIELD 1, 3 AS A$
```

will perform the following operations:

1. Store $DE00 into 'TEMP'
2. Store contents of 'TEMP' into 'pointer to variable A$'
3. Store length of '3' into 'length of variable A$'
4. Add '3' to contents of 'TEMP'

After the operation is complete, 'TEMP' will contain $DE03.

From our FOR-NEXT loop example, you can see that the purpose of the filler variable D$ is merely to 'PUSH' the contents of 'TEMP'. to the next location in the file buffer with each iteration of the loop. Note that in the first pass through the loop, the length of D$ is zero. Hence, the location of A$(1) will be the beginning of the file buffer. With each additional pass, the length of the filler, D$, is increased by one. Therefore, on the 128th pass through the loop, the first 127 bytes will be fielded as D$, while the last byte will be fielded as A$(128) with a length of one. BASIC will return an error if you try to field past the end of the file buffer.

String data with a random file is accessed differently than strings created by other means (F$ = "TEST STRING", etc.). To put data into a file string, HDE has provided two commands, LSET and RSET. The syntax for either command is:

```
LSET A$ = "TEST STRING" or
RSET A$ = "TEST STRING"
```

These commands cause the data specified on the right hand side of the "=" sign to be stored into the field specified by the variable following the command. In addition, they cause the unused portion of the field to be filled with spaces!

The difference between LSET and RSET is that LSET stores the data beginning on the left side of the field and pads to the right, while RSET stores the data on the right side of the field and pads to the left.

| | |
|---|---|
| LSET | TEST STRINGb̸b̸b̸b̸ |
| RSET | b̸b̸b̸b̸TEST STRING |

If the length of the data is greater than the field size, the data is truncated on the right. The data is always truncated on the right, whether the command is LSET or RSET.

| | |
|---|---|
| TEST STR1 | LSET or RSET TRUNCATION |

To read data from the file buffer, use the normal BASIC string functions:

```
PRINT A$(1)
F$ = A$(128)
D = VAL(B$)
etc.
```

*Note:* Never reference a fielded variable unless you are working with the file! For example, the statement:

```
A$(1) = F$
```

will destroy the relationship between A$(1) and the file buffer. The statement:

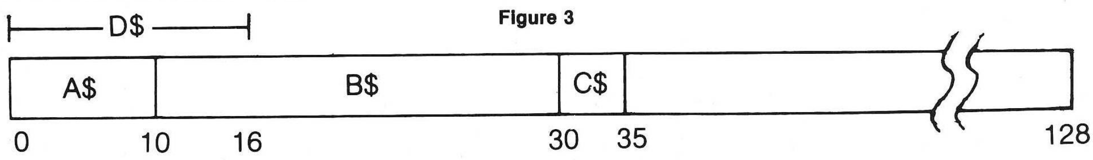
```
LSET A$(1) = F$
```

will not.

So far we have only manipulated file data in memory. We *have not* caused any data to be written to or read from the disk. Two commands, PUT and GET, are used to write a record to the disk or read a record from it. The statement:

```
GET 1, 5
```

**Figure 3**



will cause record number 5 (128 bytes) of the file to be read into file buffer one.

The statement:

```
PUT 1, 5
```

will cause file buffer one (128 bytes) to be written to record number 5 in the file. Unlike sequential files, the random file buffer is not automatically written to the disk when the file is CLOSEd. You are responsible for PUTing the record to the file.

HDE has provided two very useful record pointers, LRN and NRN. These two functions first return the Last Record Number and then the Next Record Number of the specified file.

LRN refers to the *highest* record number accessed in the file. For example, if a file has been reserved for 100 records, but only the first 50 have been accessed, the LRN of the file will be 50.

NRN equates to a value that is one higher than the record currently in the file buffer. Consider the previous example. If we access record number 5 *via* the GET 1, 5 statement, (NRN1) will return a value of six, while (LRN1) will still equal 50.

In the following program, we will OPEN a new file, FIELD it, access some records, and retrieve the values of NRN and LRN.

```
 10 FILES 1
 20 OPEN "R", 1, "NWFIL"
 30 X=(NRN1) : Y=(LRN1)
       REM X=1, Y=0
 40 RSV1, 100
 50 X=(NRN1) : Y=(LRN1)
       REM X=1, Y=0
 60 FIELD 1, 64 AS A$, 64 AS B$
 70 GET 1, 25
 80 X=(NRN1) : Y=(LRN1)
       REM X=26, Y=25
 90 GET 1, 10
100 X=(NRN1) : Y=(LRN1)
       REM X=11, Y=25
110 CLOSE 1
120 OPEN "R", 1, "NWFIL"
130 X=(NRN1) : Y=(LRN1)
       REM X= 1, Y=25
```

Notice that when the file was CLOSEd and then reOPENed, LRN was preserved, while NRN was reset to one.

The value of LRN may be reset to zero *via* the RESTORE# (note the pound sign) command. Referencing the previous program, if we add line 140 to say:

```
140 RESTORE#1
```

the value of LRN will equal zero and the value of NRN will equal one.

NRN and LRN may be used to track the size of a file. This eliminates the need for additional variables. To sequentially read ten records from a random file, NRN may be used as the record number variable.

```
200 FOR X=1 TO 10
210 GET 1, (NRN1)
220 C$(X)=A$ : D$(X)=B$
230 NEXT
```

The purpose of the array variables C$(X) and D$(X) is to move the data out of the file buffer *before* the next record is read. If this is not done, the next record will overwrite the previous one.

If the intent of reading the file is merely to PRINT the contents of the file, we can substitute:

```
220 PRINT A$ : PRINT B$
```

for line 220 in the example. This will print each record and then overwrite the data with the next record. Hence, we can read an entire random access file, print it, and only use 128 bytes of variable space. No problem with unexpected garbage collections here!

# A Sequential File Handler for a Disk-Based OSI

*by Mark Johnson and Chris Johnson*

**An easily implemented sequential file handler for the C1P, written in BASIC, that may be adapted to the OSI C2 and C3 under OS65D3.2. Structured programming is used to provide a control program that calls subroutines for speed and efficiency.**

These routines require:
OSI C1P
DOS OS65D3.2
They may be modified for C2 and C3 models.

The most significant criteria distinguishing professional machines from toys is the ability to handle files. OSI offers OS65U operating system for its professional machines, but the operating system is not available for the C1P. OS65D, the operating system for the C1P, supposedly has file handling capabilities, but is severely limited in several ways. The search routine employed by OS65D to look for the beginning of a new record is very inefficient and slow. In OS65D, files are placed in fixed-length records of 128 bytes. Even a record of only one byte takes up 128 bytes on the disk. This system is very inefficient for short records and also severely limits the use of long records.

OS65D doesn't keep track of the number of records used in a file. A program counter must be implemented to count records from the beginning of the file until an End-Of-File marker is read. Large amounts of time are consumed every time the counter program is used. Also the OS65U on the C1P consumes 22K of RAM and costs $200.00. Machine-language subroutines are difficult to access from BASIC, and are not easily modified.

The requirements for a good file handler established the following criteria:

1. The file handler must be written entirely in BASIC.
2. It must be fast in operation.
3. Variable length records must be allowed.
4. It must be easy to use.
5. The SAVE and LOAD functions should be written as subroutines so that they may be easily accessed from the calling program.

The program evolved into a General Initialization Routine (listing 2), a SAVE and LOAD Initialization Routine (listing 3), a SAVE Routine (listing 4), and a LOAD Routine (listing 5). Listing 1 is a sample control program to show how a typical calling program would operate, and is the basis of a simple text editor. Listing 6 is a sample run of the program.

## General Initialization Routine (Listing 2)

The REM statements, lines 60000 through 60090, make the program self-explanatory and may be removed to conserve memory. Variable PK is set to 2K less than your memory size to allow 2K workspace at the top of memory. The POKE statements on lines 60110 and 60120 set BASIC's top of memory to the new memory size. The variable REC is used as a record counter and is initialized by line 60130. Note that the General Initialization Routine is a subroutine called by the control program each time a new track is to be initialized.

## SAVE and LOAD Initialization Routines (Listing 3)

The SAVE initialization is contained in lines 61000 through 61040 and is called once by the control program when a SAVE is made. It is used to write header information that the track has been used by the program. Line 61020 initializes the REC counter, and line 61040 advances the counter to the correct count to start adding records to the track.

The LOAD initialization is contained in lines 63000 through 63020 and is called once by the control program when a LOAD is made. Line 63020 sets a data pointer to the header on the track.

## SAVE Routine (Listing 4)

Lines 62000 to 62250 comprise the SAVE routine. It is a subroutine that is called by the control program each time a line of data (up to 254 bytes) is to be saved on disk. Line 62050 increments the record counter, RC. Lines 62060 through 62205 detect a full track condition and allow data to be written on the next succeeding track, if desired. If "NO" is selected as an option to writing on the next track, then all data are killed. Lines 62210 through 62250 save the data into the track buffer, write a "0" as the last byte as an End-Of-Record flag, and reset the REC location pointer to the proper record number.

Lines 63500 through 63570 comprise the LOAD subroutine. It is called by the control program each time a record is to be loaded from the track buffer into string RC$. Lines 63500 through 63530 increment the record counter, RC, and check to see if the value of RC exceeds the number of records on the track. If so, RC$ is set to a null string and return is made to the control program. If the character is not an End-Of-Record symbol, return is then made to the control program with RC$ holding a line of data.

## Control Program (Listing 1)

The sample control program shown in lines 10 through 440 is included to show the function of the previously described subroutines. It may be modified as desired. The subroutines are universal for almost any file-manipulation technique. The control

### *Listing 1:* Control Program

```
10 GOSUB60000:PRINT"MENU:":GOSUB130:RC=0
20 PRINT"  S) SAVE DATA ONTO A         TRACK,"
30 PRINT"  L) LOAD DATA FROM A         TRACK,"
40 PRINT"  Q) QUIT."
50 POKE2797,58:POKE9682,161:POKE2888,0:POKE8722,0:
               POKE2976,13
60 POKE2972,13
70 INPUT"CHOICE";C$:C$=LEFT$(C$,1)
80 IFC$="S"THEN190
90 IFC$="L"THEN290
100  IFC$="Q"THENPOKE2972,44:POKE2976,58:END
110  PRINT"INVALID OPTION."
120  GOTO70
130  P1=INT(PK/4096)
140  P2=INT((PK-(P1*4096))/256)
150  P3=INT((PK-((P1*4096)+(P2*256)))/16)
160  P4=PK-((P1*4096)+(P2*256)+(P3*16))
170  P$=CHR$(48+P1):IFP2>9THENP$=P$+CHR$(55+P2):
                   GOTO172
171  P$=P$+CHR$(48+P2)
172  IFP3>9THENP$=P$+CHR$(55+P3):GOTO174
173  P$=P$+CHR$(48+P3)
174  IFP4>9THENP$=P$+CHR$(55+P4):GOTO176
175  P$=P$+CHR$(48+P4)
176  RETURN
180  REM SAVE & LOAD CONTROL ROUTINES
190  GOSUB260:GOSUB61000
200  PRINT"ENTER DATA:"
210  INPUTRC$
220  IFRC$=""THEN250
230  GOSUB62000
240  GOTO210
250  DISK!"SA "+TR$+",1="+P$+"/8":GOTO10
260  INPUT"TRACK TO BE USED";TR$
270  IFVAL(TR$)<16ORVAL(TR$)>39THEN260
280  RETURN
290  GOSUB260:GOSUB63000
300  DISK!"CA "+P$+"="+TR$+",1"
310  IFPEEK(PK)=161THEN340
320  PRINT"**TRACK NOT FORMATTED      (NOT USED)."
330  GOTO10
340  IFPEEK(PK+1)>0THEN370
350  PRINT"**TRACK EMPTY."
360  GOTO10
370  PK=PK+2
380  PRINT"TRACK:"TR$
390  FORX=1TOPEEK(REC)
400  GOSUB63500
410  PRINTRC;TAB(5);RC$
420  NEXTX
430  PRINT"**END OF DATA."
440  GOTO10
```

### *Listing 2:* General Initialization Routine

```
60000 REM SET UP MEMORY SIZE
60010 REM AND VARIABLE 'PK'.
60020 REM IF YOU HAVE A
60030 REM DIFFERENT MEMORY
60040 REM SIZE THAN 32K,
60050 REM CHANGE THE NUMBER 'PK'
60060 REM TO YOUR MEMORY
60070 REM (EX:16K), THEN
60080 REM SUBTRACT 2K
60090 REM (EX:14K).
60100 PK=30*1024
60110 POKE133,INT(PK/256)
60120 POKE132,PK-(INT(PK/256)*256)
60130 REC=PK+2
60140 RETURN
```

### *Listing 3:* SAVE and LOAD Initialization Routines

```
61000 REM SAVE INITIALIZATION
61010 REM SUBROUTINE
61020 POKEPK+1,161
61030 POKEPK+2,0
61040 PK=PK+3:RETURN
63000 REM LOAD INITIALIZATION
63010 REM SUBROUTINE
63020 PK=PK+1:RETURN
```

### *Listing 4:* SAVE Routine

```
62000 REM SAVE ONE LINE
62010 REM OF STRING ONTO
62020 REM TRACK BUFFER,
62030 REM UPDATE RECORD
62040 REM COUNTER.
62050 RC=RC+1
62060 IFRC<256THEN62210
62070 PRINT"**YOU HAVE OVER 255
               LINES ON THIS TRACK."
62080 PRINT"  DO YOU WISH TO SAVE
              THIS TRACK AND"
62090 PRINT"  CONTINUE ON THE NEXT ONE";
62100 POKE2797,63
62110 INPUTA$:POKE2797,58
62120 IFLEFT$(A$,1)="Y"THEN62150
62130 GOTO10
62140 REM READY NEXT TRACK FOR INFORMATION
62150 DISK!"SA "+TR$+",1="+P$+"/8"
62160 TR=VAL(TR$)+1:IFTR>39THEN62200
62170 TR$=RIGHT$(STR$(TR),2)
62180 GOSUB60000:GOSUB61000
62190 GOTO62210
62200 PRINT"**YOU'RE AT THE END OF
              THE DISK, THE LINE"
62205 PRINT"  YOU HAVE JUST TYPED
              IS LOST, SORRY!!":END
62210 FORX=1TOLEN(RC$)
62220 POKEX+PK-1,ASC(MID$(RC$,X,1))
62230 NEXTX:PK=PK+X
62240 POKEPK-1,0:POKEREC,RC
62250 RETURN
```

program is all that needs to be changed to develop a field-oriented data base management system or a full text editor. These control programs will be presented in following issues.

Lines 10 through 40 call the General Initialization Routine, translate the decimal memory size to hexadecimal for use by OS65D subroutines, initialize record counter RC, and list a menu of choices. Lines 50 through 60 remove some limitations on BASIC that prevent use of commas and colons in text and allow a carriage return as a valid input. (See the listing of POKE values for a detailed description.) Lines 70 through 120 direct program control to the proper subroutine, depending on the menu choice.

Lines 180 through 280 form a control program for the SAVE function. It asks for the desired track number to start data storage. Note that tracks 1 through 15 are blocked because this is where OS65D and the DOS reside. Tracks greater than 39 are also blocked because the C1P drive is a 40-track drive. These restrictions may be altered by changing the parameters in line 270. The SAVE initialization and SAVE subroutines are called by the control block.

Lines 290 through 440 form the LOAD control program. It asks for the desired track number to start data retrieval from disk. The track number restrictions indicated in the SAVE control program also apply. These restrictions are automatically altered if the SAVE control program parameters in line 270 are changed. The LOAD control program loads the desired track into the track formatting. The track is next checked to see whether or not it contains data. Each record is printed, one at a time, along with its record number.

## Sample Run (Listing 6)

The MENU is printed and choices are given to:

S) SAVE data onto a track,
L) LOAD data from a track,
Q) QUIT (Exit to BASIC).

If a SAVE or LOAD is selected, the user is asked for the desired track number. The sample run shows selection of SAVE, entry of data, selection of LOAD, retrieval of the data, and exit to BASIC.

## POKE Locations and Their Purposes

POKE 2797, 58 changes the question mark used on INPUT to a colon.

POKE 9682, 161 changes the cursor symbol to a block.

POKE 2888, 0 and POKE 8722, 0 must be made in conjunction to allow null inputs (carriage return only) without automatic exit into BASIC.

POKE 2976, 13 allows entry of a comma as part of an INPUT.

POKE 2972, 13 allows entry of a colon as part of an INPUT.

POKE 133,INT(PK/256) and POKE 132, PK − (INT(PK/256)*256) are the high and low bytes defining the top of memory.

## Variables

| | |
|---|---|
| RC | Record counter — used to keep track of present record number. |
| C$ | Temporary input storage. |
| P1-P4 | Temporary storage for decimal/hexadecimal conversion. |
| P$ | Hexadecimal value of beginning of track storage area. |
| RC$ | Temporary storage area for data in a record. |
| TR$ | Selected track number. |
| PK | Pointer to present location in track storage area. |
| X | Iteration counter. |
| REC | Storage location for total number of records in use. |
| A$ | Temporary input storage. |
| TR | Temporary track number storage. |
| P | Temporary storage present character. |

### *Listing 5:* LOAD Routine

```
63500 REM LOAD ONE LINE INTO
63510 REM STRING ´RC$´.
63520 RC=RC+1:RC$=""
63530 IFRC>PEEK(REC)
      THENRC$="":RETURN
63540 P=PEEK(PK)
63550 IFP=0THENPK=PK+1:RETURN
63560 RC$=RC$+CHR$(P)
63570 PK=PK+1:GOTO63540
```

### *Listing 6:* Sample Run

```
MENU:
S> SAVE DATA ONTO A TRACK.
L> LOAD DATA FROM A TRACK.
Q> QUIT.
CHOICE: S
TRACK TO BE USED: 32
ENTER DATA:
: NOW IS THE TIME FOR
: ALL GOOD MEN TO COME
: TO THE AID OF THEIR
: COUNTRY.
:
MENU:
S> SAVE DATA ONTO A TRACK.
L> LOAD DATA FROM A TRACK.
Q> QUIT.
CHOICE: L
TRACK TO BE USED: 32
TRACK:32
1     NOW IS THE TIME FOR
2     ALL GOOD MEN TO COME
3     TO THE AID OF THEIR
4     COUNTRY.
**END OF DATA.
MENU:
S> SAVE DATA ONTO A TRACK.
L> LOAD DATA FROM A TRACK.
Q> QUIT.
CHOICE: Q

OK
```

---

Chris Johnson is employed by Westinghouse Electric Corporation as an applications engineer for a line of microprocessor-based programmable controllers. He learned FORTRAN in 1960, but currently uses BASIC and 6502 machine language. He has been involved in solid state design since 1956.

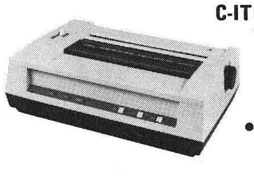
Mark Johnson is sixteen years old and has been writing BASIC programs for the past five years. He is conversant in 6502 assembler, COBOL, FORTRAN, ALGOL, and Pascal.

Mark and Chris work as a team, with Mark specializing in software and Chris specializing in hardware. They currently have a homebrew, an extensively modified OSI C1P, and an Atari. They are presently investigating high-resolution color graphics and a new homebrew — based on either the 6809 or the 68000. They may be contacted at 7204 S. Yarrow St., Littleton, CO 80123.

# 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**

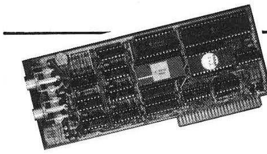
**NEW! U-Z-80 PROCESSOR BOARD (from Europe)** Software compatible with Softcard and ALS Software .............. (List: $199) .... **$149**

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 decenders.

| | |
|---|---|
| **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.

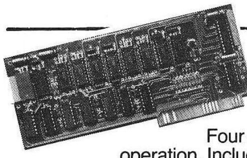
(List: $49) ....... **$35**

**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**

# Skyles Electric Works

### BASIC Programmer's Command-O™

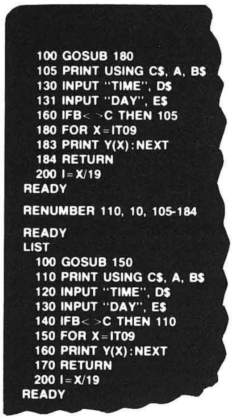
# For PET/CBM™ Owners Who Want More Fun And Fewer Errors with Their Programs

Here are nineteen commands you'll need, on a single chip you can install in two minutes without tools, **on any PET or CBM 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.

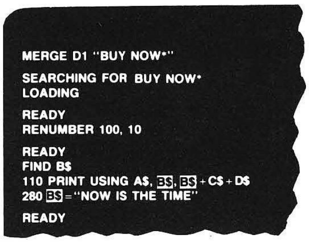
For PET 4016, 4032, CBM 8032, SuperPET or BASIC 4.0 upgrade

### BASIC Programmers Command-O™

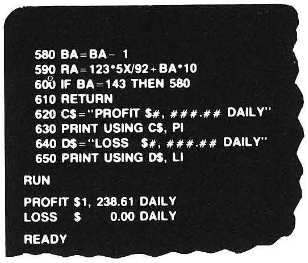
**AUTO DUMP DELETE FIND (improved) HELP KILL OFF TRACE (improved) RENUMBER (improved) INITIALIZE MERGE MOVE EXECUTE SCROLL OUT SET SEND PRINT USING BEEP**



```
  100 GOSUB 180
  105 PRINT USING C$, A, B$
  130 INPUT "TIME", D$
  131 INPUT "DAY", E$
  160 IFB< >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 IFB< >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$, B$, 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 $#, ###.## DAILY"
  630 PRINT USING C$, PI
  640 D$="LOSS  $#, ###.## DAILY"
  650 PRINT USING D$, LI

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**

231E South Whisman Road
Mountain View, CA 94041
(415) 965-1735

# Face Synthesizer for PET

by David Heise

**This program creates an animated face on a PET screen that changes expression on keyboard command. Animation is controlled from BASIC programs, so the PET face can be used in any application — education, marketing, games, etc.**

**Face Synthesizer**
requires:
PET
8K or larger
3.0 or 4.0 Operating System

Facial expressions reflect emotions. This well-known fact has been a topic of study for social psychologists, who analyze expressions by observing the shape and position of eyes, mouth, eyebrows, and other facial features. The psychologists' precise technical descriptions of these features provided me with the data I needed to represent facial expressions on a computer. Now your PET can express emotions too.

This program generates a face that smiles, winks, and pouts; shows fear, disgust, or anger; or widens its eyes in surprise. You operate the face from the keyboard. The facial expressions also can be called from BASIC programs for practical applications. A salesroom display program could use the PET face to call attention to a product with moving eyes, winks, and smiles. Education programs could use the face to provide rewards or reproofs for right or wrong answers.

### What Are Facial Expressions?

Emotional messages are constructed on the face by the shape of the mouth, eyes, and eyebrows (and sometimes the nose, cheeks, and forehead as well). Each of these features has a limited number of major shapes produced by the action of certain facial muscles. Whether a group of muscles is tugging gently or straining hard may suggest the intensity of feeling, but the real information is in the fact that certain muscles are operative, producing the characteristic shape for that muscle group.

The brows have four major shapes other than a neutral relaxed position. They may be curved upward (as in surprise), flattened and raised (as in fear), flattened and lowered (as in sadness), or pulled down and inward (as in anger).

The opened eyes have six major shapes: neutral, wide open (as in surprise), raised lower lids (as in disgust), raised and tensed lower lids (as in fear), squinting (as in anger), and upper lids drooping and sloped (as in sadness).

Major shapes of the mouth, aside from neutral, are: dropped open (as in surprise), corners pulled horizontal (as in fear), lips pressed tight (as in anger), squared outthrust lips baring teeth (as in anger), upper lip pulled up (as in disgust), corners down (as in sadness), corners raised (as in happiness, with extra stretching for smiles, grins, or laughs).

The end of the nose may be normal or raised by pressure from the upper lip; the upper nose may be normal or crinkled. Cheeks may be normal or raised during laughter. The forehead may be normal or wrinkled by pressures from the eyebrows.

Variations in one feature combine with variations in another feature; for example, any eyebrow formation can occur with any mouth shape. But not quite every combination of features is possible. For example, the mouth isn't disgusted alone; "disgusted" mouth occurs with nose raised.

Expressions for the primary emotions are universal. Surprise combines arched eyebrows with wide open eyes and a dropped open mouth. Fear shows in raised and flattened eyebrows, raised and tensed lower eyelids, along with sidestretched lips. Disgust involves raised lower eyelids, and the upper lip curled up so to raise the nose; the upper nose may be crinkled. In anger the brows pull down and inward, the eyes squint, and the lips either are pressed tight or squared into a snarl. Happiness is revealed in upturned corners of the mouth; laughing also raises the cheeks which in turn may push the lower eyelids up.

Blends can be formed by combining signs of two emotions. For example, arched eyebrows and a smile indicate surprised happiness. Subtle feelings also may be communicated by rapid sequences of expressions — an angry expression interrupted by a flash of disgust.

### The Face Program

The face synthesizer presented here consists of 2K of assembly-language code designed to run on 32K PET/CBM microcomputers with 40-column screens. Instructions are given on relocating the code for 16K or 8K machines. The synthesizer does not work with Commodore operating system 1.0, and it produces a long and narrow face on 80-column screens.

The facial image was created by tracing a photo of a woman's face in a magazine onto graph paper, and then matching features in each cell as closely as possible with Commodore graphics. The happy, grinning face that appears by default is the original. Feature variations were created artistically, with guidance from photographs of facial expressions. Gaze variations were constructed so that the face can be made to look forward, left, right, or down. Left and right eyelids can be independently controlled for winks, blinks, and closed eyes.

The program includes limited feature variations. Each feature shape is represented in a single form, though real faces can produce gradations. Some eye variations are too subtle for Commodore graphics, so a single approximate shape has to serve multiple duty. Nose crinkling for disgust and raised cheeks for laughter are not included. No asymmetric moves for brows or mouth are included.

Paul Ekman and Wallace Friesen's book, *Unmasking the Face* (Prentice-Hall, 1975), guided composition of the feature variations. This paperback is an essential manual for anyone using the face program in application programming.

### The Emotional Keyboard

Running the programs listed here produces the PET face and an initial display of some of its expressive variations. After the opening show, you can produce new expressions from the keyboard. Each feature variation is linked to a single key. The brow is controlled by keys in the top row of the keyboard. Eyes and eyelids are controlled by keys in the next two rows, with left eye variations on the left and right eye variations on the right. The mouth is controlled by keys in the bottom row. The six basic emotion configurations are available from single keys — parentheses, brackets, and inequalities. The number keys control compositions that are needed in programming. Chart 1 indexes command characters and their effects.

The demonstration program allows you to construct a string of commands that can produce an animated sequence of expressions. To begin a string, press the "+" key. When you have completed the sequence and are ready to view it, press the "=" key. Use the delay command (shifted SPACE) between different expressions so that the sequence runs slow enough for you to see. You may press the "*" key to see the last sequence again. If you press the "/" key, the commands producing the last sequence are printed on the screen and the program ends. Use this option to work out desired effects before entering commands into your BASIC program.

### Faces in BASIC

Listing 1 presents the demonstration program that illustrates all of the points essential to using the Face routine with BASIC. Commands are sent to the routine *via* a string variable named FACE$ (or FA$) (this must be defined in the program before calling the Face routine or you will get an error). The FACE$ string may be constructed by direct quotes; by GET, INPUT, or READ statements; or by string manipulations.

**Listing 1**

```
100 REM  COMPOSE THE FACE, AND SUBSTITUTE A SMILE FOR THE DEFAULT GRIN.
110 FACE$="01234?"
120 SYS 30729 :REM 14345 FOR 16K; 6153 FOR 8K.
130 REM  GO THROUGH SEQUENCE OF BASIC EMOTIONS WITH PAUSES BETWEEN CHANGES.
140 REM  "[ ]" IS SHIFTED SPACE.
150 FA$="[ ][ ][ ][ ][ ])[ ][ ][ ][ ][ ][ ][[ ][ ][ ][ ][ ][ ]]"
160 FA$=FA$+"[ ][ ][ ][ ][ ][ ]<[ ][ ][ ][ ][ ][ ]>[ ][ ][ ][ ][ ][ ]("
170 SYS 30729 :REM 14345 FOR 16K; 6153 FOR 8K.
180 REM  PRODUCE SLY LOOKS TO LEFT AND RIGHT, THEN A WINK.
190 FA$="[ ][ ][ ][ ][ ][ ]Q[ ][ ][ ]P[ ][ ][ ]T[ ][ ][ ]A[ ]T"
200 SYS 30729 :REM 14345 FOR 16K; 6153 FOR 8K.
210 REM  CLEAR THE COMMAND STRING.
220 FA$=""
230 REM  GET KEYBOARD COMMAND.
240 GET A$:IF A$="" THEN 240
250 REM  START A SEQUENCE? SET THE SERIAL FLAG AND CLEAR FA$.
260 IF A$="+" THEN A=-1:GOTO 220
270 REM  DISPLAY THE SEQUENCE? CLEAR FLAG AND STORE STRING FIRST.
280 IF A$="=" THEN A=0:B$=FA$:GOTO 200
290 REM  CONSTRUCTING A SEQUENCE? ADD TO FA$ AND GET MORE.
300 IF A THEN FA$=FA$+A$:GOTO 240
310 REM  DISPLAY LAST SEQUENCE? USE STORED STRING.
320 IF A$="*" THEN FA$=B$:GOTO 200
330 REM  PRINT COMMANDS IN LAST SEQUENCE? CLEAR SCREEN, PRINT STRING, AND END.
340 IF A$="/" THEN PRINT "<s>"B$:END
350 REM  OTHERWISE DISPLAY COMMAND FROM KEYBOARD.
360 FA$=A$
370 GOTO 200
```

**Listing 2**

```
100 : REM: LOADER PROGRAM
110 :
120 : REM: ROUTINE TO LOAD THE ASSEMBLED CODE, THEN THE MAIN PROGRAM.
130 : REM: DELETE NEXT LINE TO LOAD FROM TAPE, INCLUDE IT TO LOAD FROM DISK.
140 DV$=",8"
150 : REM: SET GRAPHICS MODE.
160 POKE 59468,12
170 : REM: STORE 4 CARRIAGE RETURNS IN INPUT BUFFER,
180 POKE 158,4:POKE 623,13:POKE 624,13:POKE 625,13:POKE 626,13
190 : REM: SET UP SCREEN TO INVOKE THE SUBROUTINE FILE -- 'CODE',
200 : REM: "<s>" IS CLEAR SCREEN. "<Q>" IS CURSOR DOWN.
210 IF DV$<>",8" THEN DV$="<Q>"
220 PRINT "<s><Q><Q>LOAD" CHR$(34) "CODE" CHR$(34)DV$:PRINT "<Q><Q><Q><Q>SYS 30720"
230 : REM: THEN LOAD PROGRAM 'MAIN'
240 PRINT "<Q><Q>LOAD" CHR$(34) "MAIN" CHR$(34)DV$
250 : REM: "<S>" IS CURSOR HOME.
260 PRINT "<Q><Q><Q><Q>RUN<S>";:END
```

The first string sent to the routine should begin with "01234". These five commands display the face on the screen. Thereafter the FACE$ string consists only of commands for desired feature changes.

Commands in the FACE$ string are implemented by SYS 30729 on a 32K PET, SYS 14345 on a 16K PET, or SYS 6153 on an 8K PET.

Listing 2 is a BASIC utility program that automatically loads the Face code from a file named CODE, protects the code from BASIC, and then loads another BASIC program named MAIN. Program MAIN would be listing 1 when you set up the demonstration procedure; otherwise it is the application program you have written. On tape, the Loader program should be first, the CODE file second, and MAIN third. On disk, the order is immaterial, but the names CODE and MAIN are required. (The DOS wedge can be loaded after the face routines are loaded.)

### Assembly-Language Routine

Listing 3 is the assembly-language routine. Listing 4 contains the data used to compose the face and its variations on the screen.

You enter the code in listings 3 and 4 with the PET/CBM monitor (SYS 1024). To begin enter:

```
.M 7800,787F
```

Then overwrite the contents of the cells with the hexadecimal values at the left of listing 3, pressing RETURN after each line. Continue with:

```
.M 7880,7900
```

and so on. When you have finished

# Commodore Programming Moves into the *Fast Lane!*

**Petspeed — The Optimizing Basic Compiler that runs Basic 40 times faster. Petspeed brings speed, compatibility and ease to your Basic programming. Now you can dramatically reduce long processing times, tedious disk handling, and long print runs.**

**Compatible** Petspeed compiles any Basic program and is available for any combination of 4000 and 8000 series Commodore machines.

**Faster** The key that sets Petspeed apart from other compilers is **optimization.** Where most compilers merely translate from one language to another, Petspeed analyzes your source program to eliminate unnecessary complexities, thus speeding processing time. Your program is reduced to much smaller components and reassembled into a far more efficient form. Compare these optimizing features:

- 4-Pass compiler.
- Automatically uses faster integer arithmetic when possible.
- Automatically handles frequently occurring variables and arrays.
- Subroutines no longer need be located at the beginning of your program.
- Petspeed automatically calls all subroutines at maximum speed.
- Petspeed runs twice as fast as other compilers.
- Larger programs require far less memory when compiled.

**Easy to Use** Petspeed is as easy to use as these screen displays illustrate.

```
*** commodore basic 4.0 ***

 31743 bytes free

ready.

directory d1
 1 "petspeed         " ps 2c
 91   "your program"     prg
 1961 blocks free.

ready.
```

*Directory BEFORE compilation.*

```
PETSPEED

     PROGRAM NAME : YOUR PROGRAM

ISSUE 2.3     (C) O.C.S.S. 1982
```

*Simply type in your program name.*

```
*** commodore basic 4.0 ***

 31743 bytes free

ready.
cAd1
 1 "petspeed         " ps 2c
 91   "your program"     prg
 65   "your program.gt"  prg
 18   "your program.w"   prg
 1779 blocks free.
ready.
```

*Directory AFTER compilation*

It isn't necessary to add compiler directives. Simply type in the program name. In less than 2 minutes, you'll see your program run up to 40 times faster.

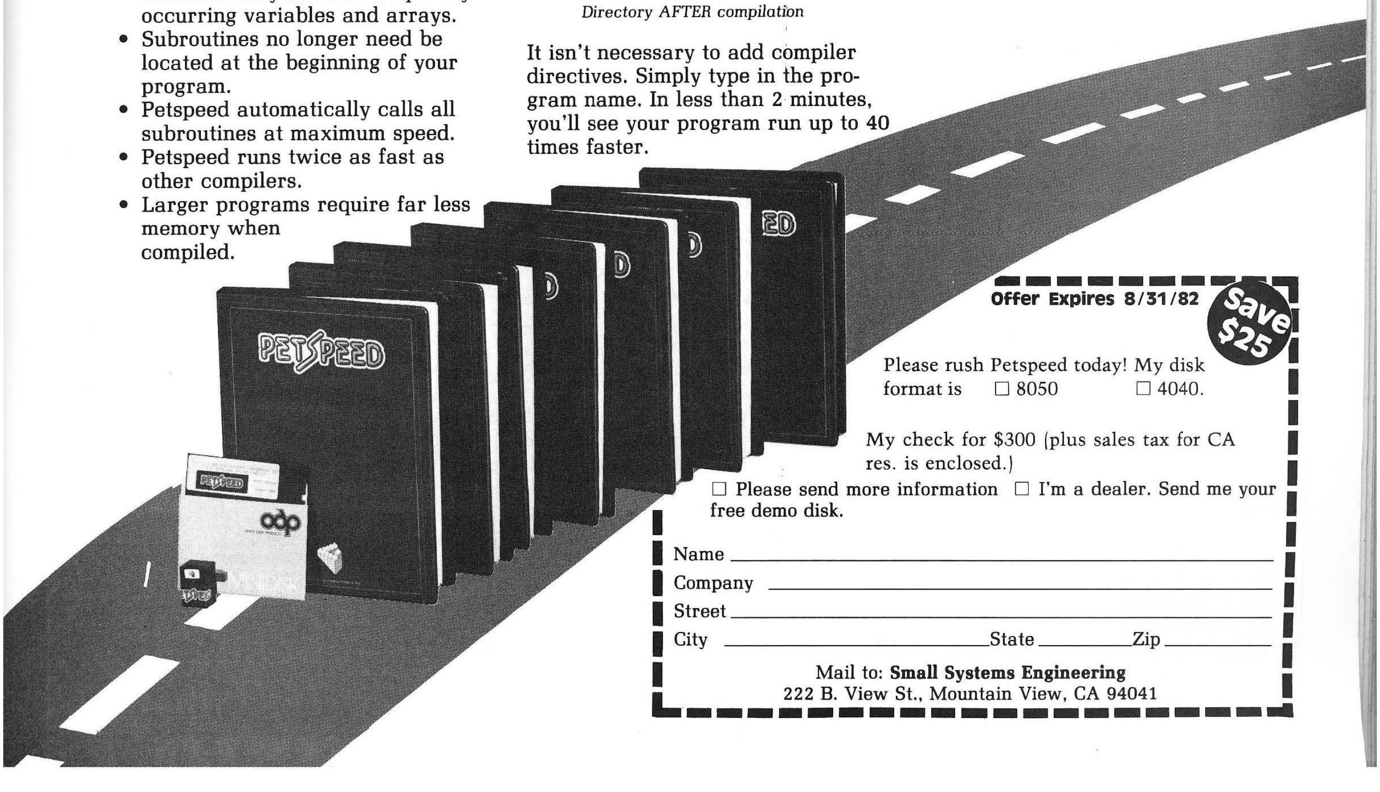
**Security** A security device is provided to run Petspeed, but no runtime key is necessary for compiled programs. You're free to build in your own protection. Petspeed code cannot be listed by others, so compiled programs cannot be tampered with. Your programs belong entirely to you.

Move your Commodore into the fast lane today with Petspeed. And save $25 too! Take this coupon to your dealer, or send to

SSE

**SMALL SYSTEMS ENGINEERING**
222 B View St.
Mountain View, California 94041
**(415) 964-8201**

**Dealers: Ask for our free demo disk.**
**Price $325**



**Offer Expires 8/31/82**

Save $25

Please rush Petspeed today! My disk format is ☐ 8050 ☐ 4040.

My check for $300 (plus sales tax for CA res. is enclosed.)

☐ Please send more information ☐ I'm a dealer. Send me your free demo disk.

Name ____________________
Company ____________________
Street ____________________
City __________ State _____ Zip _____

Mail to: **Small Systems Engineering**
222 B. View St., Mountain View, CA 94041

with the data in listing 3, go on to listing 4. Print out lines of memory the same as they appear in listing 4 to simplify entry.

When all code has been entered, save it on tape with:

```
.S "CODE",01,7800,7FF2
```

or on a preinitialized disk in unit 0 with:

```
.S "0:CODE",08,7800,7FF2
```

If you are saving on tape, remember to save the Loader program (listing 2) on the tape before you begin to enter the code in listings 3 and 4.

To relocate the code for a 16K machine, change all $7000 addresses to $3000 addresses. For example, you would begin entering code with:

```
.M 3800,387F
```

Some addresses within the program have to be changed. Relevant lines are flagged by <SIZE> in the comments column. Change the 7 in the address high byte to 3 wherever <SIZE> appears. For example, A9 77 in line 440 would become A9 37. In addition, the last byte of each entry in the INDEX must be changed (the INDEX begins at line 2080). For example, 79 in the SCALP entry would be changed to 39, 7A in the BROW entry becomes 3A, etc.

Relocation for 8K is similar except that sevens are changed to ones.

The address in line 1890 of listing 3 is for operating system 3.0. Change 69 C3 to 00 BF for operating system 4.0.

The data in listing 4 remain the same for all machines.

## Program Notes

Lines 70-100 in listing 3 are instructions for the assembler program.

Lines 140-210 indicate parts of the PET/CBM operating system that are used in the Face program.

Lines 240-350 show the locations in screen memory where facial features begin.

Lines 420-460 are a short routine for putting the Face code outside the bounds of BASIC. This routine is called by the Loader program in listing 2 immediately after loading the Face code.

The Face program begins in line 540. First the FACE$ command string has to be found (lines 540-590 plus the subroutine in lines 1630-2030).

Commands in the FACE$ string are transferred to the stack in reverse order (lines 620-670), after a zero is pushed on the stack to signal the end of the commands (lines 600-610).

Commands are retrieved from the stack by the routine in lines 700-740. If a zero is encountered, then all commands have been processed and control returns to BASIC. If the command value is negative (greater than 127), then the routine drops into a dummy loop that causes a delay. A shifted space has ASCII value 160 so it causes a delay. Otherwise control shifts to line 850, and a search is initiated to find the command in the Index (lines 850-940). If the command is not found, it is ignored, and the program branches to get the next command from the stack.

When a command to change a feature is found, lines 950-990 transfer the screen pointer for the relevant feature to cells in the floating point accumulator (FACC serves as free zero-page memory for this routine). Then lines 1000-1080 set up a short subroutine in FACC to fetch bytes from the stored data. The screen pointer and the pointer to the stored data both are obtained from the Index entry for the command being implemented.

Lines 1110-1150 get a byte from the stored data and set indexes for current use. The data byte is tested in lines 1160-1210 to see if it is an ordinary datum or a special subcommand.

If the byte is zero, it means that all data have been transfered, and control branches to get the next Face command. Values one, two, and three, are special subcommands used to reduce the amount of space needed for data.

Value one is a skip command. The byte following is fetched to determine how many screen cells to skip, and then the screen pointer is adjusted to accomplish this (lines 1330-1400).

Value two is a duplicate command. The byte following the two is fetched to define the character to be duplicated. The byte after that is fetched to determine how many times the character is to be displayed. Then the character is put on the screen the required number of times (lines 1420-1580).

Value three causes a feature manipulation to be appended after the current one. The byte following the three is fetched and pushed on the stack. Thus it will be the next Face command to be implemented (lines 1250-1270).

If the data byte is not zero, one, two, or three, then it is a character byte, and it is transfered directly to the screen (lines 1300-1310). Processing of data bytes continues in a loop until a zero value is found.

The subroutine in lines 1630-2030 searches the BASIC variable list for the FACE$ string. When it is found, the pointer to the string and the string's length are saved for use in the main program. If FA$ is not found, the procedure aborts *via* the BASIC error routine.

The Index shows the ASCII value for each command character, the place on the screen where the relevant feature begins, and the point in memory (listing 4) where data for that command are stored.

**Chart 1**

| **Keyboard Character** | **Effect on Face** |
|---|---|
| *Complete face* | |
| ( | Happiness |
| ) | Sadness |
| [ | Fear |
| ] | Anger |
| < | Disgust |
| > | Surprise |
| *Eyebrows* | |
| & | Normal |
| \ | Arched high |
| % | Flat, lowered |
| # | Flat, raised |
| $ | Furrowed down |
| *Eyes* | |
| T | Normal, gaze forward |
| I | Normal, gaze down |
| W | Normal, gaze left |
| O | Normal, gaze right |
| E | Lids drooping, gaze forward |
| Y | Lids high, gaze forward |
| R | Lids high, gaze downward |
| U | Lids down, gaze forward |
| Q | Lids down, gaze left |
| P | Lids down, gaze right |
| S | Left eye closed |
| K | Right eye closed |
| A | Left eye tense closed |
| L | Right eye tense closed |
| *Mouth* | |
| X | Normal |
| ? | Smiling |
| Z | Grinning |
| B | Saddened |
| V | Frightened |
| C | Disgusted |
| N | Dropped open |
| M | Opened showing teeth |
| , | Lips pressed tight |
| *Other* | |
| Shift Space | Brief delay |
| 01234 | Construct grinning face |
| 5 | Chin up |
| 6 | Chin down |
| 7 | Nose normal |
| 9 | Nose up |

**Listing 3**

```
                0010             ;              ****************************
                0020             ;              *                          *
                0030             ;              *           FACE           *
                0040             ;              *                          *
                0050             ;              ****************************
                0060             ;
                0070 START       .BA 30720     ;PLACE WHERE CODE BEGINS WHEN
                                               USED.
                0080             ;
                0090             .MC $7800     ;PLACE WHERE CODE ASSEMBLES.
                0100             .OS  ;        ;CREATE CODE.
                0110             ;
                0120             ;
                0130             ;              BASIC SUBROUTINE
                0140 OUT         .DE $C369     ;ERROR ROUTINE ($BFOO IN O.S.
                                               4.0)
                0150             ;
                0160             ;              ZERO-PAGE USAGE
                0170 PTR         .DE 42        ;BEGINNING OF VARIABLES
                0180 LIM         .DE 44        ;END OF VARIABLES
                0190 RAM         .DE 52        ;LOCATION FOR HIGHEST RAM
                                               ADDRESS
                0200 NAME        .DE 66        ;VARIABLE NAME
                0210 FACC        .DE 94        ;FLOATING POINT ACCUMULATOR
                0220             ;
                0230             ;              SCREEN VARIABLES
                0240 ROW         .DE 40
                0250 SCALP@      .DE 32768
                0260 BROW@       .DE 32968
                0270 MIDFACE@    .DE 33168
                0280 CHEEKS@     .DE 33368
                0290 NECK@       .DE 33568
                0300 BROWS@      .DE 33059
                0310 EYES@       .DE 33140
                0320 R\EYE@      .DE 33150
                0330 NOSE@       .DE 33344
                0340 MOUTH@      .DE 33463
                0350 CHIN@       .DE 33620
                0360             ;
                0370             ;              ****************************
                0380             ;              SYS 30720 FOR 32K,
                0390             ;                  14336 FOR 16K,
                0400             ;                   6144 FOR  8K.
                0410             ;              PROTECT SUBS FROM BASIC.
7800- A9 FF     0420 INVOKE      LDA #L,INVOKE-1
7802- 85 34     0430             STA *RAM
7804- A9 77     0440             LDA #H,INVOKE-1       ;<SIZE>
7806- 85 35     0450             STA *RAM+1
7808- 60        0460             RTS
                0470             ;
                0480             ;              ****************************
                0490             ;              SYS 30729 FOR 32K,
                0500             ;                  14345 FOR 16K,
                0510             ;                   6153 FOR  8K.
                0520             ;
                0530             ;              STACK THE SET OF COMMANDS.
7809- A9 46     0540 FACE        LDA #'F'      ;FIND THE FACE$ VARIABLE.
780B- 85 42     0550             STA *NAME
780D- A9 41     0560             LDA #'A'
780F- 09 80     0570             ORA #128
7811- 85 43     0580             STA *NAME+1
7813- 20 AE 78  0590             JSR FIND      ;      ;<SIZE>
7816- A9 00     0600             LDA #0        ;END COMMANDS WITH A 0.
7818- 48        0610             PHA
7819- A4 5E     0620             LDY *FACC     ;GET LENGTH OF COMMAND STRING.
781B- B1 42     0630 STORE       LDA (NAME),Y
781D- 48        0640             PHA  ;        ;PUT COMMANDS ON STACK IN
                                               REVERSE ORDER.
781E- 88        0650             DEY
781F- C0 FF     0660             CPY #255
7821- D0 F8     0670             BNE STORE
                0680             ;
                0690             ;              IMPLEMENT THE COMMANDS.
7823- A2 00     0700 PRINT       LDX #0        ;GET THE NEXT COMMAND.
7825- 68        0710             PLA
7826- 30 03     0720             BMI TIME
7828- D0 0D     0730             BNE SEARCH
782A- 60        0740             RTS  ;        ;DONE IF ZERO.
782B- A2 60     0750 TIME        LDX #96       ;SHIFTED SPACE CAUSES DELAY.
782D- A0 00     0760             LDY #0
782F- 88        0770 DELAY       DEY
7830- D0 FD     0780             BNE DELAY
7832- CA        0790             DEX
7833- D0 FA     0800             BNE DELAY
7835- F0 EC     0810             BEQ PRINT
                0820             ;
                0830             ;              PREPARE FOR DATA TRANSFER.
                0840             ;             ;FIND THE POINTERS FOR THE
                                                COMMAND.
7837- DD F7 78  0850 SEARCH      CMP INDEX,X   ;       ;<SIZE>
783A- F0 0B     0860             BEQ DUMP
783C- E8        0870             INX
783D- E8        0880             INX
783E- E8        0890             INX
783F- E8        0900             INX
7840- E8        0910             INX
7841- E0 D7     0920             CPX #INDEX\SIZE
7843- D0 F2     0930             BNE SEARCH
7845- F0 DC     0940             BEQ PRINT     ;NOT FOUND--GO TO NEXT
                                               COMMAND.
```

**Listing 3** *(Continued)*

```
7847- E8        0950 DUMP        INX  ;        ;SET UP SCREEN POINTER IN FACC.
7848- BD F7 78  0960             LDA INDEX,X   ;       ;<SIZE>
784B- 85 5E     0970             STA *FACC
784D- BD F8 78  0980             LDA INDEX+1,X         ;<SIZE>
7850- 85 5F     0990             STA *FACC+1
7852- A9 BD     1000             LDA #$BD      ;STORE LDA,X IN FACC.
7854- 85 60     1010             STA *FACC+2
                1020             ;             ;SET UP DATA POINTER IN FACC.
7856- BD F9 78  1030             LDA INDEX+2,X         ;<SIZE>
7859- 85 61     1040             STA *FACC+3
785B- BD FA 78  1050             LDA INDEX+3,X         ;<SIZE>
785E- 85 62     1060             STA *FACC+4
7860- A9 60     1070             LDA #$60      ;STORE RTS IN FACC.
7862- 85 63     1080             STA *FACC+5
                1090             ;
                1100             ;              GO THROUGH DATA.
7864- A0 FF     1110             LDY #255      ;GET NEXT BYTE.
7866- A2 FF     1120             LDX #255
7868- C8        1130 GETONE1     INY
7869- E8        1140 GETONE2     INX
786A- 20 60 00  1150             JSR FACC+2
786D- F0 B4     1160             BEQ PRINT     ;CODE=0 IS END: DO NEXT COMMAND.
786F- C9 02     1170             CMP #2
7871- 90 11     1180             BCC SKIPON    ;CODE=1 MEANS SKIP CELLS.
7873- F0 1E     1190             BEQ REPEAT    ;CODE=2 MEANS REPEAT BYTE.
7875- C9 03     1200             CMP #3
7877- D0 07     1210             BNE TRANSFER
7879- E8        1220             INX  ;        ;CODE=3 MEANS APPEND.
                1230             ;
                1240             ;             ;FOR APPEND:
787A- 20 60 00  1250             JSR FACC+2    ;GET COMMAND
787D- 48        1260             PHA  ;        ;AND PUSH ON STACK.
787E- D0 E9     1270             BNE GETONE2
                1280             ;             ;FOR NORMAL BYTE:
                1290             ;             ;PUT IT ON SCREEN.
7880- 91 5E     1300 TRANSFER    STA (FACC),Y
7882- D0 E4     1310             BNE GETONE1
                1320             ;             ;FOR SKIP:
7884- E8        1330 SKIPON      INX  ;        ;FIND NUMBER OF CELLS TO SKIP
7885- 20 60 00  1340             JSR FACC+2
7888- 18        1350             CLC  ;        ;AND ADD TO THE SCREEN POINTER.
7889- 65 5E     1360             ADC *FACC
788B- 85 5E     1370             STA *FACC
788D- 90 DA     1380             BCC GETONE2
788F- E6 5F     1390             INC *FACC+1
7891- B0 D6     1400             BCS GETONE2
                1410             ;             ;FOR REPEAT:
7893- E8        1420 REPEAT      INX  ;        ;GET THE CHARACTER
7894- 20 60 00  1430             JSR FACC+2
7897- 48        1440             PHA
7898- E8        1450             INX  ;        ;AND THE NUMBER OF REPEATS.
7899- 86 64     1460             STX *FACC+6   ;(REMEMBER THE CURRENT
                                               DATA INDEX.)
789B- 20 60 00  1470             JSR FACC+2
789E- AA        1480             TAX
789F- 68        1490             PLA
                1500             ;             ;PUT THE CHARACTER ON THE
                                               SCREEN
78A0- 91 5E     1510 ZIP         STA (FACC),Y
78A2- C8        1520             INY
78A3- D0 02     1530             BNE CONT
78A5- E6 5F     1540             INC *FACC+1
78A7- CA        1550 CONT        DEX
78A8- D0 F6     1560             BNE ZIP       ;REPEATEDLY.
78AA- A6 64     1570             LDX *FACC+6   ;RECOVER INDEX
78AC- D0 BB     1580             BNE GETONE2   ;TO CONTINUE WITH DATA.
                1590             ;
                1600             ;
                1610             ;              ****************************
                1620             ;              FIND VARIABLE IN BASIC LIST
78AE- A5 2A     1630 FIND        LDA *PTR      ;SAVE TABLE POINTER.
78B0- 48        1640             PHA
78B1- A5 2B     1650             LDA *PTR+1
78B3- 48        1660             PHA
78B4- A0 00     1670 CHECK       LDY #0        ;COMPARE NAME IN VARIABLE TABLE
78B6- B1 2A     1680             LDA (PTR),Y
78B8- C5 42     1690             CMP *NAME     ;WITH CRITERION NAME.
78BA- D0 08     1700             BNE AGAIN2
78BC- C8        1710             INY
78BD- B1 2A     1720             LDA (PTR),Y
78BF- C5 43     1730             CMP *NAME+1
78C1- F0 1D     1740             BEQ POINT     ;IF NO MATCH, THEN
78C3- 88        1750             DEY
78C4- A9 07     1760 AGAIN2      LDA #7        ;GET POSITION OF NEXT VARIABLE
78C6- 18        1770             CLC
78C7- 65 2A     1780             ADC *PTR
78C9- 85 2A     1790             STA *PTR
78CB- A9 00     1800             LDA #0
78CD- 65 2B     1810             ADC *PTR+1
78CF- 85 2B     1820             STA *PTR+1
78D1- C5 2D     1830             CMP *LIM+1    ;IF NOT END OF VARIABLES
78D3- D0 DF     1840             BNE CHECK     ;THEN CONTINUE SEARCHING.
78D5- A5 2C     1850             LDA *LIM
78D7- C5 2A     1860             CMP *PTR
78D9- D0 D9     1870             BNE CHECK
78DB- A2 81     1880             LDX #$81      ;IF NAME NOT FOUND, ABORT.
```

### Listing 3 *(Continued)*

```
78DD- 4C 69 C3  1890              JMP OUT
78E0- A0 02     1900 POINT        LDY #2         ;FOR THE MATCHING VARIABLE,
78E2- B1 2A     1910              LDA (PTR),Y    ;GET LENGTH OF STRING
78E4- 85 5E     1920              STA *FACC      ;AND STORE FOR LATER.
78E6- C8        1930              INY  ;         ;THEN GET POINTER TO STRING
78E7- B1 2A     1940              LDA (PTR),Y
78E9- 85 42     1950              STA *NAME      ;AND STORE IN THE NAME
                                                 CELLS.
78EB- C8        1960              INY
78EC- B1 2A     1970              LDA (PTR),Y
78EE- 85 43     1980              STA *NAME+1
78F0- 68        1990              PLA  ;         ;RESTORE POINTER FOR
                                                 VARIABLE TABLE.
78F1- 85 2B     2000              STA *PTR+1
78F3- 68        2010              PLA
78F4- 85 2A     2020              STA *PTR
78F6- 60        2030              RTS
                2040              ;
                2050              ;               ***************************
                2060 INDEX\SIZE .DE 215
                2070              ;
78F7- 30        2080 INDEX        .BY '0'
78F8- 00 80     2090              .SE SCALP@
78FA- CE 79     2100              .SI SCALP
                2110              ;
78FC- 31        2120              .BY '1'
78FD- C8 80     2130              .SE BROW@
78FF- 24 7A     2140              .SI BROW
                2150              ;
7901- 32        2160              .BY '2'
7902- 90 81     2170              .SE MIDFACE@
7904- 89 7A     2180              .SI MIDFACE
                2190              ;
7906- 33        2200              .BY '3'
7907- 58 82     2210              .SE CHEEKS@
7909- 07 7B     2220              .SI CHEEKS
                2230              ;
790B- 34        2240              .BY '4'
790C- 20 83     2250              .SE NECK@
790E- 6D 7B     2260              .SI NECK
                2270              ;
7910- 26        2280              .BY '&'
7911- 23 81     2290              .SE BROWS@
7913- FF 7C     2300              .SI BROW\UP
                2310              ;
7915- 5C        2320              .BY '\'
7916- 23 81     2330              .SE BROWS@
7918- 8F 7D     2340              .SI BROW\HIGH
                2350              ;
791A- 25        2360              .BY '%'
791B- 23 81     2370              .SE BROWS@
791D- 20 7D     2380              .SI BROW\SAD
                2390              ;
791F- 23        2400              .BY '#'
7920- 23 81     2410              .SE BROWS@
7922- 47 7D     2420              .SI BROW\FEAR
                2430              ;
7924- 24        2440              .BY '$'
7925- 23 81     2450              .SE BROWS@
7927- 6A 7D     2460              .SI BROW\DOWN
                2470              ;
7929- 54        2480              .BY 'T'
792A- 74 81     2490              .SE EYES@
792C- C5 7B     2500              .SI EYES\AHEAD
                2510              ;
792E- 52        2520              .BY 'R'
792F- 74 81     2530              .SE EYES@
7931- E0 7B     2540              .SI EYES\DOWN
                2550              ;
7933- 45        2560              .BY 'E'
7934- 74 81     2570              .SE EYES@
7936- FB 7B     2580              .SI EYES\SAD
                2590              ;
7938- 59        2600              .BY 'Y'
7939- 74 81     2610              .SE EYES@
793B- 16 7C     2620              .SI EYES\BULGE
                2630              ;
793D- 57        2640              .BY 'W'
793E- 74 81     2650              .SE EYES@
7940- 31 7C     2660              .SI EYES\LEFT
                2670              ;
7942- 4F        2680              .BY 'O'
7943- 74 81     2690              .SE EYES@
7945- 4C 7C     2700              .SI EYES\RIGHT
                2710              ;
7947- 53        2720              .BY 'S'
7948- 74 81     2730              .SE EYES@
794A- 67 7C     2740              .SI L\EYE\CLOS
                2750              ;
794C- 4B        2760              .BY 'K'
794D- 7E 81     2770              .SE R\EYE@
794F- 72 7C     2780              .SI R\EYE\CLOS
                2790              ;
7951- 41        2800              .BY 'A'
7952- 74 81     2810              .SE EYES@
7954- 7D 7C     2820              .SI L\EYE\SQEZ
                2830              ;
```

### Listing 3 *(Continued)*

```
7956- 4C        2840              .BY 'L'
7957- 7E 81     2850              .SE R\EYE@
7959- 88 7C     2860              .SI R\EYE\SQEZ
                2870              ;
795B- 55        2880              .BY 'U'
795C- 74 81     2890              .SE EYES@
795E- 93 7C     2900              .SI EYE\HOOD
                2910              ;
7960- 51        2920              .BY 'Q'
7961- 74 81     2930              .SE EYES@
7963- AE 7C     2940              .SI EYE\HOOD\L
                2950              ;
7965- 50        2960              .BY 'P'
7966- 74 81     2970              .SE EYES@
7968- C9 7C     2980              .SI EYE\HOOD\R
                2990              ;
796A- 49        3000              .BY 'I'
796B- 74 81     3010              .SE EYES@
796D- E4 7C     3020              .SI EYE\HOOD\D
                3030              ;
796F- 37        3040              .BY '7'
7970- 40 82     3050              .SE NOSE@
7972- A2 7F     3060              .SI NOSE\NRML
                3070              ;
7974- 39        3080              .BY '9'
7975- 40 82     3090              .SE NOSE@
7977- B6 7F     3100              .SI NOSE\UP
                3110              ;
7979- 5A        3120              .BY 'Z'
797A- B7 82     3130              .SE MOUTH@
797C- AB 7D     3140              .SI GRIN
                3150              ;
797E- 3F        3160              .BY '?'
797F- B7 82     3170              .SE MOUTH@
7981- DF 7D     3180              .SI SMILE
                3190              ;
7983- 58        3200              .BY 'X'
7984- B7 82     3210              .SE MOUTH@
7986- 0D 7E     3220              .SI LIPS\FLAT
                3230              ;
7988- 4E        3240              .BY 'N'
7989- B7 82     3250              .SE MOUTH@
798B- 3B 7E     3260              .SI LIPS\OPEN
                3270              ;
798D- 42        3280              .BY 'B'
798E- B7 82     3290              .SE MOUTH@
7990- 6D 7E     3300              .SI LIPS\DOWN
                3310              ;
7992- 56        3320              .BY 'V'
7993- B7 82     3330              .SE MOUTH@
7995- 95 7E     3340              .SI LIPS\FEAR
                3350              ;
7997- 4D        3360              .BY 'M'
7998- B7 82     3370              .SE MOUTH@
799A- C9 7E     3380              .SI LIPS\SNARL
                3390              ;
799C- 2C        3400              .BY ','
799D- B7 82     3410              .SE MOUTH@
799F- 33 7F     3420              .SI LIPS\PRESS
                3430              ;
79A1- 43        3440              .BY 'C'
79A2- B7 82     3450              .SE MOUTH@
79A4- 01 7F     3460              .SI LIPS\DSGST
                3470              ;
79A6- 35        3480              .BY '5'
79A7- 54 83     3490              .SE CHIN@
79A9- 61 7F     3500              .SI CHIN\UP
                3510              ;
79AB- 36        3520              .BY '6'
79AC- 54 83     3530              .SE CHIN@
79AE- 82 7F     3540              .SI CHIN\DOWN
                3550              ;
79B0- 28        3560              .BY '('
79B1- 00 00     3570              .SE 0
79B3- CA 7F     3580              .SI HAPPY
                3590              ;
79B5- 29        3600              .BY ')'
79B6- 00 00     3610              .SE 0
79B8- D1 7F     3620              .SI SAD
                3630              ;
79BA- 5B        3640              .BY '['
79BB- 00 00     3650              .SE 0
79BD- D6 7F     3660              .SI SCARED
                3670              ;
79BF- 5D        3680              .BY ']'
79C0- 00 00     3690              .SE 0
79C2- DD 7F     3700              .SI MAD
                3710              ;
79C4- 3C        3720              .BY '<'
79C5- 00 00     3730              .SE 0
79C7- E4 7F     3740              .SI DISGUSTED
                3750              ;
79C9- 3E        3760              .BY '>'
79CA- 00 00     3770              .SE 0
79CC- EB 7F     3780              .SI SURPRISED
```

**Listing 4**

```
79CE 02 A0 06 74 02 20 08 64
79D6 6F 62 F8 E3 F8 62 6F 64
79DE 02 20 08 6A 02 A0 0D 02
79E6 20 07 6C FE 02 A0 09 62
79EE 7B 02 20 07 02 A0 0C EA
79F6 02 20 07 02 A0 0D 02 20
79FE 07 F4 02 A0 0B 61 02 20
7A06 06 E9 02 A0 0D DF 02 20
7A0E 06 E1 02 A0 0B 75 02 20
7A16 05 6C 02 A0 0F 7B 02 20
7A1E 05 6A 02 A0 06 00 02 A0
7A26 05 02 20 06 02 A0 11 02
7A2E 20 06 02 A0 0A F6 02 20
7A36 05 E9 02 A0 11 DF 02 20
7A3E 05 E1 02 A0 09 75 02 20
7A46 04 E9 A0 01 11 A0 DF 02
7A4E 20 04 76 02 A0 09 74 20
7A56 20 20 6C A0 A0 01 11 A0
7A5E A0 7B 20 20 20 6A 02 A0
7A66 09 02 20 04 FE 02 A0 04
7A6E D2 F9 C6 A0 A0 E5 A0 E7
7A76 A0 A0 C6 F9 D2 02 A0 04
7A7E FC 02 20 04 F5 02 A0 04
7A86 03 26 00 A0 A0 A0 E7 20
7A8E 6C 62 FE DD 02 A0 04 CD
7A96 79 FE FD A0 C2 A0 C2 A0
7A9E ED FC 79 CE 02 A0 04 DD
7AA6 FC 62 7B 20 76 02 A0 07
7AAE EA 20 E1 A0 9C D9 02 A0
7AB6 09 C8 A0 D4 02 A0 09 D4
7ABE CE A0 61 20 6A 02 A0 07
7AC6 EA 20 7C FB A0 A0 C7 02
7ACE A0 13 D9 A0 A0 EC 7E 20
7AD6 6A 02 A0 07 E7 20 20 20
7ADE E2 FB C8 02 A0 13 C7 EC
7AE6 E2 20 20 20 76 02 A0 08
7AEE 75 20 20 20 E1 E7 02 A0
7AF6 08 D5 A0 C9 02 A0 08 E5
7AFE 61 02 20 04 F4 02 A0 04
7B06 00 02 A0 04 61 20 20 20
7B0E 6A 02 A0 08 F2 C6 A0 C6
7B16 F2 02 A0 08 74 20 20 20
7B1E 6A 02 A0 09 75 02 20 04
7B26 F4 02 A0 08 C5 E3 C5 02
7B2E A0 08 EA 02 20 04 76 02
7B36 A0 09 02 20 05 E1 02 A0
7B3E 13 61 02 20 05 02 A0 08
7B46 EA 02 20 05 76 EE 02 A0
7B4E 11 F0 75 02 20 05 F4 02
7B56 A0 07 61 02 20 05 6A ED
7B5E EE 02 A0 0F F0 FD 74 02
7B66 20 05 E1 02 A0 04 00 A0
7B6E A0 A0 74 02 20 05 67 A0
7B76 9C 02 A0 0F CE A0 65 02
7B7E 20 05 6A 02 A0 07 02 20
7B86 06 67 02 A0 13 74 02 20
7B8E 06 02 A0 06 61 02 20 06
7B96 6A 02 A0 13 75 02 20 06
7B9E E1 02 A0 05 02 20 07 76
7BA6 02 A0 13 61 02 20 07 02
7BAE A0 04 EA 02 20 07 E1 02
7BB6 A0 13 F6 02 20 07 76 02
7BBE A0 02 03 5A 03 35 00 A0
7BC6 D2 F9 C6 A0 01 05 A0 C6
7BCE F9 D2 A0 01 19 A0 CD 79
7BD6 FE FD 01 05 ED FC 79 CE
7BDE A0 00 A0 D2 C6 C6 A0 01
7BE6 05 A0 C6 C6 D2 A0 01 19
7BEE A0 CD 20 FE FD 01 05 ED
7BF6 EC 20 CE A0 00 E4 D2 F9
7BFE C6 CB 01 05 CA C6 F9 D2
7C06 E4 01 19 A0 9C 79 FE FD
7C0E 01 05 ED FC 79 AF A0 00
7C16 A0 AF F9 9C A0 01 05 A0
```

**Listing 4** *(Continued)*

```
7C1E AF F9 9C A0 01 19 A0 9C
7C26 79 FE FD 01 05 ED FC 79
7C2E AF A0 00 A0 D2 F9 C6 A0
7C36 01 05 A0 F9 C6 D2 A0 01
7C3E 19 A0 CD 62 A0 FD 01 05
7C46 ED 62 A0 CE A0 00 A0 D2
7C4E C6 F9 A0 01 05 A0 C6 F9
7C56 D2 A0 01 19 A0 CD A0 62
7C5E FD 01 05 ED A0 62 CE A0
7C66 00 02 A0 05 01 23 A0 9C
7C6E C6 C6 FD 00 02 A0 05 01
7C76 23 ED C6 C6 AF A0 00 02
7C7E A0 05 01 23 A0 E3 C5 C4
7C86 FD 00 02 A0 05 01 23 ED
7C8E C4 C5 E3 A0 00 A0 E4 E4
7C96 E4 A0 01 05 A0 E4 E4 E4
7C9E A0 01 19 A0 CD 79 FE FD
7CA6 01 05 ED FC 79 CE A0 00
7CAE A0 E4 E4 E4 A0 01 05 A0
7CB6 E4 E4 E4 A0 01 19 A0 CD
7CBE 62 A0 FD 01 05 ED 62 A0
7CC6 CE A0 00 A0 E4 E4 E4 A0
7CCE 01 05 A0 E4 E4 E4 A0 01
7CD6 19 A0 CD A0 62 FD 01 05
7CDE ED A0 62 CE A0 00 A0 E4
7CE6 E4 E4 A0 01 05 A0 E4 E4
7CEE E4 A0 01 19 A0 CD 20 FB
7CF6 FD 01 05 ED EC 20 CE A0
7CFE 00 EF 77 63 E2 EF 02 A0
7D06 07 EF E2 63 77 EF 01 17
7D0E F7 A0 A0 F8 79 77 FB A0
7D16 A0 A0 EC 77 79 F8 A0 A0
7D1E F7 00 A0 E4 EF E4 A0 E4
7D26 EF A0 A0 A0 EF E4 A0 E4
7D2E EF E4 A0 01 17 62 F8 62
7D36 79 6F 79 F7 A0 A0 A0 F7
7D3E 79 6F 79 62 F8 62 03 45
7D46 00 F9 E2 78 E2 F9 E2 F9
7D4E A0 A0 A0 F9 E2 F9 E2 78
7D56 E2 F9 01 17 E3 A0 E3 F7
7D5E F8 F7 02 A0 05 F7 F8 F7
7D66 E3 A0 E3 00 EF 78 77 F9
7D6E EF E4 A0 DD DD DD A0 E4
7D76 EF F9 77 78 EF 01 17 F7
7D7E A0 E3 F8 79 6F 7C CB A0
7D86 CA 7E 6F 79 F8 E3 A0 F7
7D8E 00 A0 78 63 6F 64 63 FB
7D96 A0 A0 A0 EC 63 64 6F 63
7D9E 78 A0 01 17 7F E3 02 A0
7DA6 CD E3 FF A0 00 A0 A0 F4
7DAE EF E4 EF E4 A0 A0 01 1F
7DB6 7B 6C 62 62 79 62 62 7B
7DBE 6C 01 1F A0 7B 63 77 77
7DC6 77 63 6C A0 01 1F A0 A0
7DCE F7 F8 F7 F8 F7 A0 A0 01
7DD6 1E 02 A0 0B 03 35 03 37
7DDE 00 A0 A0 E4 EF E4 EF E4
7DE6 A0 A0 01 1F 7B 43 46 46
7DEE 52 46 46 43 6C 01 1F A0
7DF6 F7 79 6F 79 6F 79 F7 A0
7DFE 01 1F 02 A0 09 01 1E 02
7E06 A0 0B 03 35 03 37 00 02
7E0E A0 09 01 1F EC 78 77 63
7E16 77 63 77 78 FB 01 1F A0
7E1E 62 63 63 45 63 63 62 A0
7E26 01 1F A0 A0 A0 F7 E3 F7
7E2E A0 A0 A0 01 1E 02 A0 0B
7E36 03 35 03 37 00 02 A0 09
7E3E 01 1F A0 69 63 20 63 20
7E46 63 5F A0 01 1F A0 20 7C
7E4E E2 E2 E2 7E 20 A0 01 1F
7E56 A0 FC 02 20 05 FE A0 01
7E5E 1E 02 A0 04 F8 62 F8 02
7E66 A0 04 03 36 03 37 00 02
```

**Listing 4** *(Continued)*

```
7E6E A0 09 01 1F EC 77 63 46
7E76 63 46 63 77 FB 01 1F F7
7E7E E3 F8 62 F8 62 F8 E3 F7
7E86 01 1F 02 A0 09 01 1E 02
7E8E A0 0B 03 35 03 37 00 A0
7E96 A0 E4 EF E4 EF E4 A0 A0
7E9E 01 1F 69 6C 62 62 79 62
7EA6 62 7B 5F 01 1F DF 63 77
7EAE 78 77 77 63 63 E9 01 1F
7EB6 A0 A0 F7 F8 F7 F8 F7 A0
7EBE A0 01 1E 02 A0 0B 03 35
7EC6 03 37 00 A0 A0 EF F9 EF
7ECE F9 EF A0 A0 01 1F A0 69
7ED6 6C F8 F8 F8 7B 5F A0 01
7EDE 1F E7 02 20 07 E5 01 1F
7EE6 A0 DF 77 F9 EF F9 77 E9
7EEE A0 01 1E 02 A0 03 F7 F8
7EF6 62 F8 F7 02 A0 03 03 36
7EFE 03 37 00 A0 EC 77 77 78
7F06 77 77 FB A0 01 1F A0 6C
7F0E 02 A0 05 7B A0 01 1F A0
7F16 79 20 63 63 63 20 79 A0
7F1E 01 1F A0 A0 A0 E3 A0 E3
7F26 A0 A0 A0 01 1E 02 A0 0B
7F2E 03 35 03 39 00 A0 EF F9
7F36 E2 F9 E2 F9 EF A0 01 1F
7F3E C5 F8 62 79 62 79 62 F8
7F46 C5 01 1F CA A0 A0 A0 F2
7F4E A0 A0 AC CB 01 1F 02 A0
7F56 09 01 1E 02 A0 0B 03 35
7F5E 03 39 00 ED EE 01 0B F0
7F66 FD 01 1A A0 C4 EE 01 07
7F6E F0 C4 A0 01 1D A0 C3 D2
7F76 C0 D2 C0 D2 C3 A0 01 20
7F7E 02 A0 07 00 CD A0 01 0B
7F86 A0 CE 01 1A CA C9 02 A0
7F8E 09 D5 CB 01 1D 9C 02 A0
7F96 07 AF 01 20 E3 C4 E3 C4
7F9E E3 C4 E3 00 A0 A0 D5 A0
7FA6 C9 A0 A0 01 22 F2 C6 A0
7FAE C6 F2 01 24 C5 E3 C5 00
7FB6 CE A0 D5 A0 C9 A0 CD 01
7FBE 22 ED D2 A0 D2 FD 01 24
7FC6 A0 E3 A0 00 03 26 03 54
7FCE 03 5A 00 03 25 03 42 00
7FD6 03 23 03 59 03 56 00 03
7FDE 24 03 55 03 2C 00 03 26
7FE6 03 55 03 43 00 03 5C 03
7FEE 59 03 4E 00 12 20 12 20
```

## Additional Readings

Paul Ekman and Wallace V. Friesen, *Unmasking the Face* (Prentice-Hall, 1975), and Carrol E. Izard, *The Face of Emotion* (Appleton-Century-Crofts, 1971).

---

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 plans to use the face program in a longer program for simulating social interaction. He may be contacted at the Department of Sociology, Indiana University, Bloomington, IN 47405.

# 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.

# 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 find), DOS 3.3 and a minimum of 32K only **$30.00**

**Sensible Software**

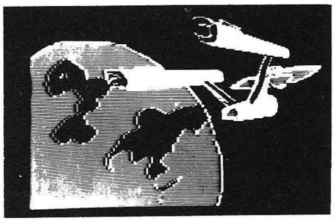
6619 Perham Drive Dept. M
West Bloomfield, Michigan
48033 • (313) 399-8877
Visa and Mastercard
Welcome
Please add $1.25
postage and
handling per
diskette

# VersaWriter & APPLE II:

## The Keys to Unlimited Graphics

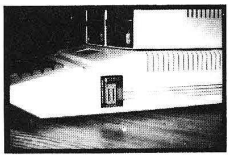
**DRAWING TABLET**
Although VersaWriter operates on a simple principle, it produces graphics which match or exceed those of other digitizers. Rugged construction, translucent base, easy to use — plugs directly into APPLE II.

**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'.

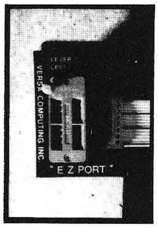
**Sound familiar?**

# 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**

# A Low-Cost Digitizer for the Apple

### by Jay Sinnett

**If you have a computer with high-resolution graphics, then you already own 99% of a digitizer with equal resolution. This article tells you how to use a sheet of half-reflecting plastic to build the rest. The modular demonstration program should be easy to customize to your application. The article also describes a "crashproof" disk access which should be of interest to those who have puzzled over details of the Applesoft ONERR GOTO command.**

**Digitizer Demo**
requires:

Apple II or Apple II Plus
with 48K
One disk drive
Applesoft BASIC

Although **Digitizer Demo** is implemented on the Apple II, the method described here can be implemented on any microcomputer offering high-resolution graphics

A digitizer is a device used to translate pictures into sets of X and Y coordinates for use in a computer. Digitizers are used in research for measuring the size, shape, or position of objects that have been photographed or videotaped, reading strip chart recordings, etc. Hobbyists might use digitizers to copy shapes into the computer's memory from drawings made with pencil and paper. Unfortunately, most digitizers available for use with popular microcomputers cost several hundred dollars. In this article, I will describe one that can be built for under $20.00 and can be used with any microcomputer having high-resolution graphics. A sample program written for the Apple II[1] demonstrates the simplicity of the software required to implement the digitizer.

Most inexpensive digitizers work this way: the operator moves a mechanical pointer around on a photo or drawing while its X and Y coordinates are measured electrically and sent to the computer. Usually the pointer has to be in physical contact with the picture so that the operator's perspective won't cause parallax errors. In this digitizer the pointer is actually a cursor on the Apple's high-resolution screen. This is optically superimposed onto the photo, drawing, or flat object to be digitized using a half-silvered ("two-way") mirror. When this is done correctly, there is no parallax error because both images are optically in the same place.

To build the digitizer, I first placed two identical monitors facing each other, about 10" apart, with a sheet of window-plastic exactly halfway between them and parallel to the monitor screens, as shown in figure 1. This worked, but plastic alone reflected weakly, and there was an image from both near and far sides, which was distracting. Then I obtained some half-silvered two-way mirror plastic film and applied it to the near side of the window: success! Both direct and reflected images were equally strong, and only one reflection was visible.

The reflecting plastic I used was secondhand and did not form a perfectly smooth surface. Because the uneveness

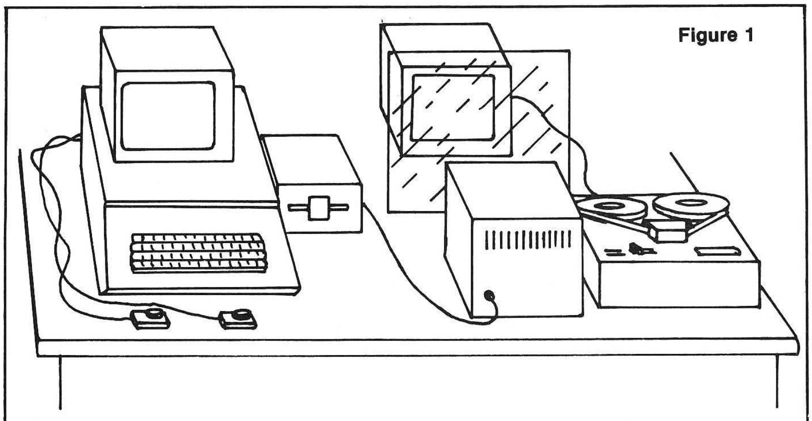


Figure 1

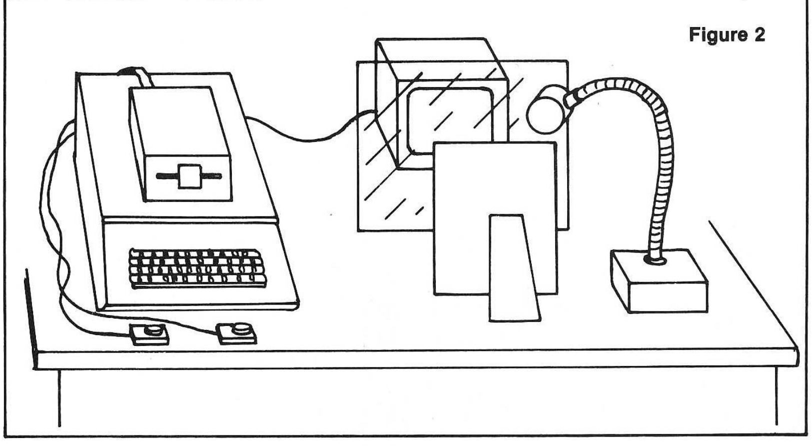


Figure 2

caused by tiny bubbles in the plastic distorted the reflected image, I found it best to look through the mirror at the video image, while using the reflected image of the Apple monitor. (A slight distortion of the cursor on the reflected Apple monitor seemed to be less distracting than distortion of the picture being digitized.) This arrangement also let me see the picture without a left-right reversal caused by mirror reflection. A third monitor was used for normal Apple monitor functions, so that the one being used for digitizing could be left in place once it was aligned.

If you want to digitize photos, drawings, etc., and you don't have extra monitors, I recommend you follow the layout of figure 2. In this setup you can read your Apple monitor normally through the glass. New one-way plastic carefully applied to a clean pane of window glass should give you acceptably clear reflections of the picture you are working on; if not, you can get a glass one-way mirror. The one-way plastic is sold in nationwide automotive parts and customizing shops; a 20-inch by 10-foot roll costs $10 - $15. One-way glass is more expensive; an 18-inch by 18-inch pane will set you back $20 -$30. A reading lamp may be needed to balance the intensities of the two images.

Before you can begin digitizing, the images must be aligned correctly, and the best tools are your eyes. First, set up the equipment as shown in figure 2. Be careful to have the mirror, monitor, and picture vertical, not leaning either way. Run the DIGITIZER DEMO program and use the paddles to align the cursor from the Apple monitor directly over the reflection of some spot on the picture. A small dot that just fits within the cursor works well. Now lean your head close to the mirror. With the aid of binocular (both eyes) depth perception, you should be able to see whether the cursor or the spot appears to be closer to you. Move either the picture or the monitor until the reflected image appears to be exactly the same distance from you as the non-reflected one. (This is easier than it sounds!) Check two or three points to be sure the angles are right.

With precise alignment, the image of a stationary cursor won't move with respect to the picture even when you move your head. In fact two people can look at the setup at once and see the same thing! The principle behind this position-independent superimposition is the optical law that states: "the angle of incidence is equal to the angle of reflection." This law will work even if the monitor screen and the picture are not parallel to the glass. Just be sure that the reflection lines up exactly with the direct image. In figure 3, as long as D1 = D2 and Θ is the same on both sides, the eye can be moved freely and the images will remain superimposed.

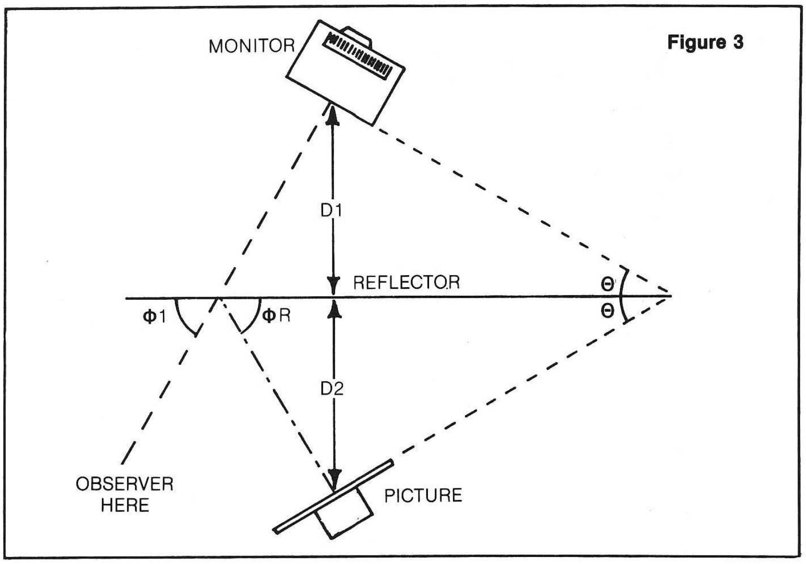


**Figure 3**

Obviously, the size of the picture you can digitize is limited by the size of your monitor screen, and the resolution is set by the Apple's 280 by 192 high-resolution grid. Most television monitors introduce some distortion to both horizontal and vertical axes, squeezing the picture in some places and stretching it in others. To remedy this, calibrate your own particular monitor by digitizing many points from an accurate target grid (such as graph paper). Then calculate correction factors to map the monitor's coordinates into an accurate array.

The digitizer demonstration program presented in listing 1 was designed to allow easy modification or expansion. It is written in structured format with several small blocks of code, each performing only one well-defined function, much like procedures in Pascal.

| Lines | |
|---|---|
| 1000-1240 | Initialize variables. |
| 1700-1850 | Load a binary file which defines cursor shapes. |
| 2000-2050 | Read paddle values. Scale to be within the range 0-1. |
| 3000-3070 | Translate paddle values to X and Y coordinates for display. Use offset and scale factors. |
| 4000-4080 | Draw cursor on screen at selected location. Remember location. |
| 4500-4550 | Erase cursor from remembered screen location. |
| 5000-5050 | Print current cursor location on screen in text window. |
| 6000-6090 | Read and debounce buttons. |
| 7000-7560 | Toggle from coarse to fine motion control and back again. |
| 8000-8050 | Pseudo output routine... print cursor location on screen in text windows. |
| 9000-9050 | Check keyboard for any activity. |
| 20000-20160 | Main routine. |

Lines 1060, 1070, and 1180 extract the address of a BLOADED shapefile from the DOS file pointer[2] (of a 48K Apple) and copy it into the Applesoft shapefile pointer. (This technique can be helpful in shape-generating programs, too.)

Lines 1700-1850 are a crashproof version of a BLOAD command that returns an error flag, ER, if the specified file is not on disk. If there is any kind of error in reading file SF$ from the disk in line 1780, control will immediately be passed to line 1800. This is the function of the Applesoft ONERR GOTO 1800 command in line 1770. The Applesoft manual[3] warns you that ONERR GOTO has a bug and gives you a machine-language patch to fix it. This is exercised by the CALL 768 in line

1840 (which assumes that the patch has been placed into memory as instructed in the manual). When the Apple returns from this call, the return address for the most recent unfinished GOSUB has been discarded. I use a dummy GOSUB in line 1730, so it can be thrown away if there is an error. A normal BLOAD will result in line 1790 RETURNING to line 1740 but the error routine ends with a GOTO 1740 instead of a RETURN. Both the normal and the error exits from the BLOAD routine execute lines 1740 and 1750.

Control of cursor motion can be accomplished in at least two ways. I have used the paddles for cursor motion (lines 1000-3070), but I have seen perfectly good cursor motion controls implemented on a PET with just the standard keyboard. In this program I provide both coarse and fine motion control for two reasons. First, the 256 possible values for the X-axis paddle (0) cannot be translated into 280 values for the screens X-axis without leaving out some points, whereas only 192 of the possible 256 Y-axis paddle (1) values are used. Second, it is easier to make fine position adjustments when the paddles are less sensitive and operate over a small section of the screen.

The mode is controlled by the button on paddle 0, which makes the program toggle back and forth between coarse and fine resolution every time it is pressed. The apparent center of the fine-tuning square is placed exactly where the coarse cursor was when the button was pressed, and different cursors are used for each mode so that the operator knows which mode he is using. The button on paddle 1 triggers an output subroutine that could send a value to disk or tape (but is displayed on the screen in this demonstration).

There are no IF statements in the routines that draw and erase the cursor, and the only IF statements in the paddle-to-screen translation are for range checking. The coarse/fine switching is all accomplished by resetting variables for screen origin (XO,YO), scale factor (XM,YM), paddle shift (FT), and cursor shape (CU) (lines 7000-7560). I have found this technique of modifying variables to set flags and to pass parameters is much easier to write and debug than a maze of IF statements. It forces you to plan your program before you start writing, but the effort is repaid many times over.

Whenever I plan to use high-resolution graphics, I first reset the Apple's memory pointers to protect the

**Listing 1**

```
10  GOTO 20000: REM MAIN PROGRAM (STRUCTURED PROGRAM HAS SUBROUTINES FIRS
     T)
20  REM ***************
30  REM
40  REM    DIGITIZER DEMO REV 1
50  REM
60  REM  BY JAY C. SINNETT
70  REM        JULY 8, 1981
80  REM  U.S. EPA ENVIRONMENTAL RESEARCH LABORATORY
90  REM  SOUTH FERRY ROAD
100  REM NARRAGANSETT, RI 02882
110  REM
120  REM **************
1000  REM
1010  REM     INITIALIZATION ROUTINE
1020  REM
1030  TEXT : HOME
1040 CD$ =  CHR$ (4): REM CONTROL-D
1050 KA = 49152: REM KEYBOARD ADDRESS
1060 BA = 43634: REM ADDRESS OF DOS POINTER TO BLOADED FILE (48K APPLE)
1070 PL = 232: REM POINTER TO SHAPE TABLE USED BY APPLESOFT
1080 S0 = 49249:S1 = 49250: REM ADDRESSES FOR GAME CONTROL BUTTONS
1090 XO = 0:YO = 0: REM OFFSET (FLOATING ORIGIN)
1100 XH = 279:YH = 191: REM HIGHEST VALUES FOR SCREEN X,Y
1110 XF = 64:YF = 64: REM SCALE FACTOR FOR FINE MOTION
1120 XM = XH:YM = YH: REM MULTIPLIER (EITHER FULL SCREEN OR FINE MOTION)
1130  PRINT "WHAT IS THE NAME OF YOUR CURSOR SHAPE TABLE ";
1140  INPUT "(RETURN FOR BUGCURSOR)? ";FI$
1150 FI = 0:SF$ = "BUGCURSOR": IF  LEN (FI$) <  > 0 THEN SF$ = FI$:FI = 1

1160  GOSUB 1700: REM SAFE BLOAD
1170  IF ER THEN  PRINT "SORRY, CAN'T FIND "SF$: PRINT CD$"CATALOG": GOTO
     1130
1180  POKE PL, PEEK (BA): POKE PL + 1, PEEK (BA + 1): REM SET SHAPETABLE
     POINTER
1190 CC = 2:CF = 3: IF  NOT FI THEN 1220: REM DEFAULT IS BUGCURSOR SET
1200  INPUT "WHICH CURSOR NUMBER DO YOU WANT TO USE  FOR COARSE POSITION?
      ";CC
1210  INPUT "WHICH CURSOR DO YOU WANT FOR FINE       POSITIONING? ";CF
1220 CU = CC: REM START WITH COARSE CURSOR
1230  HOME : SCALE= 1: ROT= 0: HGR
1240  RETURN
1700  REM
1710  REM      BLOAD SF$
1720  REM
1730  GOSUB 1760: REM DUMMY CALL FOR ERROR TRAP TO WORK
1740  POKE 216,0: REM TURN OFF ERROR PROCESSING
1750  RETURN : REM TO MAIN CALLING ROUTINE
1760 ER = 0: REM ERROR FLAG RETURNED TO CALLING ROUTINE
1770  ONERR  GOTO 1800
1780  PRINT CD$"BLOAD"SF$
1790  RETURN : REM SUCCESSFUL
1800  REM
1810  REM    ERROR TRYING TO LOAD FILE
1820  REM
1830 ER =  PEEK (222): REM ERROR CODE, IF YOU WANT
1840  CALL 768: REM STRAIGHTEN OUT SUBROUTINE STACK, REMOVING ONE RETURN
     ADDRESS
1850  GOTO 1740
2000  REM
2010  REM     READ PADDLES, SCALE 0 TO 1
2020  REM
2030 XP =  PDL (0) / 255
2040  FOR N = 1 TO 10: NEXT : REM DELAY FOR PDL CIRCUIT
2050 YP =  PDL (1) / 255
3000  REM
3010  REM    TRANSLATE PDL VALUES
3020  REM
3030 X = XO +  INT ((XP - FT) * XM): IF X < 0 THEN X = 0
3040  IF X > XH THEN X = XH
3050 Y = YO +  INT ((YP - FT) * YM): IF Y < 0 THEN Y = 0
3060  IF Y > YH THEN Y = YH
3070  RETURN
4000  REM
4010  REM    PLACE CURSOR ON SCREEN
4020  REM
4030 XS = XH - X: REM REVERSE LEFT/RIGHT
4040 YS = YH - Y: REM ORIGIN AT BOTTOM OF SCREEN
4050  HCOLOR= 3
4060  DRAW CU AT XS,YS
4070 X1 = X:Y1 = Y: REM REMEMBER COORDINATES USED LAST
4080  RETURN
4500  REM
4510  REM    ERASE CURSOR FROM SCREEN
4520  REM
4530  HCOLOR= 0
4540  DRAW CU AT (XH - X1),(YH - Y1)
4550  RETURN
5000  REM
5010  REM    DISPLAY CURRENT CURSOR LOCATION
```

*(Continued)*

**Listing 1** *(Continued)*

```
5020  REM
5030  VTAB 22: PRINT "                                         ": REM ERASE LAST
      DATA
5040  VTAB 22: PRINT "X = "X"  Y = "Y
5050  RETURN
6000  REM
6010  REM     READ BUTTONS (DEBOUNCING)
6020  REM
6030 F0 = 0:F1 = 0
6040 SA =  PEEK (S0) > 127: REM 1 IF BUTTON 0 DOWN
6050 SB =  PEEK (S1) > 127: REM BUTTON 1
6060  IF SA AND  NOT SQ THEN F0 = 1: REM F0 IS FLAG FOR BUTTON 0 BEING PR
     ESSED
6070  IF SB AND  NOT SR THEN F1 = 1: REM FLAG FOR BUTTON 1
6080 SQ = SA:SR = SB: REM REMEMBER PREVIOUS STATE
6090  RETURN
7000  REM
7010  REM    TOGGLE SCALE FACTOR, ORIGIN
7020  REM
7030  IF XM = XH  THEN 7500
7040  REM ELSE HERE TO CHANGE FROM FINE TO COARSE
7050 XO = 0:YO = 0: REM ORIGIN
7060 XM = XH:YM = YH: REM SCALE FACTOR
7070 CU = CC: REM CURSOR TYPE
7080 FT = 0: REM NO SHIFT OF PADDLE
7090  GOTO 7560 REM END ELSE
7500  REM HERE TO CHANGE FROM COARSE TO FINE MOTION
7510 XO = X
7520 YO = Y
7530 XM = XF:YM = YF: REM  SCALE
7540 CU = CF: REM CURSOR
7550 FT = .5: REM SHIFT PADDLES TO CENTERED PLUS/MINUS RANGE
7560  RETURN
8000  REM
8010  REM    OUTPUT ROUTINE...TO SCREEN FOR THIS DEMO
8020  REM
8030  VTAB 23: PRINT "                                        ": REM ERASE
     OLD POINT
8040  VTAB 23: PRINT "LAST 'PLOTTED' POINT: X = "X"  Y = "Y
8050  RETURN
9000  REM
9010  REM    CHECK KEYBOARD
9020  REM
9030 KF =  PEEK (KA) > 127: REM TRUE IF A KEY HAS BEEN PRESSED
9040  IF KF THEN  GET KB$: REM RETURN WITH IDENTITY OF KEY IN KB$
9050  RETURN
20000  REM
20010  REM      MAIN ROUTINE
20020  REM
20030  GOSUB 1000: REM INITIALIZE
20040  GOSUB 2000: REM GET DESIRED COORDINATES
20050  IF X1 = X AND Y1 = Y THEN  GOTO 20110
20060  REM IF COORDINATES HAVE CHANGED, ERASE OLD CURSOR AND PLACE NEW ON
      E
20070  GOSUB 4500: REM ERASE OLD CURSOR
20080  GOSUB 4000: REM PLACE NEW CURSOR
20090  GOSUB 5000: REM DISPLAY CURRENT LOCATION
20100  REM END IF
20110  GOSUB 6000: REM READ BUTTONS INTO F0, F1
20120  IF F0 THEN  GOSUB 4500: GOSUB 7000: GOSUB 4000: REM TOGGLE SCALE F
      ACTOR
20130  IF F1 THEN  GOSUB 8000: REM 'OUTPUT' A DATA POINT
20140  GOSUB 9000: REM CHECK KEYBOARD
20150  IF KF THEN  END : REM FOR THIS DEMO
20160  GOTO 20040
```

**Listing 2**

```
 800.882

0800- 03 00 08 00 1F 00 50 00
0808- 24 24 24 B4 92 D2 3F 3F
0810- 3F 4D 49 09 2D 2D 2D DF
0818- DB 13 36 36 16 24 00 DB
0820- 3F 3F 24 64 0D 18 0C 0D
0828- 18 2D 35 36 26 08 18 08
0830- 2D 0E 0E 0E 0E 36 FE 1B
0838- 2D AD 36 1E 1E 1E 1E 3F
0840- 27 24 B4 D2 3F 0F 18 0F
0848- 18 0F 18 0F 18 24 04 00
0850- 1B 3F 3F 6F 20 64 0D 18
0858- 2D 25 30 16 36 0C 18 08
0860- 2D 0E 0E 36 FE 2B 6D FD
0868- 8B 36 CE F3 1E 38 0F 18
0870- 08 18 36 36 26 C8 3B 0F
0878- 18 0F 18 0F 98 51 E2 1C
0880- 24 04 00
*
```

entire space from $800 to the top of page one high-resolution graphics at $4000. This leaves about 20K of contiguous program space and 6K of shapefile space available. The following series of POKEs should be used to accomplish this:

```
POKE 103,1
POKE 104,64
POKE 16384,0
```

The shape table used by this program is shown in monitor format (listing 2), occupying locations $800-$882. It actually has three cursors available: a crosshair and two different circle/crosshair combinations. The DIGITIZER DEMO program assumes that it is stored on disk under the name BUGCURSOR. If you use it as shown, you *must* reset the memory pointers before running DIGITIZER DEMO, or the program will be destroyed when the shapefile is loaded.

The hardware and software I have described here were originally developed for a government research project, but they should be useful for a great many other applications. Your output routine might simply connect points as you enter them, making a line drawing, or you might want to save points in a shapefile or in a disk data file. All sorts of graphics programming should be easier and more fun with this affordable digitizer.

## References

1. Mention of commercial products does not imply endorsement by the U.S. Environmental Protection Agency.

2. *The DOS Manual*, Apple Computer, Inc., Cupertino, CA 95014, 1980. Page 144 (DOS Entry Points).

3. *Basic Programming Reference Manual*, Apple Computer, Inc., Cupertino, CA 95014, 1978. Page 136 (Machine-Language program to clear up ONERR GOTO problem).

---

Jay Sinnett has a B.S. degree in electronics from MIT (1968). After three years in the Navy, he joined the Environmental Protection Agency and has been at his present position for seven years. He programs extensively in machine language and Microsoft BASIC on the SYM, PET, and Apple II at work. He may be contacted at the United States Environmental Protection Agency, Environmental Research Laboratory, South Ferry Road, Narragansett, RI 02882.

---

MICRO™

# ARK COMPUTING

**LOWERS THE BOOM ON HIGH PRICES**

## SPECIALS

| | |
|---|---|
| 16K Ram Board | **$79.95** |
| 80 Column Board (Wiz-80) | **$195.00** |
| Parallel Printer Interface (W/Cable) (compatible with Pascal, Basic, CP/M) | **$59.95** |
| Z-80 Softcard | **$259.95** |
| Joyport | **$59.95** |
| ARK Special (includes Wiz-80, Lazer Keyboard Plus, Lazer Lower Case Plus) | **$295.00** |
| On-Line Systems: General Manager | **$99.95** |
| Diskettes w/hubring (10) | **$19.95** |
| Hi-Res Secrets | **$84.95** |
| Expediter II | **$59.95** |
| The Dictionary | **$69.95** |
| Microsoft 16K Ram Card | **$99.95** |
| Time Zone | **$69.95** |
| Lazer Keyboard Plus | **$69.95** |
| Lazer Lower Case Plus | **$44.95** |
| Lazer Lower Case Plus II | **$19.95** |
| Anix 1.0 | **$34.95** |
| Lazer Pascal | **$29.95** |
| Anix-Pac (Anix, Pascal, Sources) | **$99.95** |
| Using 6502 Assembly Language | **$14.95** |
| Datamost Joystick | **$39.95** |
| Datamost Expandaport | **$49.95** |

## HARDWARE

| | |
|---|---|
| 12" hi-res green display | $285.00/**$159.95** |
| 12" lo-res color display | $430.00/**$349.95** |
| 12" hi-res RGB color display | $1095.00/**$895.95** |
| Apple Cat II | $389.00/**$329.00** |
| Apple Clock | $280.00/**$195.95** |
| Supertalker SD2000 | $199.00/**$149.95** |
| Romplus | $155.00/**$124.95** |
| Romwriter | $175.00/**$139.95** |
| Music System | $395.00/**$295.95** |
| A/D+D/A | $350.00/**$249.95** |
| Expansion Chassis (125v) | $750.00/**$495.00** |
| CPS Multifunction Card | $239.00/**$149.95** |
| RAM Plus+ (32 RAM board w/16K installed) | $189.00/**$139.95** |
| COPYROM | $55.00/**$44.95** |
| Keyboard Filter ROM | $55.00/**$44.95** |
| 13 Key Keypad (new or old) | $125.00/**$94.95** |
| Bar code reader | $195.00/**$145.95** |
| Softkey (15-key user-defin.) | $150.00/**$119.95** |
| Asynchronous Serial | $179.95/**$129.95** |
| Datamost Joystick | $59.95/**$39.95** |
| Datamost Expandaport | $69.95/**$49.95** |
| Micromodem | $379.00/**$269.00** |
| Smartmodem (RS-232) | $279.00/**$199.00** |
| Lower Case +(Rv.6 & earlier) | $64.95/**$44.95** |
| Lower Case + II (Rv. 7 & Ltr.) | $24.95/**$14.95** |
| Lower Case+ III (Rv. 7 & Ltr.) | $49.95/**$34.95** |
| Keyboard #Plus | $99.95/**$69.95** |
| Character Set+ Plus | $24.95/**$19.95** |
| Sup. 'R' Term 80-col. board | $395.00/**$295.00** |
| Sup. 'R' Switcher (pwr. suppl.) | $295.00/**$195.00** |
| A35 35-track disk drive w/o controller | $479.00/**$379.00** |
| A35 35-track disk drive w/ controller | $579.00/**$449.00** |
| A40 40-track disk drive w/o controller | $449.00/**$359.00** |
| A40 40-track disk drive w/ controller | $549.00/**$449.00** |
| A70 70-track disk drive w/o controller | $599.00/**$499.00** |
| A70 70-track disk drive w/ controller | $699.00/**$599.00** |

## BUSINESS SOFTWARE

| | |
|---|---|
| Mail Label & Filing System | $74.95/**$49.95** |
| Payroll | $395.00/**$295.00** |
| General Ledger(w/payables) | $495.00/**$395.00** |
| Home Money Minder | $34.95/**$24.95** |
| The Mail Room | $34.95/**$24.95** |
| CPA I General Ledger | $249.95/**$169.95** |
| CPA II Accounts Receivable | $249.95/**$169.95** |
| CPA III Accounts Payable | $249.95/**$169.95** |
| CPA IV Payroll | $249.95/**$169.95** |
| CPA V Property Manage. | $249.95/**$169.95** |
| The Home Accountant | $74.95/**$54.95** |
| Word Star | $375.00/**$195.00** |
| Supersort | $200.00/**124.95** |
| Mail Merger | $125.00/**$79.95** |
| Data Star | $300.00/**$195.00** |
| Spellstar | $200.00/**$124.95** |
| Calc star | $200.00/**$124.95** |
| Visicalc 3.3 | $250.00/**$179.95** |
| PFS (Personal Filing System) | $125.00/**$89.95** |
| PFS. Report | $95.00/**$69.95** |
| The Correspondent | $59.95/**$49.95** |
| Mailing List Database | $49.95/**$39.95** |
| Higher Text II | $39.95/**$29.95** |
| Directory Manager v.2 | $29.95/**$21.95** |
| Real Estate Analysis Pgm | **30% OFF** |
| PIE WRITER | **(call for details)** |
| Easy Mover 40-column | $49.95/**$39.95** |
| Easy Mailer 40-column | $69.96/**$49.95** |
| Easy Writer 40-column | $99.95/**$74.95** |
| Pro Easy Mailer | $175.00/**$124.95** |
| Pro Easy Writer | $249.95/**179.95** |
| Super Scribe II | $129.95/**$79.95** |

## GAMES

| | |
|---|---|
| Softporn Adventure (no graphics) | $29.95/**$19.95** |
| #0 Mission Asteroids | $19.95/**$14.95** |
| #1 Mystery House | $24.95/**$16.95** |
| #2 Wizard & Princess | $32.95/**$19.95** |
| #3 Cranston Manor | $34.95/**$24.95** |
| #4 Ulyesses & the Golden Fleece | $34.95/**$24.95** |
| #5 Time Zone | $99.95/**$69.95** |
| Hi-res Cribbage | $24.95/**$19.95** |
| Hi-res Soccer | $29.95/**$19.95** |
| Hi-res football | $39.95/**$29.95** |
| Sabotage | $24.95/**$19.95** |
| Jawbreaker | $29.95/**$19.95** |
| Threshold | $39.95/**$24.95** |
| Missle Defense | $29.95/**$19.95** |
| Crossfire | $29.95/**$19.95** |
| Pegasus | $29.95/**$19.95** |
| Warp Destroyer | $29.95/**$21.95** |
| Star Crusier | $24.95/**$19.95** |
| Adventure | $29.95/**$21.95** |
| Both Barrels | $24.95/**$19.95** |
| Cyber Strike | $39.95/**$29.95** |
| Phantoms Five | $29.95/**$21.95** |
| Space Eggs | $29.95/**$19.95** |
| Autobahn | $29.95/**$21.95** |
| Pulsar II | $29.95/**$21.95** |
| Orbitron | $29.95/**$21.95** |
| Gamma Goblins | $29.95/**$19.95** |
| Gorgon | $39.95/**$29.95** |
| Sneakers | $29.95/**$19.95** |
| EPOCH | $34.95/**$24.95** |
| Cops & Robbers | $34.95/**$24.95** |
| Outpost | $29.95**$19.95** |
| Dark Forest | $29.95/**$19.95** |
| Beer Run | $29.95/**$19.95** |
| Borg | $29.95/**$19.95** |
| Joy Port w/foosball | $74.95/**$59.95** |
| Hadron | $34.95/**$24.95** |
| Twerps | $29.95/**$19.95** |
| Computer Foosball | $29.95/**$19.95** |
| Wizardry | $49.95/**$39.95** |
| Galactic Attack | $29.95/**$21.95** |
| Minator | $34.95/**$24.95** |
| Oylmpic Decathlon | $29.95/**$21.95** |
| Three Mile Island | $39.95/**$29.95** |
| ABM | $24.95/**$19.95** |
| Robot Wars | $39.95/**$29.95** |
| Global War | $24.95/**$19.95** |
| Castle Wolfenstein | $29.95/**$21.95** |
| Falcons | $29.95/**$21.95** |
| Suicide | $29.95/**$21.95** |
| Grand Prix | $29.95/**$21.95** |
| The Best of Muse | $39.95/**$29.95** |
| Flight Simulator | $33.50/**$26.95** |
| Dungeon Campaign | $17.50/**$14.95** |
| Odyssey | $29.95/**$21.95** |
| Escape From Arcturus | $24.95/**$19.95** |
| Palace in Thunderland | $24.95/**$19.95** |
| MAD Venture | $24.95/**$19.95** |
| Roach Motel | $34.95/**$26.95** |
| English SAT #1 | $29.95/**$21.95** |
| U.S. Constitution | $29.95/**$21.95** |

| | |
|---|---|
| Upper Reaches of Apshai | $19.95/**$15.95** |
| The Keys to Acheron | $19.95/**$15.95** |
| Datestones of Ryn | $19.95/**$15.95** |
| Morloc's Tower | $19.95/**$15.95** |
| Ricochet | $19.95/**$15.95** |
| Starfleet Orion | $24.95/**19.95** |
| Invasion Orion | $24.95/**$19.95** |
| The Dragon's Eye | $24.95/**$19.95** |
| Tues. Morning Quarterback | $29.95/**$21.95** |
| Jabbertalky | $29.95/**21.95** |
| Sorcerer of Siva | $29.95/**$21.95** |
| Rescue at Rigel | $29.95/**$21.95** |
| Crush, Crumble, and Chomp | $29.95/**$21.95** |
| Temple of Apshai | $39.95/**$29.95** |
| Hellfire Warrior | $39.95/**$29.95** |
| Star Warrior | $39.95/**$29.95** |
| Hi-res golf | $29.95/**$21.95** |
| Race for Midnight | $29.95/**$21.95** |
| Midnight Music | $34.95/**$24.95** |
| Star Blazer | $31.95/**$24.95** |
| Apple Panic | $29.95/**$19.95** |
| Space Quark | $29.95/**$19.95** |
| Red Alert | $29.95/**$19.95** |
| Track Attack | $29.95/**$19.95** |
| Genetic Drift | $29.95/**$19.95** |
| Arcade Machine | $44.95/**$29.95** |
| Raster Blaster | $29.95/**$21.95** |
| Trilogy of Games | $29.95/**$21.95** |
| Space Album | $39.95/**$29.95** |
| Fender Bender | $24.95/**$19.95** |
| 3-D Graphics | $39.95/**$29.95** |
| Akalabeth | $34.95/**$24.95** |
| Appleoids | $29.95/**$21.95** |
| Ulitma | $39.95/**$29.95** |
| LA Land Monopoly | $29.95/**$21.95** |
| Hyperspace Wars | $29.95/**$21.95** |
| 3-D Skiing | $24.95/**$19.95** |
| Torpedo Terror | $24.95/**$19.95** |
| Computer Bingo | $24.95/**$19.95** |
| Kaves Karkhan | $49.95/**$34.95** |
| Dragon Fire | $49.95/**$34.95** |
| Rings of Saturn | $39.95/**$29.95** |
| Alkemstone | $39.95/**$29.95** |
| Snack Attack | $29.95/**$19.95** |
| Casino | $39.95/**$24.95** |
| Thief | $29.95/**$19.95** |
| County Fair | $29.95/**$19.95** |
| Swashbuckler | $34.95/**$24.95** |
| Firebird | $29.95/**$19.95** |
| Russki Duck | $34.95/**$25.95** |
| Horizon V | $34.95/**$25.95** |
| Sargon II | $34.95/**$25.95** |
| Reversal | $34.95/**$25.95** |
| Zork | |
| Zork II | $39.95/**$29.95** |
| Pool 1.5 | $34.95/**$24.95** |
| Shuffleboard | $29.95/**$21.95** |
| Trick Shot | $39.95/**$29.95** |
| Crossword Magic | $49.95/**$39.95** |
| Master Type (hi-res) | $39.95/**$29.95** |
| Dogfight | $29.95/**$21.95** |
| Crown of Arthain | $34.95/**$26.95** |

## UTILITIES

| | |
|---|---|
| **Hi-Res Secrets (D. Fudge)** | **$124.95/$84.95** |
| Super Shape Drg & Animate | $34.95/**$24.95** |
| The Creative Tool Box | $44.95/**$34.95** |
| Applesoft Compiler | $175.00/**129.95** |
| Datadex | $150.00/**$99.95** |
| Using 6502 Assembly Lang. | $19.95/**$14.95** |
| ANIX (UNIX-like Oper. sys.) | $49.95/**$39.95** |
| Lazer Pascal | $39.95/**$29.95** |
| DOSOURCE 3.3 | $39.95/**$24.95** |
| Painter Power | $39.95/**$29.95** |
| The Voice | $39.95/**$29.95** |
| E-Z Draw | $49.95/**$39.95** |
| Pascal Graphics Editor | $99.95/**$69.95** |
| ACE | $39.95/**$29.95** |
| Apple-DOC | $49.95/**$39.95** |
| List Master | $39.95/**$29.95** |
| ASCII Express | $79.95/**$59.95** |
| Z-Term (req's Z-80 Card) | $99.95/**$69.95** |
| Z-Term Pro | $149.95/**$99.95** |
| Speed Star | $134.95/**$99.95** |
| On-Line | $89.95/**$69.95** |
| DB Master | $229.00/**$159.95** |

## UTILITIES

| | |
|---|---|
| LISA v2.5 | $79.95/**$59.95** |
| LISA Educational Pack | $119.95/**$79.95** |
| Speed/ASM | $39.95/**$29.95** |
| Expediter II | $99.95/**$59.95** |
| Disk Organizer II | $29.95/**$21.95** |
| Applesoft Plus | $24.95/**$19.95** |
| Applesoft Optimizer | $24.95/**$19.95** |
| Disk Recovery | $29.95/**$21.95** |
| Multi-disk Catalog | $24.95/**$19.95** |
| Back it up | $59.95/**$39.95** |
| Image Printer | $29.95/**$29.95** |
| Pascal Lower Case | $24.95/**$19.95** |
| Dos Plus | $24.95/**$19.95** |
| The Original Quick Loader | $24.95/**$19.95** |
| Typing Tutor | $24.95/**$19.95** |
| Fortran | $195.00/**$139.95** |
| A.L.D.S. | $125.00/**$89.95** |
| Basic Compiler | $395.00/**$295.00** |
| Cobol | $750.00/**$495.00** |
| TASC | $175.00/**$129.95** |
| Data Plot | $59.95/**$44.95** |
| DB Utility Pack | $99.00/**$69.95** |

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.

Apple is a registered trademark of APPLE COMPUTER INC.

YOUR SALVATION IN THE SEA OF INFLATION

**ARK COMPUTING**

P.O. BOX 2025
CORONA, CALIFORNIA 91720
**(714) 735-2250**

# AIM Logic Trainer

by Larry Kollar

**Test your logic circuits with this program. It allows you to control the inputs and send the outputs to LEDs or printer.**

**Logic Trainer**
requires:
AIM 65
and your logic circuit
Information is provided to aid conversion for other 6502 computers

In the course of a hardware project, sooner or later you need to test it. If your breadboard comes with a power supply, debounced switches, and lights, it's simple enough.

However, most of us have to debounce switches and put limiting resistors in series with our LEDs. If your project uses several IC packages, you may not have room to add the necessary circuitry.

Listing 1 is the assembly language program of my computerized circuit tester. Just attach some wires to a 44-pin connector (see table 4 for connections), plug the connector onto J1 and attach the circuit. (Do this with the AIM turned off.) Turn on the computer, load the program, and start it at $0200. The display shows eight switches (all initially OFF) as '1' = ON, '0' = OFF. Eight outputs (lights) are represented by '*' = ACTIVE, '–' = INACTIVE. (See listing 2.)

Now you can toggle any switch by hitting '1' through '8' on the AIM keyboard. '1' toggles the leftmost switch, '8' toggles the one on the far right. Hitting any switch twice in a row will return it to its original state. The following procedure enables you to add features that you probably couldn't find on any hardware laboratory trainer.

***Table 1:*** **Logic Trainer Commands**

| Command | Function |
|---|---|
| '1'..'8' | Toggle switches 1 through 8. |
| (space) | Single-step. |
| 'A' | Auto-step. |

***Table 2:*** **Delay Routine**

| Address | Initial Value | Effects of Change |
|---|---|---|
| $02B9 | $00 | To $80 will cut the delay by half (approximately 3/8 second), larger values will cut it more. |
| $02BB | $71 | For these locations, increasing the values will lengthen the delay. |
| $02C0 | 0B | $02BB is a "fine tune" value; $02C0 is more critical. |

***Table 3:*** **I/O Routines and VIA**

| Name | Function |
|---|---|
| CRLF | Output Carriage Return and Line Feed to the Display/Printer. |
| DE1 | Delay subroutine. |
| OUTPUT | Send the ASCII character in the accumulator to the Display/Printer. |
| READ | Get an ASCII character from the keyboard to the accumulator. |
| ORB | VIA Port B (lights). |
| ORA | VIA Port A (switches). |
| DDRB | VIA Port B data direction register. |
| DDRA | VIA Port A data direction register. |
| ACR | VIA Auxiliary Control Register. |
| PCR | VIA Peripheral Control Register. |

**Listing 1**

```
              ;* LOGIC TRAINER PROGRAM
              ;*
              ;*  BY LARRY KOLLAR
              ;*
              ;
              ;EXTERNAL REFERENCES:
              ;
              READ    EQU $E93C
              OUTPUT  EQU $E97A
              CRLF    EQU $E9F0
              TIMER   EQU $EC18
              PORTB   EQU $A000
              PORTA   EQU $A001
              DDRB    EQU $A002
              DDRA    EQU $A003
              T2L     EQU $A008
              T2H     EQU $A009
              ACR     EQU $A00B
              PCR     EQU $A00C
              PRIFLG  EQU $A411
                      ORG $200
                      OBJ $800
              ;
              ;INITIALIZE AND GET INSTRUCTION
              ;
0200 20 3B 02         JSR INIT        ;INITIALIZE VIA
0203 20 4C 02 RSTRT   JSR RESET       ;INITIALIZE SWITCHES
0206 20 52 02 START   JSR STATE       ;DISPLAY LOGIC STATES
0209 20 3C E9 GTCMD   JSR READ        ; AND GET A COMMAND
020C C9 20            CMP #$20        ;SPACE? (FOR 1-STEP)
020E F0 25            BEQ STEP1
0210 C9 41            CMP 'A          ;ELSE 'A'? (AUTOSTEP)
0212 F0 74            BEQ AUTO
0214          ;
0214          ;TOGGLE SWITCHES 1-8:
0214          ;
0214 38       TOGGL   SEC             ;OR TEST FOR TOGGLE
0215 E9 30            SBC #$30        ;ASCII CHAR TO VALUE
0217 30 F0            BMI GTCMD       ;<1? THEN TRY AGAIN
0219 C9 09            CMP #$09
021B 10 EC            BPL GTCMD       ;ALSO, IF >9 TRY AGAIN
021D 85 00            STA $00         ;NOW TURN IT AROUND
021F A9 09            LDA #$09        ; SO THAT '1' REFERS
0221 38               SEC             ; TO THE LEFTMOST ONE
0222 E5 00            SBC $00
0224 AA               TAX             ;USE THIS FOR COUNTER
0225 A9 00            LDA #$00
0227 38               SEC             ;PUT A BIT IN
0228 2A       OUT1    ROL A           ; AND FIND OUT WHERE
0229 CA               DEX             ; IT GOES
022A D0 FC            BNE OUT1
022C 4D 01 A0         EOR PORTA       ;TOGGLE THAT BIT
022F 8D 01 A0         STA PORTA       ; AND PUT IT IN
0232 4C 06 02         JMP START       ; AND GO BACK FOR MORE
0235          ;
0235          ;SINGLE-STEP SWITCHES
0235          ;
0235 EE 01 A0 STEP1   INC PORTA
0238 4C 06 02         JMP START
023B          ;
023B          ;INITIALIZE VIA
023B          ;
023B A9 FF    INIT    LDA #$FF
023D 8D 03 A0         STA DDRA        ;PORT A=SWITCHES
0240 A9 00            LDA #$00
0242 8D 02 A0         STA DDRB        ;PORT B=LIGHTS
0245 8D 0B A0         STA ACR         ;JUST MAKE SURE THESE
0248 8D 0C A0         STA PCR         ;DON'T DO ANYTHING
024B 60               RTS             ; AND GO BACK
024C          ;
024C          ;RESET ALL SWITCHES
024C          ;
024C A9 00    RESET   LDA #$00
024E 8D 01 A0         STA PORTA
0251 60               RTS
0252          ;
0252          ;UPDATE SWITCH/LIGHTS
0252          ;
0252 20 F0 E9 STATE   JSR CRLF        ;CLEAR DISPLAY
0255 AD 01 A0         LDA PORTA       ;GET THE SWITCHES
0258 A2 08            LDX #$08        ;LOOP FOR SWITCHES
025A 2A       SWOUT   ROL A           ;MOVE IT INTO CARRY
025B 48               PHA             ;SAVE A
025C A9 30            LDA '0          ;ASSUME SWITCH=OFF
025E 90 02            BCC *+4         ;IF CARRY, WE'RE WRONG
0260 A9 31            LDA '1          ;SO LOAD WITH '1'
0262 20 7A E9         JSR OUTPUT      ; AND PRINT THAT
0265 68               PLA             ;GET A BACK
0266 CA               DEX             ; AND KEEP GOING
0267 D0 F1            BNE SWOUT       ; UNTIL ALL 8 ARE OUT
0269 A2 04            LDX #$04        ;PUT IN 4 SPACES
026B A9 20            LDA #$20        ; BETWEEN SWITCHES AND
026D 20 7A E9 SPACE   JSR OUTPUT      ; THE LIGHTS
0270 CA               DEX
0271 D0 FA            BNE SPACE
```

**Listing 1** *(Continued)*

```
0273 AD 00 A0         LDA PORTB       ;NOW GET THE LIGHTS
0276 A2 08            LDX #$08        ; AND TREAT THEN THE
0278 2A       LITES   ROL A           ; SAME WAY AS SWITCHES
0279 48               PHA
027A A9 2D            LDA '-          ;EXCEPT OFF='-'
027C 90 02            BCC *+4
027E A9 2A            LDA '*          ; AND ON='*'
0280 20 7A E9         JSR OUTPUT
0283 68               PLA
0284 CA               DEX
0285 D0 F1            BNE LITES
0287 60               RTS
0288          ;
0288          ;AUTO-STEP SWITCHES:
0288          ;
0288 A2 00    AUTO    LDX #$00        ;ASK FOR NO. OF SWITCHES
028A 20 F0 E9         JSR CRLF
028D BD 01 00 PRMSG   LDA MSG,X
0290 20 7A E9         JSR OUTPUT
0293 E8               INX
0294 E0 12            CPX #$12
0296 D0 F5            BNE PRMSG
0298 20 3C E9 CNVRT   JSR READ        ;GET ANSWER
029B 38               SEC             ;MAKE ASCII INTO VAL
029C E9 30            SBC #$30
029E 30 F8            BMI CNVRT
02A0 C9 09            CMP #$09
02A2 10 F4            BPL CNVRT
02A4 AA               TAX             ; AND TREAT AS BEFORE,
02A5 A9 00            LDA #$00        ; WITH TOGGLE
02A7 38       FCNT    SEC
02A8 2A               ROL A
02A9 CA               DEX
02AA D0 FB            BNE FCNT
02AC AA               TAX             ;USE X FOR COMPARES
02AD A9 00            LDA #$00        ;INITIALLY ALL
02AF 8D 01 A0         STA PORTA       ; SWITCHES ARE OFF
02B2 8A       STEP    TXA             ;SAVE X
02B3 48               PHA
02B4 20 52 02         JSR STATE       ;DISPLAY STATES
02B7 68               PLA             ; AND GET X BACK
02B8 AA               TAX
02B9 AD 11 A4         LDA PRIFLG      ;SEE IF PRINTER IS ON
02BC 2A               ROL A           ; AND SKIP DELAY IF SO
02BD B0 12            BCS DONE
02BF A0 00            LDY #$00        ;3/4 SECOND DELAY
02C1 A9 71    DELAY   LDA #$71
02C3 8D 08 A0         STA T2L
02C6 A9 0B            LDA #$0B
02C8 8D 09 A0         STA T2H
02CB 20 18 EC         JSR TIMER
02CE C8               INY
02CF D0 F0            BNE DELAY
02D1 EC 01 A0 DONE    CPX PORTA       ;DONE INCREMENTING?
02D4 F0 06            BEQ TOTOP       ;IF SO, BACK TO TOP
02D6 EE 01 A0         INC PORTA       ;OTHERWISE STEP ONE
02D9 4C B2 02         JMP STEP        ; AND DISPLAY & DELAY
02DC 4C 03 02 TOTOP   JMP RSTRT
02DF          ;
02DF          ;
0001                  ORG $1
0001                  OBJ $800
0001          ;
0001 48 4F 57 MSG     ASC 'HOW MANY SWITCHES?'
0004 20 4D 41
0007 4E 59 20
000A 53 57 49
000D 54 43 48
0010 45 53 3F
```

***Listing 2:*** **Sample Output of the Logic Trainer Program**

```
HOW MANY SWITCHES?
00000000                    --------
00000001                    -------*
00000010                    -------*
00000011                    ------*-
00000100                    -------*
00000101                    ------*-
00000110                    ------*-
00000111                    ------**
```

The circuit being checked is an adder with carry-in capability. The inputs ($C_i$, X, Y) are displayed by switches 6, 7, and 8 respectively. The outputs ($C_o$, Z) are displayed by lights 7 and 8.

The user has turned on the printer, chosen the Auto-step command, and asked for three switches.

To single-step the switches, hit the space bar. Switch # 8 will toggle the fastest. To automatically step through any number of switches, type 'A' for auto-step. You will get the message:

HOW MANY SWITCHES?

Type a number between 1 and 8 (anything else is ignored). The switches are all reset to '0' and the program puts your circuit through its paces.

There is a ¾-second delay between steps, which can be changed according to the instructions in table 2. If the printer is turned on, the delay will be skipped. You can use the printer to generate a truth table (see listing 2 for a sample printout).

If you have a different 6502-based system, see table 3 for a list of system references. Any 6502 system can run this program if it has a VIA and at least 20 columns to display.

***Table 4:* Application Connector (J1) Wiring**

| Pin | Name | Function |
|---|---|---|
| A-1 | Ground | |
| A-2 | PA3 | Switch 5 |
| A-3 | PA2 | Switch 6 |
| A-4 | PA1 | Switch 7 |
| A-5 | PA4 | Switch 4 |
| A-6 | PA5 | Switch 3 |
| A-7 | PA6 | Switch 2 |
| A-8 | PA7 | Switch 1 |
| A-9 | PB0 | Light 8 |
| A-10 | PB1 | Light 7 |
| A-11 | PB2 | Light 6 |
| A-12 | PB3 | Light 5 |
| A-13 | PB4 | Light 4 |
| A-14 | PA0 | Switch 8 |
| A-15 | PB7 | Light 1 |
| A-16 | PB5 | Light 3 |
| A-17 | PB6 | Light 2 |
| A-A | +5V | |

My power supply is not the AIM standard and I know I have enough extra current to supply the circuits being tested. But if your supply is running very near its capacity, the milliamps might make a difference. An extra .5 amps should be plenty for most applications.

If you hit an invalid key, the program ignores it. However, the burden of getting the wiring right is entirely on the user. Don't short out the VIA! If you are careful, this program could be a useful tool for hardware development.

---

Larry Kollar is a senior at Michigan Technological University, majoring in Technical Communications with an option in Computer Science. Upon graduating in May of 1982, he plans to write and revise software manuals. Contact Mr. Kollar at 5500 Greenboro SE, Kentwood, MI 49508.

# Commodore Computers Bloom with CP/M® and Multi-User Capability.

## COMMODORE DEALERS

Expand your customer base by including the vast range of CP/M software development and applications!

Small Systems Engineering provides everything you need to enter the CP/M marketplace. Excellent dealer margins. Advertising and marketing support. CP/M software on Commodore formats. Corvus products. Watch for innovative products from SSE to support your Commodore Computer line.

**Now your Commodore can blossom into a full-blown CP/M-based or multi-user business system with the addition of 1 or 2 innovative peripherals. SOFTBOX alone gives you complete CP/M capability. HARDBOX interfaces to the Corvus Winchester disk drive, and multiplexing units, for high storage capacity and multi-user capability.**

## RUN CP/M WITH SOFTBOX

Add SOFTBOX. The Z80-based computer which connects to your Commodore PET or CBM through the IEEE bus.

Now you have the ability to run any CP/M software or application, giving you a universe of computing capabilities you could only wish for until now.

Simply by plugging the SMALL SYSTEMS SOFTBOX into your Commodore Computer's IEEE-488 port and loading the CP/M disk, your computer will run under the world's most popular disk operating system. No internal connections or modifications to your computer are necessary.

Applications packages designed to work with specific terminals (like Lear Siegler ADM3A, Televideo 912 or Hazeltine 1500) need no modification to work with your computer's screen, since the SOFTBOX allows your computer to emulate any of these devices.

Interfaces that come with your SOFTBOX let you add Corvus hard disk, with capabilities to 80 MBytes of storage. You can run a printer, and even to talk to another CP/M based system.

Available for immediate shipment. 90-day Warranty. Sugested retail $895

Call or write for our free CP/M Software Catalog.

## HARDBOX FOR LARGE STORAGE & MULTI-USER CAPABILITY

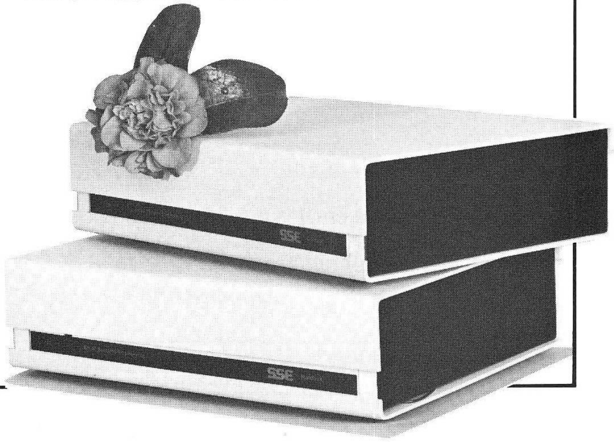
The HardBox introduces your PET software to the world of high-capacity, high speed disk storage. It acts as an intelligent controller for up to four Corvus Winchester disk drives. Each Corvus drive can have a capacity of 6, 10 or 20 million bytes.

The HardBox is designed to be software-compatible with PET DOS version 1 or 2, so that it will work with existing programs. The HardBox is designed to appear to the PET as a fast, high-capacity floppy disk unit.

When used with the Corvus range of drives, the HardBox provides much more than just a hard disk interface. By using an 8-way multiplexing unit, up to eight PET users, each equipped with a HardBox, may simultaneously address the same hard disk. By using two levels of multiplexing, up to 64 users may be accommodated. The hard disk may be divided logically into user areas.

A comprehensive set of utility programs are provided with the hardbox for file transfer, hard disk configuration, mirror backup, system maintenance etc.

Available for immediate shipment. 90 day Warranty. Suggested retail $695.



**SSE**

**SMALL SYSTEMS ENGINEERING**
222 B View St.
Mountain View, California 94041
**(415) 964-8201**

*CP/M is a registered trademark of Digital Research.*
*© 1982 Small Systems Engineering, Inc.*

# A Versatile Disk Label Printer

### by David Allen

**This disk label print program includes provisions for one- and two-sided disks, generating a different type of label for each.**

**Disk Label Print Program**
requires:

Apple II or Apple II Plus
Applesoft BASIC
Written for MX-80, but can be adapted to others.

I recently discovered that I had acquired a fairly large number of diskettes with a miscellaneous hodge-podge of programs on them. I finally bit the bullet and spent several hours rearranging them by category on dedicated diskettes. After all this work, I was totally dissatisfied with my diskette labelling system. I found I had a completely uncoordinated label system, with differing sizes, locations, pen colors, and unreliable information about the contents. So, I decided to create a utility program for printing diskette labels. This program is the result.

## Two Kinds of Labels

I know that there are reasons not to record on the back side of a single-sided diskette; I do it anyway. But it was inconvenient to have labels on both sides of the diskette. I needed one label on the front to identify the contents of both front and back. For those less frequent occasions when I leave the back vacant, I can use the full space of the label for just the information for that side. As a result, the program has two distinctly different formats for printing the labels.

## Screen Format of Input

To avoid printing lines too long for the format, I had the program print a video screen format so I could stay inside the lines. A degree of self-centering can be automatically developed.

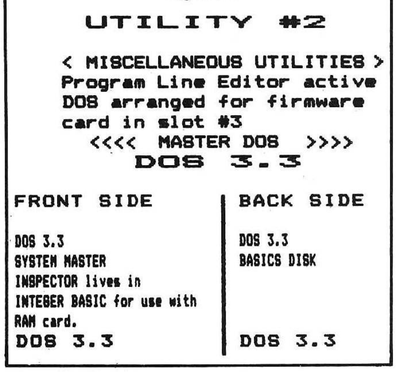
```
15  REM

FOR USE WITH KONZEN'S
PROGRAM LINE EDITOR LOADED FIRST
SO THAT UPPER/LOWER CASE MAY BE
USED.  OTHERWISE CHANGE LINES
3040 AND 3050 TO ELIMINATE THE
'&' COMMAND


20  TEXT : HOME : VTAB 10
30 N$ = "APPLE-DISK LABEL PRINTER": GOSUB 1000
40 N$ = "FOR 1 7/16 INCH BY 4 INCH LABELS": GOSUB 1000
50 N$ = "BY DAVID P. ALLEN": GOSUB 1000
60 N$ = "OCTOBER 7, 1981": GOSUB 1000
70  FOR X = 1 TO 1000: NEXT
80  POKE  - 16368,0:D$ =  CHR$ (13) +  CHR$ (4):VT$ =  CHR$ (27) +
     CHR$  (62) +  CHR$ (202) +  CHR$ (27) +  CHR$ (61): REM
VT$ PRINTS VERTICAL BAR
IN TRS80 BLOCK CHARACTERS


95 N1$ = "ENTER DATA FOR "
97 N3$ = "LABEL"
98 N2$(1) = "LEFT ":N2$(2) = "RIGHT "
100 F = 1
101  HOME
102 N$ = N1$ + N2$(F) + N3$: GOSUB 1000
110  PRINT "LINE #1: _____________": PRINT
120  PRINT "LINE #2: ___________________________": PRINT
130  PRINT "LINE #3: ___________________________": PRINT
140  PRINT "LINE #4: ___________________________": PRINT
150  PRINT "LINE #5: ___________________________": PRINT
160  PRINT "LINE #6: ___________________________": PRINT
180  VTAB 2: POKE 36,8: INPUT A$(F)
190  IF  LEN (A$(F)) > 13 THEN  GOSUB 2010: GOTO 180
200  VTAB 4: POKE 36,8: INPUT B$(F)
205 L = 2
210  IF  LEN (B$(F)) > 27 THEN  GOSUB 2020: GOTO 200
220  VTAB 6: POKE 36,8: INPUT C$(F)
225 L = 3
230  IF  LEN (C$(F)) > 27 THEN  GOSUB 2020: GOTO 220
240  VTAB 8: POKE 36,8: INPUT E$(F)
245 L = 4
250  IF  LEN (E$(F)) > 27 THEN  GOSUB 2020: GOTO 240
260  VTAB 10: POKE 36,8: INPUT F$(F)
265 L = 5
270  IF  LEN (F$(F)) > 27 THEN  GOSUB 2020: GOTO 260
280  VTAB 12: POKE 36,8: INPUT G$(F)
285 L = 6
290  IF  LEN (G$(F)) > 27 THEN  GOSUB 2020: GOTO 280
320  VTAB 16
330 N$ = "USING DOS 3.3? (Y/N-DEFAULT='YES')": GOSUB 1000
350  GET Y$
360  IF Y$ =  CHR$ (13) GOTO 400
370 DOS$(F) = "DOS 3.2"
380  GOTO 404
400 DOS$(F) = "DOS 3.3"
404  IF F = 2 THEN  GOTO 415
405  HOME : VTAB 10
407 N$ = "REVERSE SIDE USED? (Y/N)": GOSUB 1000
409  GET Y$
411  IF Y$ = "Y" THEN F = 2: GOTO 101
415  HOME
420  PRINT D$"PR#1"
423  PRINT  CHR$ (9) +  CHR$ (1);: REM
```

*(continued)*

```
RESETS PRINTER FLAG FROM
CONTROL-I TO CONTROL-A


425  PRINT  CHR$ (27); CHR$ (64);: REM

EPSON RESET COMMAND


427  IF F = 1 THEN  GOTO 600
430  PRINT  CHR$ (15); CHR$ (27); CHR$ (71);: REM

SETS CONDENSED PRINT
DOUBLE STRIKE


435  PRINT  CHR$ (27);"D"; CHR$ (29); CHR$ (32); CHR$ (0);: REM

SETS HORIZONTAL TABS


440  PRINT  CHR$ (14);A$(F - 1); CHR$ (20); CHR$ (1)"K": PRINT : PRINT
CHR$  (9);VT$;: PRINT  CHR$ (9); CHR$ (14);A$(F); CHR$ (1)"K": THEN 270
PRINT  CHR$ (9)VT$: REM


PRINTS FIRST LINE


450  PRINT B$(F - 1);: PRINT  CHR$ (9);VT$;: PRINT  CHR$ (9);B$(F): REM

PRINTS SECOND LINE


460  PRINT C$(F - 1);: PRINT  CHR$ (9);VT$;: PRINT  CHR$ (9);C$(F): REM

PRINTS THIRD LINE


470  PRINT E$(F - 1);: PRINT  CHR$ (9);VT$;: PRINT  CHR$ (9);E$(F): REM

PRINTS FOURTH LINE


480  PRINT F$(F - 1);: PRINT  CHR$ (9);VT$;: PRINT  CHR$ (9);F$(F): REM

PRINTS FIFTH LINE


490  PRINT G$(F - 1);: PRINT  CHR$ (9);VT$;: PRINT  CHR$ (9);G$(F): REM

PRINTS SIXTH LINE


510  PRINT  CHR$ (14);DOS$(F - 1); CHR$ (20); CHR$ (1)"K": PRINT : PRINT
      CHR$ (9);VT$;: PRINT  CHR$ (9); CHR$ (14);DOS$(F); CHR$ (1)"K": REM


PRINTS 'DOS' LINE


550  PRINT  CHR$ (27); CHR$ (64);: REM

RESETS PRINTER


560  PRINT D$"PR#0"
570  GOTO 3000
600  PRINT  CHR$ (27); CHR$ (69); CHR$ (27); CHR$ (71);: REM

SETS EMPHASIZED PRINT
DOUBLE STRIKE


605 T =  INT ((40 - ( LEN (A$(F)) * 2)) / 2)
610  PRINT  TAB( T) CHR$ (14)A$(F): PRINT : REM

SETS EXPANDED MODE FOR 1ST LINE


620  PRINT  TAB( 5);B$(F)
630  PRINT  TAB( 5);C$(F)
640  PRINT  TAB( 5);E$(F)
650  PRINT  TAB( 5);F$(F)
660  PRINT  TAB( 5);G$(F)
680  PRINT  TAB( 11); CHR$ (14);DOS$(F): REM
```

*(Continued)*

**Figure 1**



UTILITY #2

< MISCELLANEOUS UTILITIES >
Program Line Editor active
DOS arranged for firmware
card in slot #3
<<<< MASTER DOS >>>>
DOS 3.3

| FRONT SIDE | BACK SIDE |
|---|---|
| DOS 3.3<br>SYSTEM MASTER<br>INSPECTOR lives in<br>INTEGER BASIC for use with<br>RAM card. | DOS 3.3<br>BASICS DISK |
| DOS 3.3 | DOS 3.3 |

Since I was using an Epson MX-80 printer, I tried to include as much as I could in the label. Different print sizes, double strike, and even some graphics can be used.

Figure 1 shows the final result. The labels are neat and have room for more information than I could ever put on by hand. Here's how the program does it.

I set up a center-justified print routine in line 1000 that I used for the opening billboard and instruction screens. In lines 30-60, for example, I set N$ equal to the line of information and GOSUB to line 1000 to print it. Line 80 is a housekeeping line that defies the DOS command string and VT$, which issues the necessary ingredients to print the vertical divider on the two-sided labels. This is formatted for the MX-80 with Graftrax graphics ROMs installed. This line is formed by the TRS-80 graphics characters, which are also available without Graftrax if you know how to access them.

Lines 95-160 set up the video screen with a format to give me the exact length I can use for each of the seven possible label lines. Line 180 and the similar ones following POKE the cursor on the start of the underlined format. You can write a program line, such as 120, with all those underline strokes, even if your Apple doesn't have an underline key in two ways. The easier is Neil Konzen's Program Line Editor, which enables you to print an underline with an 'ESC 3' combination. If you don't have this utility, you can get a line of underscores by executing the following line from the immediate mode:

```
]FOR X = 1 TO 40: PRINT CHR$(95)
     ;:NEXT
```

Lines 180-290 input all the label information into a series to two-element string arrays (one element for each side of the diskette). The IF statements examine the length of each line you have typed in. If you have gone longer than the underscored line, it skips down to line 2020 for a new format line. The effect is magically to erase your offending line and put the cursor back at the beginning, waiting for you to try again. Notice 'CALL –868'. This little-used monitor routine clears the line from the cursor location to the right margin and removes the extra characters which did not fit on the line.

```
SETS EXPANDED MODE FOR DOS LINE

690  PRINT  CHR$ (27); CHR$ (64);
700  PRINT D$"PR#0"
710  GOTO 3000
1000 T = (40 - ( LEN (N$))) / 2: PRINT  TAB( T)N$: RETURN
2010  VTAB 2: PRINT "LINE #1: ________________";: CALL  - 868: RETURN
2020  VTAB (2 * L): PRINT "LINE #";L;": __________________________________"
      ;: CALL  - 868: RETURN
3000  HOME : TEXT : VTAB 12
3005 N$ = "TO CONTINUE, PRESS <SPACE BAR>": GOSUB 1000
3010 N$ = "TO END, PRESS <RETURN>": GOSUB 1000
3020 X =  PEEK ( - 16384)
3030  IF X <  > 160 AND X <  > 141 THEN  GOTO 3020
3040  IF X = 141 THEN  & : END
3050  & : RUN
```

Lines 350-405 tell the program which DOS line we want printed on the bottom of the label. 'F' in line 404 is a flag that we use to determine whether we are printing a single-sided or double-sided label. It is set by the routine in lines 405-411 and sends the program back for another batch of information pertaining to the back side of the diskette, if it is used.

Lines 415-700 offer two different print routines — one for each type of label we might wish to print. All of those 'CHR$' characters are understood by the MX-80 firmware; your Graftrax manual will help you identify them. If you are using another printer, you can insert your printer subroutines here in accordance with the protocol for your particular machine.

Lines 3000-end constitute a little routine that lets you sign off gracefully when you are finished with the program. It is simply a matter of reading the keyboard for your instructions and following through (the keyboard strobe was cleared in line 80). Line 3050 is used only if you have Konzen's program line editor to help you. It will permit you to enter your labels in upper and lower case, but PLE gets turned off when the printer slot is invoked in line 420. The '&' command in line 3050 turns PLE back on for use in creating your next label.

## Caution

*Never* try to scroll label stock *backwards* through your printer in order to save a few unused labels. There is a phosphor bronze pressure plate on the MX-80 which will strip off a back-sliding label and firmly implant it on the inaccessable innards of your printer platen assembly. This can mean an $80 repair bill to get the label out of the printer and to replace the phosphor bronze spring. Three or four saved labels are obviously not worth that risk.

David Allen may be contacted at 19 Damon Road, Scituate, MA 02066.

MICRO

# 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!

## 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!

**VIDEX:**

| | | | |
|---|---|---|---|
| 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 ASIO | 115 | Keyboard Co. Keypad | 129 |
| SSM Par. APIO | 99 | Sunshine Joystick | 40 |
| SSM A488 IEEE | 425 | Cisco Paddles | 29 |
| Novation APL. CAT | 339 | Prometheus Z Card | 229 |
| VERSAWRITER | 249 | SUP'R'MOD | 29 |

## OTHER SOFTWARE for APPLE:

**PERSONAL SOFTWARE:**

| | | | |
|---|---|---|---|
| Visicalc 3.3 | 159 | Visifile | 199 |
| Visiplot | 149 | Visiplot/Visitrend | 199 |
| Visiterm | 125 | Visidex | 169 |
| Desktop Plan II | 169 | Visipak | 559 |

**MICROSOFT:**

| | | | |
|---|---|---|---|
| APPLE Fortran (Z80) | 149 | APPLE Cobol (Z80) | 499 |
| TASC Basic Compiler | 139 | Olympic Declathon | 25 |

**MICRO-PRO:**

| | | | |
|---|---|---|---|
| Wordstar | 245 | Mail-Merge | 99 |
| Spell-Star | 149 | Data-Star | 199 |

**PEACHTREE: BIZ Packages, all** . . . . . . . . . . . . . . . . . . . . . 219

**CPA BIZ Packages, all** . . . . . . . . . . . . . . . . . . . . . . . . . . 199

## 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 | | |

## WORD-PROCESSORS for APPLE:

| | | | |
|---|---|---|---|
| Wordstar CP/M | 245 | EZWriter Prof. Sys. | 215 |
| Magic Wand CP/M | 299 | 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

## PRINTERS

**EPSON:**

| | | | |
|---|---|---|---|
| MX80 | 495 | MX80 F/T | 575 |
| MX100 w/Graftrax | 775 | MX70 w/Graftrax | 375 |
| APPLE Intfce/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 |

**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:** | | **DIABLO:** | |
| 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 |

## 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 326 KB | 559 |
| 160 Track 652KB | 799 | 8" A800 Controller | 549 |
| V1000 Dual 8" Subsystem | 1300 | V1000 Dual 8" Dbl. Side | 1600 |

**SORRENTO VALLEY ASSOC:**

| | | | |
|---|---|---|---|
| AMS 8000 Dual 8" | 1945 | AMS 5000 5 Mb. Drive | 2995 |
| AMS 8000 Dual 8", Dbl. Sd. | 2595 | AMS 5010 10 Mb. Drive | 3495 |
| | | Disk 2 + 2 Controller | 359 |
| ZVX4 Quad Controller | 495 | | |

**CORVUS:**

| | | | |
|---|---|---|---|
| 5 Mbyte | 3195 | 10 Mbyte | 4395 |
| 20 Mbyte | 5600 | Mirror | 675 |

## MONITORS & PLOTTERS

**ZENITH:**

| | | | |
|---|---|---|---|
| 12" Green | 125 | 13" Color | 375 |

**AMDEK:**

| | | | |
|---|---|---|---|
| 12" Green | 135 | 13" Color | 375 |

**WATANABE FLATBED PLOTTER:**

| | | | |
|---|---|---|---|
| 4 Pen | 1150 | 6 Pen | 1400 |

**STROBE CYLINDRICAL PLOTTER:**

| | | | |
|---|---|---|---|
| Plotter | 699 | APPLE Intfce | 75 |

**HOUSTON INSTR FLATBED PLOTTER:**

| | | | |
|---|---|---|---|
| DMP3 Smart Plotter | 1299 | 6-Pen Adaptor | 349 |

## 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 | 290/M Dual 5" Drive | 2125 |

**NEC:**

| | | | |
|---|---|---|---|
| PC8001A 32K Computer | 1099 | PC8012A I/O Unit | 699 |
| PC8031A Dual Drives | 1099 | CP/M 2.2 for NEC | 125 |
| Wordstar for NEC | 299 | Super Calc for NEC | 279 |

**COMMODORE:**

| | | | |
|---|---|---|---|
| VIC20 Computer | 259 | VIC1515 Printer | 315 |
| 8032 Micro Mainframe | 1100 | 8050 IMB Dual Disk | 1375 |
| 8250 2 MB Dual Drive | 1760 | 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 |

**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 DD | 2695 | 64K QD | 2995 |
| DSS 10 Meg | 3195 | Emulator | 725 |
| Intertube | 725 | | |

## 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 |

AMPEX Dialogue 80, 2-page, Detachable 995

IBM 3101-10 1199

## MODEMS

| | | | |
|---|---|---|---|
| CAT | 159 | DCAT | 169 |
| VENTEL 212-3, 1200 BD | 849 | AUTOCAT | 229 |
| LEX-11 | 149 | UNIVERSAL AUTO. ANS | 220 |
| UNIVERSAL ORIG/ANS | 180 | UNIVERSAL 1200 BD | 259 |

## S100 CARDWARE

| | | | |
|---|---|---|---|
| SSM MB64 64K Mem. | 775 | HAYES S100 MODEM | 349 |
| CCS 64K Dynamic | 429 | QT Clock/Cal | 135 |

## FLOPPY DISCS

| | | | |
|---|---|---|---|
| 3M 5" Soft SSSD/bx | 29 | 3M 8" Soft SSSD/bx | 33 |
| Elephant 5.25" Soft SSSD/bx | 25 | Elephant 8" 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 typestyles, 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 typestyle. 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 Stock-market 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.

LABELMAKER: 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 BONUSES 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**

# Program for Inverting a Matrix

*by Brian J. Flynn*

**Include this short BASIC routine in your mathematical and statistical programs.**

Matrix inversion is used in many business and statistical computer programs. For example, econometric techniques ranging from ordinary least squares to multiple linear regression analysis almost always use a matrix inversion routine. This article presents a short BASIC program for inverting a matrix. You can easily make part of it a separate module in one of your bigger programs. The algorithm, Gauss-Jordan Sweep without pivoting (named for mathematicians Carl Friedrich Gauss and Camille Jordan), quickly and accurately inverts most small matrices, when executed in double precision. Other routines, such as Gauss-Jordan Sweep with complete pivoting, generally produce more accurate solutions, but only by sacrificing program simplicity.

## Matrix Inversion

A matrix is a rectangular array of numbers or symbols. Only square matrices can be inverted, with the inverse being that matrix which, when multiplied by the original array, yields an identity matrix. An identity matrix contains 1s along the principal diagonal and 0s everywhere else. The principal diagonal runs like this: " \ ".

In the example in figure 1, matrix X denotes the original array. Its inverse is $X^{-1}$, and I is the identity matrix.

The trick of course, is to find $X^{-1}$. In the Gauss-Jordan Sweep procedure this is done by tacking an identity matrix onto the original array, and converting the original array into an identity matrix using elementary row and column operations. The right-side matrix that emerges is $X^{-1}$. This process is illustrated in figure 2.

```
100 REM INVERTING A MATRIX USING GAUSS-JORDAN SWEEP, WITHOUT PIVOTING
110 REM
120 REM  BY BRIAN J. FLYNN
130 REM  NOVEMBER 1981
140 REM
150 REM IF YOUR COMPUTER HAS DOUBLE-PRECISION ARITHMETIC,
160 REM   DEFINE ARRAY X AND VARIABLES P AND X AS SUCH
170 REM
180 REM K IS THE SIZE OR ORDER OF MATRIX
190 REM X IS THE DATA MATRIX
200 REM
210 REM INITIALIZATION
220 REM CLEAR SCREEN: PRINT"⊐"--PET, HOME--APPLE
230 PRINT"INVERTING A MATRIX USING GAUSS-JORDAN SWEEP, WITHOUT"
240 PRINT" PIVOTING."
250 PRINT
260 PRINT"PLEASE ENTER THE ORDER OF YOUR MATRIX, THAT IS, THE"
270 PRINT" NUMBER OF ROWS OR COLUMNS IN IT.  REMEMBER, HOWEVER,
280 PRINT" THAT ONLY SQUARE MATRICES CAN BE INVERTED."
290 PRINT
300 INPUT"ORDER = "; K
310 DIM X(K,2*K)
330 REM
340 REM ENTER DATA
350 REM
360 FOR I=1 TO K
370 REM CLEAR SCREEN
380 PRINT"PLEASE ENTER YOUR DATA."
390 PRINT"ROW    #";I;":"
400 PRINT
410 FOR J=1 TO K
420 PRINT"COLUMN #";J;
430 INPUT X(I,J)
440 NEXT J,I
460 REM
470 REM INVERT MATRIX
480 REM
490 REM TACK ON IDENTITY MATRIX
500 REM
510 FOR I=1 TO K
520 FOR J=1 TO K
530 IF J<>I THEN X(I,K+J) = 0
540 IF J=I THEN X(I,K+J) = 1
550 NEXT J,I
560 REM
570 REM INVERT MATRIX
580 REM
590 FOR I=1 TO K
600 REM ADJUST KEY ROW
610 P = X(I,I)
620 FOR J=1 TO 2*K
630 X(I,J) = X(I,J)/P
640 NEXT J
650 REM ADJUST REMAINING ROWS
660 FOR J=1 TO K
670 X = X(J,I)
680 FOR L = I TO 2*K
690 IF J<>I THEN X(J,L) = X(J,L) - X*X(I,L)
700 NEXT L,J,I
720 REM
730 REM PRINT MATRIX
740 REM
750 REM CLEAR SCREEN
760 FOR I=1 TO K
770 FOR J=1 TO K
780 REM USE PRINT USING, IF AVAILABLE, INSTEAD OF PRINT
790 PRINTX(I,K+J);" ";
800 NEXT J
810 PRINT
820 NEXT I
830 PRINT
```

---

Please address correspondence to the author at 1704 Drewlaine Dr., Vienna, VA 22180.

**Figure 1**

$$\underbrace{\begin{bmatrix}4 & 1\\ 6 & 2\end{bmatrix}}_{X}\ \underbrace{\begin{bmatrix}1 & -\frac{1}{2}\\ -3 & 2\end{bmatrix}}_{X^{-1}} = \begin{bmatrix}4\cdot1 + 1\cdot(-3) & 4\cdot(-\frac{1}{2}) + 1\cdot2\\ 6\cdot1 + 2\cdot(-3) & 6\cdot(-\frac{1}{2}) + 2\cdot2\end{bmatrix} = \underbrace{\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}}_{I}$$

**Figure 2**

To invert $\begin{bmatrix}4 & 1\\ 6 & 2\end{bmatrix}$ using Gauss-Jordan Sweep, without pivoting, the following steps are performed:

1. Tack an identity matrix onto the original array.

$$\begin{bmatrix}4 & 1\\ 6 & 2\end{bmatrix} \;\vdots\; \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

2. Convert the original array into an identity matrix using elementary row and column operations.

   a. Divide the entire first row of the matrix by 4, so that the upper left-hand element will equal 1:

$$\left[\begin{array}{cc:cc}1 & 0.25 & 0.25 & 0\\ 6 & 2 & 0 & 1\end{array}\right]$$

   b. Make the numeral 6 a 0 by multiplying the first row by 6 and subtacting the product from the second row:

$$\left[\begin{array}{cc:cc}1 & 0.25 & 0.25 & 0\\ 0 & 0.5 & -1.5 & 1\end{array}\right]$$

   c. Divide the entire second row by 0.5, so that the number 0.5 becomes a 1:

$$\left[\begin{array}{cc:cc}1 & 0.25 & 0.25 & 0\\ 0 & 1 & -3 & 2\end{array}\right]$$

   d. Finally, multiply the second row by 0.25 and subtract the product from the first row:

$$\left[\begin{array}{cc:cc}1 & 0 & 1 & -0.5\\ 0 & 1 & -3 & 2\end{array}\right] \quad \underbrace{\begin{matrix}1 & -0.5\\ -3 & 2\end{matrix}}_{X^{-1}}$$

**A harvest of savings from**

**Apple Tree Electronics**

**SOFTWARE**

**APPLE • ATARI • TRS80 • IBM**

A full line of software for business, games and education **up to 35% off!**

| | |
|---|---|
| MUSE | IUS |
| 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, 5¼", SS-DD | **$35.00** |
| Verbatim | Box of 10, 5¼", 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**

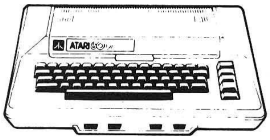
VISA

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

MasterCard

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.

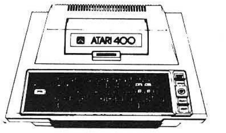
## ATARI



## 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 | $69.00 |
| Microtek 32K RAM | $139.00 |
| Ramdisk (128K) | $449.00 |
| One year extended warranty | $70.00 |
| INTEC 48K Ram | $249.00 |
| 481 Entertainer | $83.00 |
| 482 Educator | $130.00 |
| 483 Programmer | $57.00 |
| 484 Communicator | $344.00 |

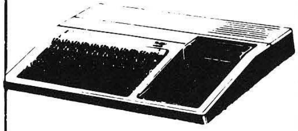


## ATARI 400

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

| | |
|---|---|
| Visicalc | $179.00 |
| Letterperfect | $109.00 |
| Ricochet | $14.50 |
| Crush Crumble & Chomp | $24.00 |
| Star Warrior | $29.00 |
| Rescue at Rigel | $24.00 |
| Datestones | $16.00 |
| Invasion Orion | $18.50 |
| Mission Asteroid | $22.00 |
| MouskATTACK | $31.00 |
| The Next Step | $34.00 |
| Softporn | $27.00 |
| Wizzard & Princess | $29.00 |
| K-BYTE Krazy Shoot Out (ROM) | $39.00 |
| Protector (Disk 32K) | 24.00 |
| 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 |
| Synapse | |
| File Manager 800 | $69.95 |
| Dodge Racer | $19.00 |
| Chicken | $24.00 |
| Slime | $24.00 |
| Nautilus | $24.00 |
| Disk Manager | $24.00 |
| Fort Apocalypse | $24.00 |
| Assembler | $39.00 |

## Texas Instruments



## TI-99/4A $299

| | |
|---|---|
| R.F. Modulator | $29.00 |
| 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 |
| 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 |

## Computer Covers

ATTRACTIVE COVERS FOR YOU COMPUTER AND DISK DRIVE.

| | |
|---|---|
| Atari 400 | $6.99 |
| Atari 800 | $6.99 |
| Atari 810 | $6.99 |

All Atari Covers are Beige.

| | |
|---|---|
| Commodore VIC-20 | $6.99 |
| Commodore 8032 | $14.99 |
| Commodore 8050/4040 | $10.99 |

All Commodore Covers are Roval Blue.

## Monitors

| | |
|---|---|
| Amdex 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 |
| Zenith ZVM 121 (Green) | $119.00 |

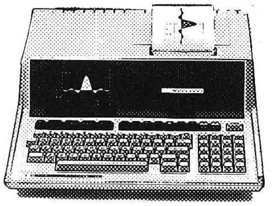
## Televideo

| | |
|---|---|
| 910 | $579.00 |
| 912C | $699.00 |
| 920C | $749.00 |
| 925C | $749.00 |
| 950 | $939.00 |

## Modems

| | |
|---|---|
| Livermore Star | $119.00 |
| Hayes | |
| Smart | $239.00 |
| Chronograph | $199.00 |
| Micromodem II | $279.00 |
| Micromodem 100 | $309.00 |
| Novation Auto | $239.00 |
| D Cat | $169.00 |
| Cat | $159.00 |

## HEWLETT PACKARD



## HP•85 $1899

| | |
|---|---|
| 80 Column Printer | $799.00 |
| HP•125 | $1999.00 |
| HP•83 | $1699.00 |
| HP•85 16K Memory Module | $169.00 |
| 5¼″ Dual Master Disc Drive | $1769.00 |
| HP•87 | $1769.00 |
| Hard Disk w/Floppy | $4349.00 |
| Hard Disk | $3440.00 |
| "Sweet Lips" Plotter | $1149.00 |

## HP41CV Calculator
## $239

| | |
|---|---|
| 41C | $189.00 |
| 11C | $104.00 |
| 12C | $114.00 |
| 34C | $114.00 |
| 38C | $114.00 |
| HP•41 Printer | $340.00 |

HPIL CALCULATOR PERIPHERALS

| | |
|---|---|
| IL Modual | $104.00 |
| Digital Cassette | $449.00 |
| Printer/Plotter | $419.00 |
| Card Reader | $164.00 |
| Optical Wand | $99.00 |

## Apple

Call for availability and prices
on all Apple computers and peripherals.

## Printers

| | |
|---|---|
| Centronics 739-1 | $519.00 |
| Centronics 739-3 | $619.00 |
| Diablo 630 Special | $1799.00 |
| Epson | |
| MX80 w/Graftrax | $449.00 |
| MX80FT | Call |
| MX100 | Call |
| NEC | |
| 8023 | $549.00 |
| 7730 | Call |
| 7720 | Call |
| 7710 | Call |
| Okidata | |
| 82A | $499.00 |
| 83A | $749.00 |
| 84 | $1129.00 |
| Citoh Starwriter | |
| F10-40 CPS | $1469.00 |
| F10-55 CPS | Call |
| Prowriter | $479.00 |
| Talley | |
| 8024-7 | $1399.00 |
| 8024-L | $1629.00 |
| IDS | |
| Paper Tiger | $Call |
| Prism | $Call |
| 2 Meter RS232-RS232 | $29.95 |

Cables Available For Most
Interfacing Purposes

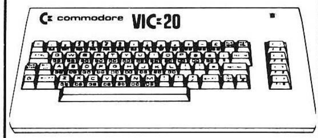
## commodore

## 8032 $1039

| | |
|---|---|
| 4032 | $969.00 |
| 4016 | $769.00 |
| 8096 Upgrade Kit | $399.00 |
| Super Pet | $1599.00 |
| 2031 | $529.00 |
| 8250 (Double Sided D. Drive) | $1699.00 |
| 5 Megabyte Hard Disk | $2399.00 |
| 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 |

SOFTWARE

| | |
|---|---|
| Commodore | Magis |
| BPI | Professional Software |
| Visicorp | Creative Software |



## VIC 20
## $249

Call for price and availability of VIC-64

| | |
|---|---|
| 16K VIC Expansion | $99.00 |
| 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 |
| Super Slot | $23.00 |
| VIC Avengers | $23.00 |
| Super Alien | $23.00 |
| Super Lander | $23.00 |
| Draw Poker | $23.00 |
| Midnite Drive | $23.00 |

# computer mail order

**east**
**800-233-8950**
**477 East Third Street**
**Williamsport, PA 17701**
**(717) 327-9575**
Patricio Habla Espanol

## CALL TOLL FREE
## HOW TO ORDER:

In-stock items shipped same day you call. No risk, no deposit on C.O.D. orders. Pre-paid orders receive free shipping within the continental United States with no waiting period for certified checks or money orders. All prices shown are cash prices add 3% for Mastercard and Visa. NV and PA residents add sales tax. All items subject to availability and price change.

**west**
**800-648-3311**
**P.O. Box 6689**
**Stateline, Nevada 89449**
**(702) 588-5654**
Franco Habla Espanol

MICRO™

# Reviews in Brief

Product Name: **Olympic Decathlon**
Equip. req'd: Apple II with 48K
Price: $29.95
Manufacturer: Microsoft Consumer Products
10700 Northrup Way
Bellevue, WA 98004

**Description:** Hi-res animation game requiring user competition against other players or against his own previous high scores. The program simulates all events of an actual decathlon. Algorithms for scoring seem well designed.

**Pluses:** Exceptional graphics. Good instructions are provided to teach the user how to play one of the most difficult games on the market.

**Minuses:** Very difficult physically. This game is definitely not for children. Good coordination and timing are required.

**Documentation:** A booklet is provided that includes in-depth instructions. It also gives clues for strategy, a rarity in the market. Grammatically correct and well written.

**Skill level required:** No programming experience required. Good coordination is a must.

**Reviewer:** Chris Williams

---

Product Name: **Executive Briefing System**
Equip. req'd: Apple II with Applesoft in ROM or RAM card. One or more disk drives. 48K or RAM memory.
Price: $199.00
Manufacturer: Professional Software Technology Inc.
180 Franklin Street
Cambridge, MA 02139
Authors: Mitchell Kapor and Todd Agulnick

**Description:** Creates and presents a slide show of Apple II hi-res screens. Provides transitions between slides (curtain-up, curtain-down, spiral, dissolve). Adjustable time per slide in automatic or manual change mode. Interrupt feature holds on any slide until ready. Quick manual advance or reverse review, if needed. Automatic shift to alternate disk drive. Automatic program restart. Multiple fonts (type-styles and sizes) with optional additional fonts. Uses VisiCalc-style cursor commands. Can use graphics made outside of program or can create graphics for use outside program.

**Pluses:** Creates separate program diskette which is unprotected and may be copied by any copy program. Has cursor draw mode for making simple borders and separations. Easy type style change in editor. Has bold-face, color, and reverse print options. Optional packed file mode allows many more slides per disk than conventional 34 sector files. Can pack save any standard graphic file. Organizes slide presentation order independent of order in which slides are created. Provides for skipping slides without deleting them from disk. Has print driver for most popular printers for paper slide copies.

**Minuses:** Inconvenient access to presentation catalog listing during editing. No provision for paper print out of presentation catalog listing. Dissolve transition mode not very neat. Smallest fonts are unreadable on any color screen (but perfectly OK on black and white monitor). No center justification function during slide creation. Master disk is copy protected.

**Documentation:** Excellent 100 plus-page manual provides hand-holding tutorial on complete program. Excellent demo program included called "The Great Conoco Auction," describing last year's take-over of Conoco by DuPont.

**Skill level:** Same skills as other business programs such as VisiCalc, etc. No computer programming skills required.

**Reviewer:** David P. Allen

---

Product Name: **600 Baud Serial Parallel Converter, p/n PI80C**
Equip. req'd: TRS-80 Color Computer
Price: $69.95
Manufacturer: The MICRO WORKS
P.O. Box 1110
Del Mar, CA 92014

**Description:** The PI80C is a small module slightly larger than a ROMPACK for the TRS-80C Color Computer. It attaches to the Color Computer *via* a 5-pin DIN plug and cable, and is powered by a wall-plug transformer. The output is a 36-pin card edge compatible with card-edge connectors which implement parallel Centronics-type printer cables. It has a fixed 600 baud data rate, requiring no operator intervention when used with the Color Computer.

**Pluses:** This unit provides a hassle-free interface between the Color Computer and printers requiring a Centronics-type parallel drive. Unlike a few simple shift register interfaces, this device provides transmission of all 7-bit and 8-bit data words, thus allowing full utility of printers requiring special escape codes and other special code sequences to activiate special printer attributes.

**Minuses:** The fixed baud rate can be changed by opening the case and adjusting the frequency of the internal oscillator for another baud rate, but this action would void the warranty. Also, no schematic is furnished.

**Documentation:** Entirely adequate for all normal usage; well done.

**Skill level required:** Ability to follow printed instructions. No special techniques are required.

**Reviewer:** Ralph Tenny

*(Continued on next page)*

## Reviews In Brief *(continued)*

| | |
|---|---|
| Product Name: | **Tricky Tutorials** |
| Equip. req'd: | Atari 400/800, 16K Cassette or 24K Disk |
| Price: | $14.95/tutorial |
| Manufacturer: | Santa Cruz Educational Software<br>5425 Jigger Dr.<br>Soquel, CA 95073 |

**Description:** These programs are designed to instruct programmers in the use of the advanced hardware buried inside each Atari. There are currently five tutorial programs. These programs can be used independently, but to get the most from them they should be studied in order. The six tutorials are: 1. display lists, 2. horizontal and vertical scrolling, 3. page flipping, 4. animation, 5. players/missiles, 6. sound and music. Each tutorial comes complete with a short manual and program media (disk or cassette). More instruction is provided within the programs as you are shown different techniques, the code that produced the display or sound with additional commentary. Suggestions are made for further experimentation and study. The power of immediate feedback with sufficient explanation is not to be underestimated.

**Pluses:** These programs combine the computer with the power inherent in color graphics and sound; this is a most effective and enjoyable learning tool. These programs reinforce information from the difficult-to-understand operating system and hardware manuals that have kept many of us up more than a few nights.

**Minuses:** The material presented is complicated and demands much from the user. The casual viewer will not learn much. Experimentation, asking the "what if" questions, and thinking, are all necessary to getting the most out of this software.

**Documentation:** The initial written documentation was somewhat skimpy. This has been updated and new material added. The entire 6-pack can be purchased in a three-ring binder.

**Skill level:** The material presented requires a comfortable knowledge of BASIC to fully explore all that's presented. Additionally, an enthusiasm for really digging into your Atari would be an advantage.

**Reviewer:** James Capparell

---

| | |
|---|---|
| Product Name: | **6502 Microcomputer, P/N 80-153** |
| Equip. req'd: | Single-board microcomputer |
| Price: | $119 assembled and tested;<br>$19.95 bare board |
| Manufacturer: | John Bell Engineering, Inc.<br>P.O. Box 338<br>Redwood City, CA 94064 |

**Description:** A single-board computer based on the 6502 microprocessor, using a 6522 VIA as a programmable I/O device, with 1024 bytes of read/write memory and one EPROM socket which can be configured for either a 2716 EPROM or a 2532 EPROM. The board is small (3¼" × 4½") and uses a 50-pin edge-card connector. No memory expansion is provided, but 25 unused pins on rear of connector would allow careful expansion within decoder limits. The memory map is decoded to allow 7168 bytes of read/write memory, 256 bytes of I/O beginning at $1C00, and 52K of EPROM. A 2716 EPROM installed in the existing EPROM socket will be addressed beginning at $F800. The result is a controller card with 16 programmable I/O lines, 1K of read/write memory and either 2K or 4K of program memory. An R/C network provides the clock.

**Pluses:** A potentially handy and versatile single-board computer at a fair price.

**Minuses:** The board has a serious, undocumented memory timing error which will cause problems in almost any application that uses read/write memory for anything except a minor amount of scratchpad. The good news is that the fix is almost zero-cost; it requires only an etch cut and a jumper wire. In view of this design error, the assembled and tested product may be over-priced, since the unwary user can lose several hours of time debugging an unreliable and erratic machine. For the informed experimenter, the required etch cut would take only minutes during assembly of the bare board.

**Documentation:** Reasonably complete and well-done; includes some start-up hints and a brief tone generator program listing.

**Skill level required:** The user must thoroughly understand all steps needed to develop a rudimentary monitor and debug a new, untested microcomputer board. He should also be skilled at board assembly, modification, and soldering if the bare-board option is chosen. The manufacturer has a monitor EPROM available, but no details of this are available wihout direct inquiry.

**Reviewer:** Ralph Tenny

---

| | |
|---|---|
| Product Name: | **Monkey Wrench** |
| Equip. req'd: | Atari 800 |
| Price: | $49.95 |
| Manufacturer: | Eastern House Software<br>3239 Linda Drive<br>Winston-Salem, N.C. 27106 |

**Description:** This ROM-based product is designed to be installed in the righthand slot of the 800. It enhances Atari BASIC with nine useful commands: (A) auto line numbering, (D) delete range of lines, (M) change margins, (T) memory test, (R) renumber, (E) cursor key without control key, ($) hex conversion, (#) decimal conversion, and a machine-language monitor providing another 15 commands. The MLM prompt is a period (.).

**Pluses:** *Monkey Wrench* puts the always empty right slot of the Atari 800 to use. It is designed to be compatible with Atari BASIC. There is room for another ROM on the cartridge and Eastern House is currently working on the new ROM enhancement. This is an excellent development tool.

**Minuses:** The cartridge uses address space $8000 to $9FFF. This will reduce your 48K machine to 32K, 8K used by the BASIC cartridge and 8K used by *Monkey Wrench*. This product uses part of page 6 for variable storage which is sure to cause some conflict. Care is needed when inserting the cartridge: it's very easy to put in backwards.

**Documentation:** The manual is short, sweet, and easy to use. A nice example of each function is provided, leaving little to your imagination.

**Skill level required:** This product can be used by any level of BASIC programmer. Serious BASIC users will find it indispensable. I recommend it.

**Reviewer:** James Capparell

# 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 microcomputer 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.

*Dr. Dobb's Journal* • A Publication of People's Computer Company, 1263 El Camino Real, P.O. Box E, Menlo Park, CA 94025

MICRO™

# PET Vet

By Loren Wright

## A New Disk Cataloging System

Programs, particularly commercial ones, should be easy to use, well-documented, and modifiable enough to meet a wide range of user needs. Terms sometimes used to describe programming of this quality include 'human-engineered' and 'user-friendly.' Many programs purport to be human-engineered or user-friendly, but few actually succeed in meeting all of the necessary criteria.

When a company calling itself "Human Engineered Software" was born last year, I viewed the event with great interest and a little suspicion. Even after I evaluated their first products — HESBAL/HESEDIT, an inexpensive assembler/editor (July 1981 issue) and HESLISTER, a program that produces 'structured' printed listings from a BASIC program on disk (August 1981 issue) — I still had some doubts about the company's ambitious name. HES's recent releases have convinced me that it is worthy of its name.

Besides human engineering, there is another theme to the company's products. All are tools, designed to aid the programmer in coding, debugging, optimizing, and generalizing his programming. One new program, called HESCOUNT, counts the number of times each line of a BASIC program is executed — a seemingly trivial task. When you think about it, this information is very valuable. Lines that never get executed can be eliminated, while lines that are executed many times show themselves as prime targets for optimization efforts.

Another product, called HESCOM, handles transfer of programs and data among different kinds of PETs and VICs. These programs will be reviewed either in this column or in "Reviews in Brief" in the near future. This month's column is reserved for HESCAT, Human Engineered Software's new disk cataloging system.

My collection of PET disks has grown to include more than 50 disks, and my ability to keep them organized has not kept pace. I keep special types of files on separate disks — WordPro, Wordcraft, HESBAL, MAE, Pascal, RPL, FORTH, and COMAL. It's the rest of the collection that is tough to keep in line. When I'm looking for a program that I haven't used lately, I can usually narrow it down to a few disks; a search of the directories involved will turn up the file in a few minutes. More and more often, though, I just can't seem to find a program, so I waste a lot of time searching. Organizing the disk collection has been on my must-do list for quite a while, but the magnitude of the task has always deferred me. When I received HESCAT for review, I was certainly interested, but I was also a bit apprehensive. Could HESCAT really solve this problem?

I gave HESCAT the ultimate test — my disk collection! The only preliminary step was to separate the 8050 disks from the rest. The system can handle 2040/4040 or 8050 disks, but not both because of the hardware differences.

HESCAT is actually a menu program that loads up the different component programs. There is a 'help' feature that will show you, on request, a brief explanation of each menu option. From the menu you select the 'catalog' option, and then, with your catalog disk in drive 0, you run through your disks one by one in drive 1.

You have to assign each disk a unique 'external ID.' These numbers are maintained separately from the internal IDs — the ones actually written on the disk. They may not be unique, and it is impossible to change them without rewriting the disk. I assigned external IDs according to the order they occurred in my pile, but it would probably be a good idea to make the internal and external IDs the same wherever possible. Cataloging each disk takes about 15 seconds; the exact time depends on how many files are on the disk. Most likely errors are trapped by the catalog program, which advises you to try again. You can recover from an error that causes a break in the program (such as inserting an 8050 disk in a 4040!).

Once you have all your disks cataloged, it is necessary to sort only the file names before you can start using the other programs. The sort is written in machine language and is very fast.

(Continued on page 61)

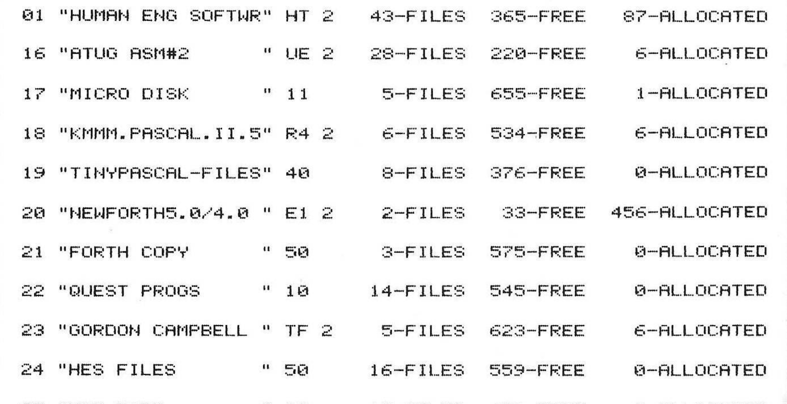


***Figure 1:*** **Portion of 'Headers' Printout**

```
01 "HUMAN ENG SOFTWR" HT 2   43-FILES  365-FREE   87-ALLOCATED

16 "ATUG ASM#2      " UE 2   28-FILES  220-FREE    6-ALLOCATED

17 "MICRO DISK      " 11      5-FILES  655-FREE    1-ALLOCATED

18 "KMMM.PASCAL.II.5" R4 2    6-FILES  534-FREE    6-ALLOCATED

19 "TINYPASCAL-FILES" 40      8-FILES  376-FREE    0-ALLOCATED

20 "NEWFORTH5.0/4.0 " E1 2    2-FILES   33-FREE  456-ALLOCATED

21 "FORTH COPY      " 50      3-FILES  575-FREE    0-ALLOCATED

22 "QUEST PROGS     " 10     14-FILES  545-FREE    0-ALLOCATED

23 "GORDON CAMPBELL " TF 2    5-FILES  623-FREE    6-ALLOCATED

24 "HES FILES       " 50     16-FILES  559-FREE    0-ALLOCATED

25 "MAE DISK        " 30     15-FILES  450-FREE    0-ALLOCATED

26 "MICRO DISK#4    " 13      3-FILES  593-FREE   -1-ALLOCATED

27 "ASSEM SYSTEM    " 01     44-FILES   74-FREE    0-ALLOCATED
```

# FUNTASTIC!

$29.95. For Apple II*

Habit forming

**SNACK ATTACK**

Betcha can't play just one game!

**DATAMOST**

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

"THIEF"

"The damn things nearly killed me."

It's you against rampaging robots. Seven levels of action and excitement to become the galaxy's master Thief.

$29.95 For Apple II*

DATAMOST,
19273 Kenya St.
Northridge, Calif. 91326.

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

DATAMOST

COUNTY FAIR
is COUNTY TERRIFIC

Pick up your gun and blast away in the most colorful, exciting shooting gallery in the world. This is the game everyone's crazy about.

$29.95
For Apple II*

*Apple II is a trademark of Apple Computer Inc.

Casino

5 great casino games. The favorites of Monte Carlo, Vegas, Atlantic City. Try your luck at Blackjack, Keno, Roulette, Poker and Baccarat. Exciting? You can bet on it!

$39.95.
For Apple II*

DATAMOST

# GAMES &

Datamost brings you the funtastic games—games that are fun, challenging . . . games where you must use skill, strategy and dexterity . . . games that span the universe from ancient pirate times to a robotic future with stops along the way for enjoyment at the county fair, a casino, a baseball game!

Scan this catalog for a list of our current games, and be sure to note the totally new ones just available: MISSING RING, BASEBALL, and SOLO POKER! Try them . . . you'll have a funtastic time.

## Missing Ring

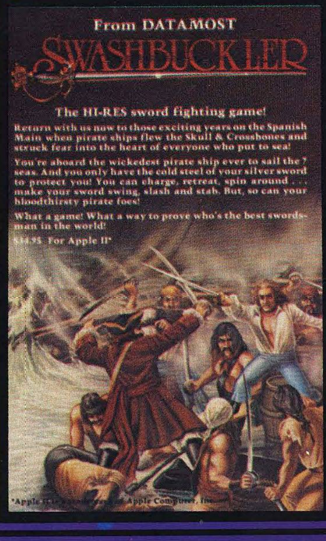
You've never enjoyed an adventure with such exciting HI-RES graphics! You're on a hunt for the mysterious Missing Ring . . . in a house full of diabolical traps. You see a FULL SCREEN layout of the level you're searching . . . plus you can call up a FULL SCREEN picture of any room you're in!

Finding the missing ring is a real challenge, a great battle of wits. And the spooky Ghost and dreaded Dragon add spice to the hunt. Here's the ultimate in HI-RES adventures . . . the one all others will be compared to! $29.95.

## World Series Baseball

Here's world series calibre Baseball in HI-RES graphics that bat a thousand. The top baseball game for two people. Takes strategy and agility and real baseball 'smarts'.

The pitch comes and the batter swings . . . your fielders race into position . . . run down grounders, have to throw to the bases. A terrifically competitive game . . . the next best thing to actually playing in the series yourself. $29.95. (paddles)



From DATAMOST

SWASHBUCKLER

The HI-RES sword fighting game!

Return with us now to those exciting years on the Spanish Main when pirate ships flew the Skull & Crossbones and struck fear into the heart of everyone who put to sea!

You're aboard the wickedest pirate ship ever to sail the 7 seas. And you only have the cold steel of your silver sword to protect you! You can charge, retreat, spin around . . . make your sword swing, slash and stab. But, so can your bloodthirsty pirate foes!

What a game! What a way to prove who's the best swordsman in the world!

$34.95 For Apple II*

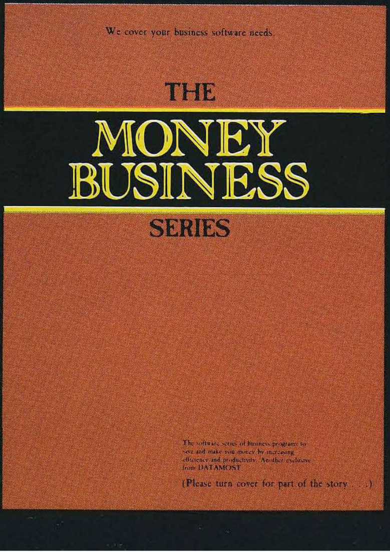
*Apple II [illegible] Apple Computer, Inc.

## Swashbuckler

Here's the best HI-RES graphic animation game you've seen. You're the sword fighting hero under attack by vicious pirates, rats, snakes, samauri and other villains. You parry, thrust, lunge . . . use your sword like a living-weapon. You start below decks and fight your way topside, if you're skillful enough to vanquish the diabolical crew! $34.95.

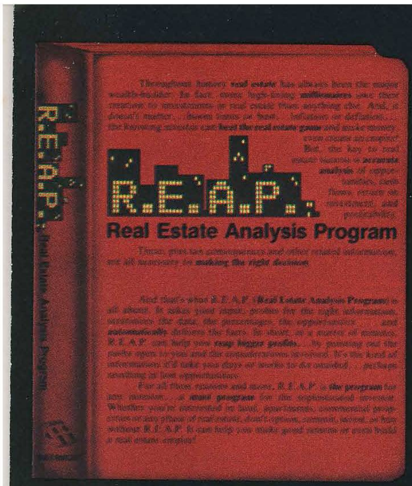
# BEYOND. . .

Datamost software covers the world of business. Our programs are not merely the brainchildren of some computer engineer. Rather, they have been written and designed by business experts — and tested in the field, in your kind of business environment. So, when you select a Datamost program, you know it'll work beautifully, easily, and live up to your expectations . . . while saving you time and money, and increasing productivity.



## Money Business series

This extensive series of programs deserves to be called the masterpiece of business software. For, unlike competitors, it combines the finest attributes you can find. Each program covers its subject thoroughly . . . so it can be applied to your business. Each program comes with documentation which is easy to read . . . easy to understand and apply, and so complete that your questions are answered almost before they're asked! (Plus we offer full professional support by phone, mail or visit.) Our MONEY BUSINESS series includes general ledger, payroll, accounts receivable, accounts payable . . . with more being proven on-site and coming your way soon. CP/M. $800/module. Complete package $2500.

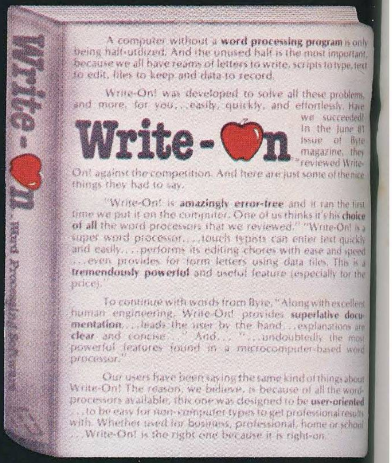


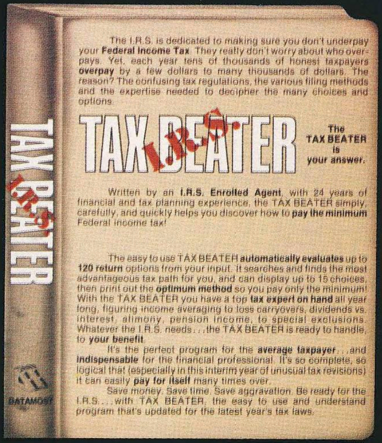
## R.E.A.P.

The Real Estate Analysis Program is a real key to the money business of real estate investment. It charts cash flows, return on investment, shows profitability and tax consequences — so you become a knowledgable investor. Loaded with the help you need to analyse every opportunity, to beat the real estate game. For Apple II. $129.95.

## Write-On!

There are word processors of all kinds, but there's only WRITE-ON! for anyone who wants the program praised by BYTE . . . the powerful word processor that lets you get started in about 5 minutes flat! Nothing easier . . . quicker to learn . . . or, so really useful. For Apple II, Apple III and for IBM. $129.95.





## Tax Beater

A necessity at tax time . . . a blessing year-round! Tax Beater takes the confusion out of taxes . . . whether the simplest or most complicated returns. It automatically evaluates up to 120 return options — so you pay the minimum. Perfect for the average taxpayer, indispensable for the financial professional. For Apple II. $129.95.

---

See your computer store for the entire line of Datamost products. If unavailable, you may order direct. For games and books, include $1.00 for shipping/handling. For business programs and hardware, add 3%. (California residents, add 6% sales tax also.)

*CP/M is a trademark of Digital Research. Apple, Apple II & III are trademarks of Apple Computer, Inc.

**DATAMOST** INC

9748 Cozycroft Ave., Chatsworth, CA 91311. (213) 709-1202

*(Continued from page 60)*

Now you can reap the benefits. Probably the most frequent use of the catalog will be to 'locate' a file. The name you give doesn't have to be specified exactly. You can use wild or 'don't care' characters. All file names that meet the criteria are listed, along with their types and disk locations.

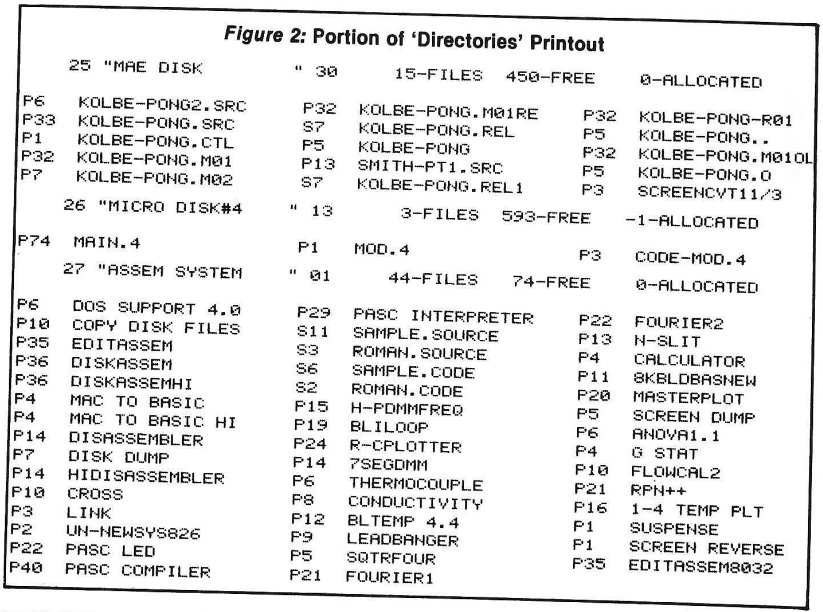
From time to time you'll want printed information on your disk files. HESCAT supports CBM, Base2, and Epson MX-80 printers. There are three types of printouts: headers (figure 1), directories (figure 2), and alphabetized names (figure 3). Headers provides summary information about each disk — name, ID numbers, number of files, number of blocks free, and number of direct access blocks allocated. Directories provides a separate directory listing for each disk, and alphabetized names provides all the names in alphabetical order, with file-type and disk number. For each print option you can specify that all disks be included in the listing, or only a specified range.

Updating the catalog is a simple matter. Just enter the 'catalog' program and catalog only the disks whose contents have changed since the last session. The names corresponding to that external ID will be replaced by those for the new contents of the disk. There is also a 'display' program, which is used to view the contents of a sequential file. You will use it occasionally to check the contents of HESCAT files, but it will also come in handy for many other applications. Finally there is a 'user function' option, which can be selected from the menu. There is no function included, but there are many possibilities.

As with the other HES products I've examined, HESCAT is well documented with both a user and a program manual. The program manual includes full listings and descriptions of the functions of all variables. The programs themselves are heavily commented. The most likely change is one to accommodate a printer not supported directly by the program.

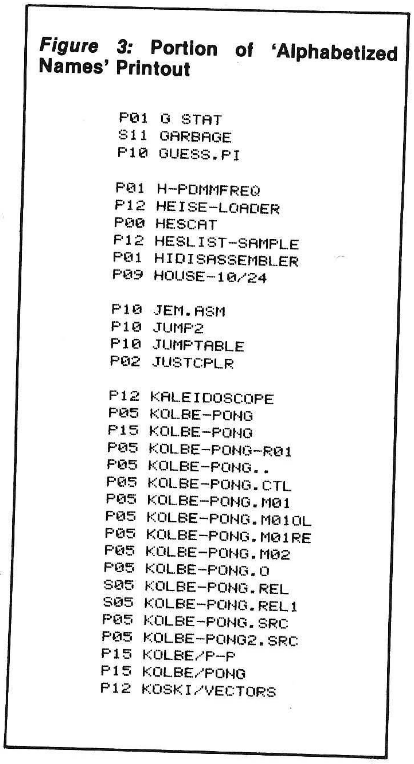
I have only one minor complaint. In the cataloging process, you are asked to supply the external ID for a disk before its directory is read. Giving the wrong ID number may accidentally wipe out the file information for a disk already cataloged. The program should check the name and IDs against what already exists in the catalog and give you a chance to confirm your decision to recatalog a disk. This would also help in matching external IDs to the internal ones.

**Figure 2: Portion of 'Directories' Printout**



```
    25 "MAE DISK           " 30      15-FILES  450-FREE     0-ALLOCATED

P6   KOLBE-PONG2.SRC      P32  KOLBE-PONG.M01RE    P32  KOLBE-PONG-R01
P33  KOLBE-PONG.SRC       S7   KOLBE-PONG.REL      P5   KOLBE-PONG..
P1   KOLBE-PONG.CTL       P5   KOLBE-PONG          P32  KOLBE-PONG.M01OL
P32  KOLBE-PONG.M01       P13  SMITH-PT1.SRC       P5   KOLBE-PONG.O
P7   KOLBE-PONG.M02       S7   KOLBE-PONG.REL1     P3   SCREENCVT11/3
    26 "MICRO DISK#4       " 13       3-FILES  593-FREE   -1-ALLOCATED

P74  MAIN.4               P1   MOD.4               P3   CODE-MOD.4
    27 "ASSEM SYSTEM       " 01      44-FILES   74-FREE     0-ALLOCATED

P6   DOS SUPPORT 4.0      P29  PASC INTERPRETER    P22  FOURIER2
P10  COPY DISK FILES      S11  SAMPLE.SOURCE       P13  N-SLIT
P35  EDITASSEM            S3   ROMAN.SOURCE        P4   CALCULATOR
P36  DISKASSEM            S6   SAMPLE.CODE         P11  8KBLDBASNEW
P36  DISKASSEMHI          S2   ROMAN.CODE          P20  MASTERPLOT
P4   MAC TO BASIC         P15  H-PDMMFREQ          P5   SCREEN DUMP
P4   MAC TO BASIC HI      P19  BLILOOP             P6   ANOVA1.1
P14  DISASSEMBLER         P24  R-CPLOTTER          P4   G STAT
P7   DISK DUMP            P14  7SEGDMM             P10  FLOWCAL2
P14  HIDISASSEMBLER       P6   THERMOCOUPLE        P21  RPN++
P10  CROSS                P8   CONDUCTIVITY        P16  1-4 TEMP PLT
P3   LINK                 P12  BLTEMP 4.4          P1   SUSPENSE
P2   UN-NEWSYS826         P9   LEADBANGER          P1   SCREEN REVERSE
P22  PASC LED             P5   SQTRFOUR            P35  EDITASSEM8032
P40  PASC COMPILER        P21  FOURIER1
```

**Figure 3: Portion of 'Alphabetized Names' Printout**



```
P01 G STAT
S11 GARBAGE
P10 GUESS.PI

P01 H-PDMMFREQ
P12 HEISE-LOADER
P00 HESCAT
P12 HESLIST-SAMPLE
P01 HIDISASSEMBLER
P09 HOUSE-10/24

P10 JEM.ASM
P10 JUMP2
P10 JUMPTABLE
P02 JUSTCPLR

P12 KALEIDOSCOPE
P05 KOLBE-PONG
P15 KOLBE-PONG
P05 KOLBE-PONG-R01
P05 KOLBE-PONG..
P05 KOLBE-PONG.CTL
P05 KOLBE-PONG.M01
P05 KOLBE-PONG.M01OL
P05 KOLBE-PONG.M01RE
P05 KOLBE-PONG.M02
P05 KOLBE-PONG.O
S05 KOLBE-PONG.REL
S05 KOLBE-PONG.REL1
P05 KOLBE-PONG.SRC
P05 KOLBE-PONG2.SRC
P15 KOLBE/P-P
P15 KOLBE/PONG
P12 KOSKI/VECTORS
```

My overall impression, though, is very favorable. HESCAT can handle many more disks and file names than any one person is likely to have (the exact number depends on the computer, disk drive type, and program modifications). If you have more than a few disks to keep in order, I recommend HESCAT to you. I'm now looking for a program to do something about my desk.

HESCAT by Jerry Bailey is available from Human Engineered Software (3748 Inglewood Blvd., Rm. 11, Los Angeles CA 90066) for $23.95.

MICRO™

## 80 × 25

**PET/CBM™**

2000/3000/4000 Series

not using a CRT, or display controller chip

$275.00*

Select either **80 × 25** or **40 × 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

## 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 lazers 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

## FORTH-79

**Ver. 2 For your APPLE II/II+**

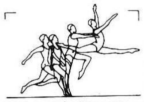
**The complete professional software system, that meets ALL provisions of the FORTH—79 Standard (adopted Oct. 1980). Compare the many advanced features of FORTH—79 with the FORTH you are now using, or plan to buy!**

| FEATURES | OURS | OTHERS |
|---|---|---|
| 79-Standard system gives source portability. | YES | |
| Professionally written tutorial & user manual | 200 PG. | |
| Screen editor with user-definable controls. | YES | |
| Macro-assembler with local labels. | YES | |
| Virtual memory. | YES | |
| Both 13 & 16-sector format. | YES | |
| Multiple disk drives. | YES | |
| Double-number Standard & String extensions. | YES | |
| Upper/lower case keyboard input. | YES | |
| LO-Res graphics. | YES | |
| 80 column display capability | YES | |
| Z-80 CP/M Ver. 2.x & Northstar also available | YES | |
| Affordable! | $99.95 | |
| Low cost enhancement option: | | |
| Hi-Res turtle-graphics. | YES | |
| Floating-point mathematics. Powerful package with own manual, 50 functions in all, AM9511 compatible. | YES | |

| | |
|---|---|
| FORTH—79 V.2 (requires 48K & 1 disk drive) | $ 99.95 |
| ENHANCEMENT PACKAGE FOR V.2 Floating point & Hi-Res turtle-graphics | $ 49.95 |
| COMBINATION PACKAGE | $139.95 |

(CA res. add 6% tax: COD accepted)

**MicroMotion**
12077 Wilshire Blvd. # 506
L.A., CA 90025 (213) 821-4340
Specify APPLE, CP/M or Northstar
Dealer inquiries invited.



## PRINTERS FOR THE 80's

| | | |
|---|---|---|
| **CENTRONICS** | 739-1/Parallel | 525. |
| | 739-3/Serial | 625. |
| | 2 Color Adapter | 69. |
| **OKIDATA** | Microline 80 | 349. |
| | Microline 82-A | 479. |
| | Microline 83-A | 749. |
| | Microline 84 | 1,055. |
| | Okigraph (82A or 83A) | 69. |
| **NEC** | PC8023-A | 509. |
| **EPSON** | MX 80 | 449. |
| | MX 80/FT | 549. |
| | MX 100 | 719. |
| | Graftrax | 69. |
| **C. Itoh** | NEW! | |
| | F-10 Daisy Wheel 40/cps | 1525. |
| | Comet I/P | 295. |
| | Starwriter 45cps serial | 1,599. |

**Special! Regularly sells for 2,595.**

master charge VISA

**Cables and interface cards available for the Atari, Apple, TRS-80 Osborne and CBM/Pet. FREE SHIPPING and HANDLING.**

**BC BUSINESS COMPUTERS** of Peterborough
*766 Middle Hancock Road*
*Peterborough, NH 03458* **(603) 525-6621**

# BASIC, FORTH, and RPL

### by Timothy Stryker

**BASIC and FORTH, two widely accepted high-level languages for microcomputers, are compared to RPL, a relative newcomer to the field. The languages are compared with respect to time-efficiency, space-efficiency, transportability, and ease-of-use considerations.**

Although RPL is currently available only for PET and CBM, this article is of general interest.

*Editor's Note: Timothy Stryker is the developer of RPL and his company, "Samurai Software" markets RPL for PET and CBM computers.*

BASIC, Beginners' All-Purpose Symbolic Instruction Code, is an excellent language in many respects. It is easy to learn and easy to use. It is very tolerant of user error, which makes the debugging of programs in BASIC a relatively simple matter. Although many different versions of BASIC exist, enough of its features have become standardized to make it reasonably transportable. A BASIC program written on a PET or CBM will generally run on other machines with minor modifications, and *vice versa.*

The main problem with BASIC is its speed. If you have ever tried to write a real-time game or process control program in BASIC, you have no doubt found that it bogs down very easily. PET/CBM BASIC is one of the fastest floating-point BASICs in existence, but there are still many applications for which this BASIC is too slow to be of value. BASIC also consumes prodigious amounts of memory, both for storing user programs and for storing the data to be processed during execution. This can lead to problems with the OUT OF MEMORY error in the course of writing large applications programs.

BASIC's lack of speed has caused many programmers to become interested in FORTH as a widespread language for small computers. Because FORTH usually manipulates numbers in integer form, it gains a significant speed advantage over BASIC (microprocessors can manipulate integers much more quickly than they can floating-point numbers). In addition, FORTH is a "compiled" language, which means faster performance in looking up variables in tables, finding destinations of control-flow branches, and the like. Primarily through the efforts of the "FORTH Interest Group" ("fig") in San Francisco, FORTH has become sufficiently well-known to make possible a reasonable degree of transportability between machines. Unfortunately, FORTH's "extensibility" (its ability to allow the user to add new constructs of all kinds to the language) has led to numerous substantially different flavors of FORTH on the market. Nonetheless, most "fig-FORTH" versions adhere fairly closely to the Interest Group's standards.

The main reason that FORTH has not caught on more strongly is that the language is considerably more difficult to use than BASIC. FORTH operations are ordered according to Reverse Polish Notation, which many people find objectionable. At the same time, the fig-FORTH text editor and its associated disk I/O standards are both unique and cumbersome, which makes FORTH source file management difficult and error-prone. The lack of worthwhile FORTH debugging tools has not helped the situation either: once debugged, FORTH programs tend to be fairly solid, but getting to that point can take a major effort.

In the midst of all this ferment, a new language called RPL has appeared on the scene. RPL, which stands for Reverse Polish Language, is to some extent a combination of BASIC and FORTH. RPL is a compiled language in the same sense that FORTH is. RPL object code is not itself machine code, but it can be interpreted by a machine-language "run-time executive" that is part of the package. RPL also uses both a parameter stack and a return stack, just like FORTH. However, RPL resembles BASIC in many respects: its implementation on the PET/CBM uses Commodore's BASIC screen editor and all of the normal BASIC source file manipulation commands, like SAVE and DSAVE, LOAD and DLOAD, LIST, DIRECTORY, and so on. This means that RPL programs can be of arbitrary length, without your having to break them up into 1024-byte sections the way you must with FORTH programs. Also, RPL program listings read from top to bottom, just like BASIC listings (which is to say, unlike FORTH listings).

Interestingly enough, RPL is substantially more efficient than FORTH in both space and time in spite of the fact that it is easier to use. Since dedicated FORTHers will no doubt find this hard to believe, I have assembled a few small benchmarks on the CBM 8032 in order to compare BASIC, fig-FORTH, and RPL in terms of their processing speeds and their memory usage.

## The Block-Move Benchmark

Listings 1a, 1b, and 1c show BASIC, fig-FORTH, and RPL implementations of a simple block-move benchmark. The three versions have been kept in as close a correspondence to one another as possible — thus, line 150 of the BASIC version corresponds directly to line 150 of the RPL version and to line 6 of the FORTH version, and so on. Since the routines must appear in bottom-up order in the FORTH version, the symmetry is somewhat broken here, but the BASIC and RPL versions are line-for-line equivalents of one another.

Each benchmark begins by zeroing the 8032's internal timer so that timing measurements can be made. In BASIC, this is accomplished by setting the variable TI$ to a string of six zeroes, whereas in both FORTH and RPL, this is done by storing a zero into the word at memory location 142 (the sequence {0 142 !} accomplishes this in both languages).

Then, a 100-pass loop is set up so that the routines to be tested will each be run 100 times. Here we notice the first difference between FORTH and RPL: the RPL version does a {100 1 FOR} to accomplish this, whereas the FORTH version does a {101 1 DO}. RPL's FOR is the equivalent of FORTH's DO, except that in FORTH the upper bound of an iterative loop like this must always be specified as 1 greater than the actual upper bound desired. RPL is more like BASIC in this regard, as you can see.

**Listing 1A**

```
100 REM **************************************************
110 REM *          BASIC BLOCK-MOVE BENCHMARK             *
120 REM * ROUTINE AT LINE 1000 MOVES C BYTES FROM F TO T *
130 REM **************************************************
140 REM
150 TI$="000000" : FOR I = 1 TO 100
160 C=150 : T=826 : F=634 : GOSUB 1000
170 NEXT I : PRINT TI;"JIFFIES" : END
1000 M=F-T : FOR J = T TO T+C-1 : POKE J,PEEK(J+M) : NEXT : RETURN
```

**Listing 1B**

```
0 ( *********************************************************** )
1 ( *               FORTH BLOCK-MOVE BENCHMARK                * )
2 ( * BLKM EXPECTS FROM-ADDR ON TOS, THEN TO-ADDR, THEN COUNT * )
3 ( *********************************************************** )
4
5 : BLKM OVER - SWAP ROT OVER + SWAP DO DUP I + C@ I C! LOOP DROP ;
6 : TEST 0 142 ! 101 1 DO
7   150 826 634 BLKM
8   LOOP 142 @ INT . ." JIFFIES" ;
```

**Listing 1C**

```
100 ***********************************************************************
110 *                   RPL BLOCK-MOVE BENCHMARK                            *
120 * THE BLKM ROUTINE EXPECTS FROM-ADDR ON TOS, THEN TO-ADDR, THEN COUNT *
130 ***********************************************************************
140 REM
150 0 142 ! 100 1 FOR
160 150 826 634 BLKM &
170 NEXT 142 @ INT STR$ PRINT " JIFFIES" PRINT STOP
1000 BLKM: ; - % 3 $ ; + 1 - % FOR # FN + PEEK FN POKE NEXT . RETURN
```

The body of the loop in each case consists of setting up parameters to be passed to the routine under test, followed by a call to the routine itself. In the BASIC version, setting up of the parameters is accomplished by assigning values to the variables C, T, and F, which in this case specify that the 150 bytes starting at 634 are to be moved up to start at 826. The FORTH and RPL versions of the benchmark, however, expect these arguments to be passed to them on the parameter stack.

A crucial difference between FORTH and RPL is apparent here in the way in which the call itself is done: note that, in FORTH, the simple statement {BLKM} is sufficient to invoke the routine of that name. In RPL, saying simply {BLKM} merely causes the *address* of the BLKM routine to be *pushed onto the stack*: it is the {&} operator (pronounced "call") that actually causes control to be transferred to the routine whose address appears on top of the stack. There are numerous reasons why calls are set up this way in RPL, and we do not have the space here to go into them in any detail. Suffice it to say that the reasons center around space efficiency, speed, and ease of use, all three of which are optimized through the use of this construct. As evidence of this I can only cite the results of the benchmarks given here. These results are almost wholly due to precisely this difference between FORTH and RPL: in FORTH, you call a routine by simply stating its name, whereas in RPL you call a routine by stating its name and then invoking the {&} ("call") operator.

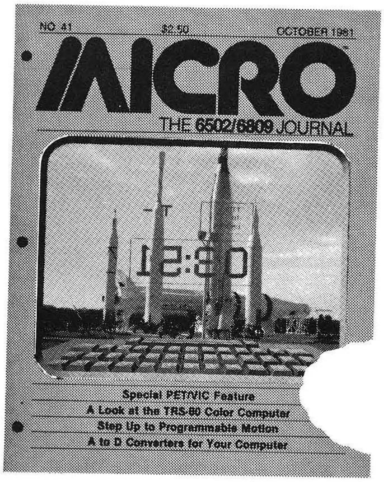
Table 1 shows a few of the RPL operators and their FORTH equivalents. Using this table, you can see that the FORTH and RPL versions of the BLKM routine itself are virtually word-for-word equivalents of one another, the only difference being that, as before, the upper bound of the DO-LOOP in FORTH is 1 greater than the upper bound of the FOR-NEXT in RPL and BASIC. Note that FORTH's {ROT} operator is equivalent to the sequence {3 $} in RPL: RPL's {$} operator takes the item on top of the stack and uses it to determine how deep into the stack the rotation process will go.

*(Continued on page 66)*

***Table 1:* A few of the RPL operators and their FORTH equivalents (TOS means Top Of Stack; NOS means Next On Stack).**

| RPL | FORTH | Effects |
|---|---|---|
| + | + | Add TOS to NOS, pop TOS |
| – | – | Subtract TOS from NOS, pop TOS |
| \ | MOD | Take NOS *modulo* TOS, pop TOS |
| IF | IF | Begin conditional based on TOS |
| THEN | ELSE | End THEN-part, begin ELSE-part |
| END | THEN | End conditional |
| @ | @ | Replace TOS with the word it points to |
| ! | ! | Store NOS into word pointed to by TOS, pop both |
| PEEK | C@ | Replace TOS with the byte it points to |
| POKE | C! | Store NOS into byte pointed to by TOS, pop both |
| # | DUP | Push a new copy of TOS |
| ; | OVER | Push a new copy of NOS |
| . | DROP | Drop (or pop) TOS |
| % | SWAP | Swap TOS and NOS |
| $ |  | Rotate out TOS'th stack entry onto TOS |

# 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.

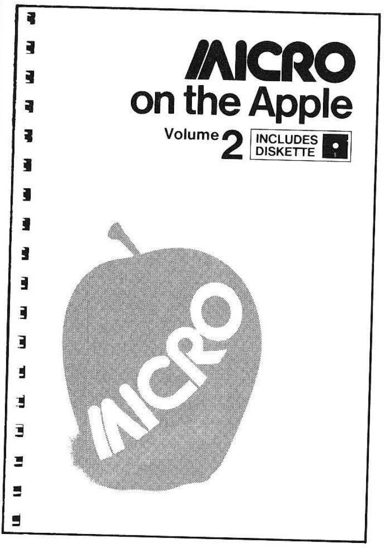
**SUBSCRIPTION RATES** (U.S. dollars) Yearly subscription (ISSN 027-9002) saves 20% off the single-issue price.

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!

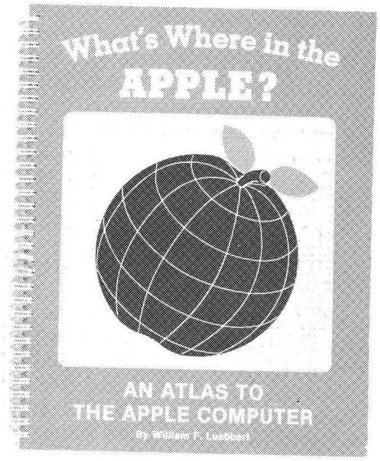
More than 40 new programs on diskette to help you get more from your Apple:

- 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.

## 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!

★ 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.

| QTY | ITEM | PRICE EACH | COST |
|---|---|---|---|
| ______ | **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<br>MICRO Journal excluded. Allow 4-6 weeks for delivery. | | ______ |
| | | **TOTAL** | ______ |

☐ Check enclosed ☐ Charge my credit card below

Name ______________________________

Company ______________________________

Street ______________________________

City ______________________ State ______ ZIP ______

Signature ______________________ ☐ VISA ☐ MasterCard

Card Number ______________________ Expiration Date ______

Once the block-move routine has been executed 100 times, each program then prints out the number of "jiffies" (1/60ths of a second) that have elapsed since it started up. In BASIC, this consists of simply printing out the variable TI. In FORTH and RPL, the contents of location 142 are fetched onto the stack, "byte-interchanged", and printed out. The "byte-interchange" operation {INT} is necessary only because PET/CBM BASIC stores the timer in high-order-byte-first order, whereas both FORTH and RPL expect fetched quantities to appear in the usual low-order-byte-first order in memory.

In the process of printing out the jiffy count and the word JIFFIES, we see another fundamental difference between FORTH and RPL: RPL treats character strings as an elementary data type, whereas standard FORTH does not. The FORTH {.} operator both converts the top stack entry to ASCII and prints it out, and FORTH's {."} operator unconditionally prints out the character string following it — at no time does fig-FORTH leave a character string sitting on the stack in such a way that the user can get at it. In RPL, character strings representing numbers are frequently placed onto the stack in such a way that the user can then manipulate them to any desired purpose, because RPL's {STR$} operator merely converts the top stack entry to a character string, and the PRINT operator is necessary actually to print the number out. Similarly, when RPL comes across a literal character string enclosed in quotes, it simply pushes that string onto the stack: once the string is on the stack, it may be manipulated further, or, as in this case, immediately printed out using PRINT.

***Table 2:* Results from the Block-Move Benchmark**

| | Program Bytes | Data Bytes | Jiffies | Figure of Merit |
|---|---|---|---|---|
| BASIC | 115 | 42 | 6044 | 23.23 |
| FORTH | 92 | 0 | 591 | 1.82 |
| RPL | 57 | 0 | 525 | 1.00 |

Table 2 displays the results from this first benchmark. In BASIC's case, the "Program Bytes" column does not count REMarks or spaces, but only the actual amount of memory taken up by the code itself. Many programs are available that compress BASIC code by removing all REMark lines and extraneous blanks, and the figure given here applies to the code size following a compression of this sort. In FORTH and RPL, the size of the object code is of course independent of the number of comments and spaces appearing in the source.

The FORTH and RPL "Program Bytes" entries pertain only to the object code actually generated by the portions of the programs shown in listings 1b and 1c. It should be kept in mind that both of these languages actually incur about one additional K in minimum run-time memory overhead — in FORTH's case, for the so-called "inner interpreter;" in RPL's, for the so-called

**Listing 2A**

```
100 REM *********************************************************
110 REM *               BASIC SHUFFLER BENCHMARK                *
120 REM * THE ROUTINE AT LINE 1000 RETURNS A SHUFFLED DECK IN A *
130 REM *********************************************************
140 REM
145 DIM A(51)
150 TI$="000000" : FOR I = 1 TO 100
160 GOSUB 1000
170 NEXT I : PRINT TI;"JIFFIES" : END
1000 FOR J = 0 TO 51 : A(J)=J : NEXT
1010 FOR J = 1 TO 200
1020   R1=INT(52*RND(1)) : R2=INT(52*RND(1))
1030   T=A(R1) : A(R1)=A(R2) : A(R2)=T
1040 NEXT : RETURN
```

**Listing 2B**

```
 0 ( ******************************************************* )
 1 ( *              FORTH SHUFFLER BENCHMARK               * )
 2 ( * THE SHUFFLE ROUTINE RETURNS A SHUFFLED DECK IN DECK * )
 3 ( ******************************************************* )
 4
 5 0 VARIABLE DECK 52 ALLOT
 6 : SHUFFLE 52 0 DO I DECK I + C! LOOP
 7   201 1 DO
 8     52 RND 52 RND
 9     DECK + DUP C@ SWAP ROT DECK + SWAP OVER C@ SWAP C! C!
10   LOOP ;
11 : TEST 0 142 ! 101 1 DO
12   SHUFFLE
13   LOOP 142 @ INT . ." JIFFIES" ;
```

**Listing 2C**

```
100 *******************************************************
110 *                RPL SHUFFLER BENCHMARK               *
120 * THE SHUFFLE ROUTINE RETURNS A SHUFFLED DECK IN DECK *
130 *******************************************************
140 REM
150 0 142 ! 100 1 FOR
160 SHUFFLE &
170 NEXT 142 @ INT STR$ PRINT " JIFFIES" PRINT STOP
1000 SHUFFLE: 51 0 FOR FN DECK FN + POKE NEXT
1010 200 1 FOR
1020   RND 52 \ RND 52 \
1030   DECK + # PEEK % 3 $ DECK + % ; PEEK % POKE POKE
1040 NEXT RETURN
1050 DECK:(<52>)
```

"run-time executive." I would have included this memory overhead in a separate column if not for the fact that standard fig-FORTH makes it very difficult to separate out the inner interpreter from the rest of the FORTH operating system. Many FORTH packages include some sort of special "target" compiler program which facilitates this process (a capability which is an inherent part of the RPL package, by the way), but mine is not one of them. In any event, because this fixed overhead for FORTH and RPL has little to do with the relative space-efficiencies of the two languages, it was ignored for the purpose of doing these benchmarks.

BASIC users may be perplexed by the figure of zero "Data Bytes" given in this benchmark for both FORTH and RPL. Naturally, both FORTH and RPL do manipulate data, and this data does need to be stored someplace during execution. In this case, though, all of the storage needed for both of these versions exists on their stacks. This makes the data storage essentially "free of charge." If this bothers you at all, consider that, unbeknownst to you, BASIC also has a stack that it uses extensively for various purposes, and the stack space that BASIC uses in these benchmarks has not been counted against it in the "Data Bytes" figures either.

The column entitled "Figure of Merit" in table 2 is based around the notion that the overall efficiency of a language is a function of both its time-efficiency (speed) and its space-efficiency. One fairly common way to combine these two measures of efficiency is to multiply each version's program size by the amount of time it took to execute: the lower this number is, then, the more efficiently the language handled the benchmark. In table 2 these Figures of Merit have been normalized in order to show their ratios to RPL.

## The Shuffler Benchmark

Listing 2 shows BASIC, FORTH, and RPL versions of a benchmark designed to test the languages' efficiencies at a typical game-related task. Each version times 100 passes through a "card shuffling" routine, in which array entries of 0 through 51 are used to represent the 52 cards of a normal playing deck.

Line 145 of the BASIC version, line 5 of the FORTH version, and line 1050 of the RPL version all accomplish essentially the same thing — each allocates space for a 52-element array that will be used to store the simulated deck of cards. This brings out another fundamental difference between RPL and FORTH: in RPL all handling of user-defined symbols is carried out in just one way, whereas FORTH handles user-defined symbols differently depending on whether they are subroutine names, variable names, constants, etc. RPL's symbol-handling concept is similar in many respects to that used in assembly language, in that a given user-defined symbol in RPL provides merely an alternate method of specifying what would otherwise appear in the code as a literal numeric value or address. As anyone who has programmed in assembly language knows, this is a simple, but surprisingly powerful way of handling symbols: in particular, the ability to treat addresses of routines as ordinary data can be very useful in certain applications. None of the benchmarks shown here makes use of this capability, because neither FORTH nor BASIC has a corresponding capacity along these lines. Note, however, that RPL makes no real distinction between the symbol DECK defined in line 1050 of listing 2c and the symbol SHUFFLE defined in line 1000. Each of them simply takes on a value equal to the address of the subsequent byte of object code, and, when referred to, simply causes that address to be pushed onto the parameter stack at run-time.

Each of the three versions of the shuffling routine operates in basically the same way: the 52-element array is initialized to the numbers 0 through 51, in sequence, and the order of these elements is then randomized by means of swapping random pairs a total of 200 times. The results of this benchmark are shown in table 3. The astonishingly long time taken by FORTH in this benchmark seems to be largely due to its handling of the MOD function, the invocation of which is internal to my version of the FORTH RND routine. The MOD operator under fig-FORTH on the CBM 8032 takes over 4.5 milliseconds to execute, whereas its RPL equivalent, { \ }, takes less than 1.2 milliseconds, worst case. Since the two operators yield identical results, it should be recognized that FORTH's poor performance in this benchmark is not primarily a function of anything inherent in the language itself, but is largely due to the speed of the modulo algorithm chosen by the FORTH Interest Group.

### The Falling-Tone Benchmark

Listing 3 contains BASIC, FORTH, and RPL versions of a benchmark designed to test the languages in terms of general logical and arithmetic manipulations, including comparisons, conditional branching, and memory accesses. Each program times itself doing 100 calls to a routine that generates a falling whistle on the 8032's internal speaker. The method used to generate the falling whistle is based on the "VDC" algorithm (see the 10/81 issue of *BYTE*, p. 391). Each octave drop in pitch takes the same amount of time, regardless of whether the octave is toward the top of the range or toward the bottom (the same cannot be said of the obvious "FOR I=1 TO 255:POKE 59466,I:NEXT" in BASIC).

This benchmark brings out yet another major difference between RPL and FORTH. First of all, note that in the BASIC version of the benchmark, a "conditional-within-a-conditional" (in line 1030) takes control out of the loop in lines 1020-1050 if the condition is met. There is every reason to suppose that this is a perfectly "structured" thing to do: only if the first condition (in line 1020) is not met will we determine whether or not it is time to exit the loop. FORTH, however, does not permit this kind of construct. One may set up a BEGIN...WHILE...REPEAT loop in FORTH, but the WHILE operator may not appear within the bounds defined by an IF...THEN pair within the loop. This is restrictive, to say the least, and in an application like this one it unavoidably leads to slower code. The best I could do to get around this in FORTH was to place the WHILE test *outside* the main conditional clause, which meant that it got executed on *every* pass through the loop, regardless of whether or not it needed to be. RPL, being much more like BASIC in this regard, is able to get around this problem through the use of a GOTO (horrors!). This naturally opens Pandora's Box as far as hard-core structured-programming people are concerned. Suffice it to say that I feel that the real value of structured programming lies in its concern with modularization and clean, well-thought-out software design, not in terms of myopic, over-applied dogmas such as "No GOTOs!" and "No Multiple Entry Points!", etc.

Table 4 shows the results from this third benchmark. It should be clear from these various figures that FORTH is more efficient than BASIC at handling the kinds of tasks shown here, and that RPL is even more efficient than FORTH at these types of tasks, by perhaps a factor of two overall.

### Other Tradeoffs

Ease of use is a very important criterion in determining the real utility of a language. BASIC is the acknowledged leader in this area, at least as far as "quick-and-dirty" programming is concerned. One of BASIC's best qualities is the interactive nature of its pro-

**Beagle Bros**
**MICRO SOFTWARE**

**DON'T BLOW YOUR BUCKS ON Locked-up Software.**

**Beagle Bros Apple Utilities are BACKUPABLE, LISTABLE, CUSTOMIZABLE and fully compatible with normal DOS.**

**Alpha Plot**
**Hi-Res Graphics/Text Utility**
**by Bert Kersey & Jack Cassidy**

Here are just a few of Alpha Plot's easy-to-use features. Compare price & functions with others on the market—

**HI-RES DRAWING:** Create pictures and charts with text, on both pages of memory; all images are **appendable to your programs.** Optional Xdraw cursor (see lines before you draw). **Color mixes** and Reverse (opposite of background). Circles, Boxes, and Ellipses, filled or outlined. **Store hi-res in 1/3 disk space.** Redraw any portion of your picture on either hi-res screen. Also **superimpose images** and convert hi-res to lo-res and back for fascinating colorful abstracts!

**HI-RES TEXT:** Beautiful **upper & lower case** with descenders (no hardware required). Color or reverse characters positionable anywhere (no vtab/htab restrictions). Attractive **proportional spacing** with adjustable height, leading (line spacing), and kerning (letter spacing). Multi-directional typing too for charts!

$39.50 +1.50 SHIPPING
☐ **Alpha Plot** on Disk (48K min.)
☐ Beagle Bros **Apple Tip Book #4**
☐ **Peeks, Pokes & Pointers** Chart

**All Disks Include:** Our famous **PEEKS & POKES** Wall Chart, an indispensable tool, AND a different Beagle Bros **APPLE TIP BOOK**—each one a gold mine of tips and juicy Apple info.

**Utility City**
**21 Useful Utilities on One Disk**
**by Bert Kersey**

21 programming aids that you can list, customize and back-up: **List Formatter** makes properly spaced and indented listings with printer page breaks; each statement on a new line with if-then's and loops called out; a great de-bugger! **Catalog in multiple-columns** and any page-width to printer or screen. Automatically post run-number and last-used DATE in your programs. Write **invisible functioning commands** into your listings. Access program lines in memory for garbage repair and "illegal" alteration. Quickly alphabetize and store info on disk. Run any program while another stays in memory. Renumber to 65535. Save inverse, trick and **invisible file names.** Convert decimal to hex & binary, or INT to FP. Append programs. Dump text to printer... and MORE!

$29.50 +1.50 SHIPPING
☐ **21 Programs** on disk (48K min.)
☐ Beagle Bros **Apple Tip Book #3**
☐ **Peeks, Pokes & Pointers** Chart

**NEW SHAPE TABLE CREATOR Apple Mechanic**
**MULTIPLE-UTILITY DISK** +TIP BOOK #5 $29.50

**DOS BOSS**
**DISK COMMAND EDITOR**
**by Bert Kersey & Jack Cassidy**

A classic utility you will ENJOY! Rename commands: For example, "Catalog" can be "CAT" or anything you want! **Save-protect** your programs: An unauthorized copy attempt can produce a (beep!) "NOT COPYABLE" message. **LIST-protection** too and one-key program sélection from catalog. Catalog customizer: **Change Disk Volume message** to your title; Omit or alter file codes. Rewrite error messages: "Syntax Error" can be "Oops!" or "Disk Full" can be "Burp!" or anything! Fascinating educational documentation included; Hours of good reading!

**Any or all changes may be appended to your programs, so that anyone using your disks (booted or not) on any Apple will be formatting DOS the way YOU designed it.**

$24.00 +1.50 SHIPPING
☐ **Dos Boss** on Disk (32K or 48K)
☐ Beagle Bros **Apple Tip Book #2**
☐ **Peeks, Pokes & Pointers** Chart

**Visit Your Apple Dealer.**
Most Apple Dealers carry our software. If yours doesn't, he can get it for you within just a few days through **BEAGLE BROS** or **SOFTSEL.**

**OR Order by Phone:**
**24-Hour TOLLFREE Order Desk**
Visa, MasterCard or COD* Orders, call:
Nationwide: **800-854-2003** ext.827
California: **800-522-1500** ext.827
Alaska/Hawaii: **800-854-2622** ext.827
(ORDERS ONLY, PLEASE) *COD add $3

**Or Mail us a check** (or Visa/MC no. & exp.)
☐ **Alpha Plot $39.50**
☐ **Dos Boss $24.00**
☐ **Utility City $29.50**
☐ **Apple Mechanic $29.50**

Add $1.50 shipping (any size order) and 6% tax if California. MAIL TO:
**BEAGLE BROS, Dept.M**
**4315 Sierra Vista**
**San Diego, Ca 92103**
Please add $4 shipping outside US/Canada

VISA
master charge

gram debugging facilities: the BASIC programmer has the option of halting his program at any point so that he can examine variables of interest, etc., and execution can then be continued from where it left off. FORTH has similar debugging tools, but their utility is blunted somewhat due to the opaque nature of the FORTH stacks, not to mention the clumsiness of FORTH's editing, source file handling, and so on. RPL solves this problem by providing a "symbolic debugger" as a separate utility program that allows any RPL program to be debugged *via* single-stepping, breakpointing, and the like. And the entire contents (up to 18 entries deep) of both the parameter stack and the return stack are available for viewing at any time.

"Extensibility," or the ability of the language to be augmented by the user, is one area in which FORTH shines brightly. Two distinct capabilities of FORTH here are sometimes lumped together in reviews of the subject. One is that normal FORTH routines, once defined, become as though part of the language itself. This feature is really no different in principle from the conventional method of subroutine calling used in other languages. The other capability is much more intriguing: the {<BUILDS} and {DOES>} operators give the FORTH user the ability to effectively modify the FORTH compiler on the fly, so that whole new language constructs can be created. No other language I know of (with the possible exception of Ada) has this feature. How desirable this is, to many people, remains to be seen: the benefits accruing to it must certainly be balanced against the code obscurity and tendency toward destandardization resulting from it. For example, the same capability is implemented in some FORTH versions as BEGIN...IF...WHILE, in some as WHILE...PERFORM...PEND, and in some as BEGIN...WHILE...REPEAT.

It is in the area of transportability that BASIC and FORTH find their strongest advantage over RPL. RPL is presently available only for the Commodore PET and CBM series of machines, whereas BASIC and FORTH have both become widespread. Every new language, though, goes through a period of limited transportability in its early stages. If the language really does present worthwhile advantages over existing languages, it will eventually be adapted to run on systems other than the one on which it was developed. In fact, Samurai Software is now actively seeking individuals who would be interested in adapting RPL to the Apple, the TRS-80, CP/M, and so on. Would *you*, by any chance, be interested?

**Listing 3A**

```
100 REM ********************************************************************
110 REM *                 BASIC FALLING-TONE BENCHMARK                     *
120 REM * ROUTINE AT LINE 1000 GENERATES CB2 TONE WITH EXPONENTIAL FALLOFF *
130 REM ********************************************************************
140 REM
150 TI$="000000" : FOR I = 1 TO 100
160 GOSUB 1000
170 NEXT I : PRINT TI;"JIFFIES" : END
1000 POKE 59464,0 : POKE 59467,16 : POKE 59466,170
1010 DY=20 : DC=0
1020 IF DC>=0 THEN 1050
1030 DY=DY+1 : IF DY=256 THEN 1060
1040 DC=DC+256 : POKE 59464,DY
1050 DC=DC-DY : GOTO 1020
1060 POKE 59467,0 : POKE 59466,0 : RETURN
```

**Listing 3B**

```
( **************************************************************** )
( *                 FORTH FALLING-TONE BENCHMARK                 * )
( * THE TONE ROUTINE GENERATES CB2 TONE WITH EXPONENTIAL FALLOFF * )
( **************************************************************** )

: TONE 0 59464 C! 16 59467 C! 170 59466 C!
  20 0
  BEGIN DUP 0< IF
  SWAP 1+
  SWAP 256 + OVER 59464 C! THEN OVER 256 < WHILE
  OVER - REPEAT
  DROP DROP 0 59467 C! 0 59466 C! ;
: TEST 0 142 ! 101 1 DO
  TONE
  LOOP 142 @ INT . ." JIFFIES" ;
```

**Listing 3C**

```
100 ********************************************************************
110 *                  RPL FALLING-TONE BENCHMARK                      *
120 * ROUTINE AT LINE 1000 GENERATES CB2 TONE WITH EXPONENTIAL FALLOFF *
130 ********************************************************************
140 REM
150 0 142 ! 100 1 FOR
160 TONE &
170 NEXT 142 @ INT STR$ PRINT " JIFFIES" PRINT STOP
1000 TONE: 0 59464 POKE 16 59467 POKE 170 59466 POKE
1010 20 0
1020 LOOP: # 0 < IF
1030 % 1 + # 256 = IF . . THATSIT GOTO END
1040 % 256 + ; 59464 POKE END
1050 ; - LOOP GOTO
1060 THATSIT: 0 59467 POKE 0 59466 POKE RETURN
```

***Table 3: Results from the Shuffler Benchmark***

| | Program Bytes | Data Bytes | Jiffies | Figure of Merit |
|---|---|---|---|---|
| BASIC | 179 | 367 | 48175 | 23.15 |
| FORTH | 117 | 54 | 15136 | 4.75 |
| RPL | 70 | 52 | 5321 | 1.00 |

***Table 4:* Results from the Falling-Tone Benchmark**

| | Program Bytes | Data Bytes | Jiffies | Figure of Merit |
|---|---|---|---|---|
| BASIC | 219 | 21 | 63701 | 32.54 |
| FORTH | 150 | 0 | 5764 | 2.02 |
| RPL | 96 | 0 | 4466 | 1.00 |

Timothy Stryker may be contacted at
Samurai Software, P.O. Box 2902,
Pompano Beach, FL 33062.

## MICRObits

### 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

### MDM-2 Modem Driver Kit

Now available in kit form, an inexpensive modem driver for the VIC-20. This device will drive most modems, and some printers without a modem. A LED indicates terminal activity. Also included is the V-TERM-20 terminal program. This converts the VIC-20 into a terminal allowing you to input ESCAPE, BREAK and CONTROL characters. Kit -$31.00. Assembled and tested - $37.00. No C.O.D.

RVR Systems
P.O. Box 265
Dewitt, NY 13214

### OSI Checkwriter 65D3.3 C4PMF

User-defined one-letter code and amount to print a complete check including "memo" line. Uses "checks-to-go" or "Nebs" check systems. Has your code and balance always visible in a screen window. $19.95 includes disk, $4.95 printed listing. Specify check system.

R.I.P. Software
P.O. Box 23
2228 El Camino Real
San Mateo, CA 94403

### 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

### 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

### 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

### 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

*(Continued on page 92)*

**Olympic Sales Company**

SERVING YOU SINCE 1947

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

YOUR BANK VISA

master charge THE INTERBANK CARD

WE HONOR
VISA and MASTERCHARGE

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.) 800-421-8045
TOLL-FREE (within Calif.) 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.*

**apple computer**

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
Pascal–Fortran–Cobol 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
Micropro
Innovative
American
System Plus
and more. . .

NOW IN STOCK!

**HEWLETT PACKARD**
*2 NEW DELUXE CALCS 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

| | *Retail* | *Your Cost* |
|---|---|---|
| HP-125 New Microcomputer 64K CPU/Terminal/Keyboard | 3750.00 | 3195.00 |
| HP-85 Microcomputer—built-in printer/monitor | 3250.00 | 2499.00 |
| HP-83 Microcomputer—built-in monitor | 2250.00 | 1799.00 |
| HP-2631B Printer, dot mtrx (ask for optns) | 3950.00 | 3199.00 |
| HP82905A 80 col printer, dot matrix | 945.00 | 755.00 |
| HP-2601A Letter quality prter, daisy wheel | | 3495.00 |
| HP-82901M Dual master (256KB 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 | 385.00 | 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 | | 26.95 |
| HP-97 Progrmble printer | 750.00 | 579.95 |
| HP-34C Progrmble scientific | 150.00 | 117.95 |
| HP-38C Progrmble business R/E | 150.00 | 119.95 |

We have the complete line of accessories, etc.

**XEROX**

Model 820-1 (5¼") 8" $3750.00
64K COMPUTER & WORD PROCESSOR
AS LOW AS $2995.00
*Special discount available to Schools & Institutions—Inquire! Required software add'l.*

NEW—FAMOUS CORVUS DISK DRIVES—5, 10, 20 MEGABYTES with fantastic new OMNINET Network System
Call us for the best prices in the USA!

**Texas Instruments Home Computer**

New—1982 Model with full typewriter-style keyboard, U/L case & more!
TI-99/4A New KEYBOARD $369.95 — *Retail* $950.00 *Your Cost* $369.95
New keyboard

| | | |
|---|---|---|
| 10" color monitor for 99/4 | 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

| | Your Cost |
|---|---|
| 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 |

**ATARI Computer** Large inventory of periphls, access. etc.

| | | *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 | 869.95 |

**OHIO SCIENTIFIC**

C8PDF—48K Retail: $3495.00 Y/C: $3195.00
- Dual 8" Drives
- 64 col x 32 line/color
- .7 MIPS - *FAST!*
- Many more stnd 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-directnal/tractors | 2965.00 | 2699.00 |
| 1640K109 keyboard, tractors | 3072.00 | 2899.95 |
| 630 RO Receive only | 2710.00 | 2499.95 |
| 1650K136 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 | 995.00 | 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 | 1099.00 |
| 560G graphics | 1394.00 | 1195.00 |
| 445 | 795.00 | 695.00 |
| 445G | 894.00 | 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 | 995.00 | 789.95 |

WE ALSO HAVE . . .

| ● NOVATION 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 |

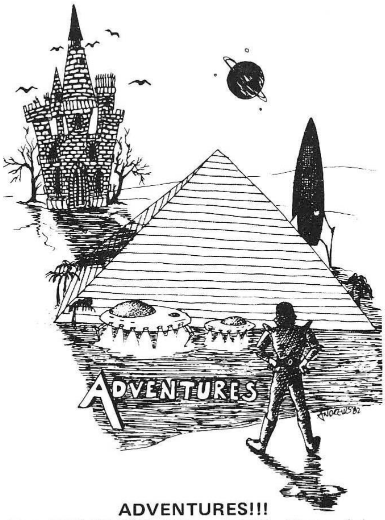
AMDEK (Leedex) High Quality Monitors

| | | | |
|---|---|---|---|
| 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, audio amp & speaker | 449.00 | 339.95 |
| Color II | 13" Color, RBG 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!

# QUALITY SOFTWARE FOR TRS-80 COLOR AND OSI



**ADVENTURES!!!**

**For TRS-80 COLOR and OSI.** These Adventures are written in BASIC, are full featured, fast action, full plotted adventures that take 30-50 hours to play. (Adventures are inter-active 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 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.

**NEW!!**

**CIRCLE WORLD** — We got Kzinti and puppeteers and problems. Our newest and biggest adventure. Requires 12k on OSI and 16K on TRS-80 Color.

**NUCLEAR SUB** — You are trapped in a nuclear sub at the bottom of the ocean. Escape and even survival is in doubt. Plotted by three of the most sadistic — I mean "creative" minds in adventure programming.

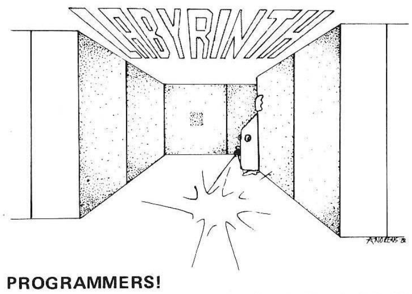
**VENTURER!**—A fast action all machine code Arcade game that feels like an adventure. Go berserk as you sneak past the DREADED HALL MONSTERS to gather treasure in room after room, killing the NASTIES as you go. Great color, high res graphics, sound and Joystick game for the TRS-80 Color or OSI machines. (black and white and silent on OSI.) Tape only. $19.95.

**BASIC THAT ZOOOMMS!!**
**AT LAST AN AFFORDABLE COMPILER FOR OSI AND TRS-80 COLOR MACHINES!!!** The compiler allows you to write your programs in easy BASIC and then automatically generates a machine code equivalent that runs 50 to 150 times faster.

It does have some limitations. It takes at least 8K of RAM to run the compiler and it does only support a subset of BASIC—about 20 commands including FOR, NEXT, END, GOSUB, GOTO, RETURN, END, PRINT, STOP, USR(X), PEEK, POKE, *, / , + , - , X , X , = , VARIABLE NAMES A-Z, A SUBSCRIPTED VARIABLE, and INTEGER NUMBERS FROM 0 - 64K.

TINY COMPILER is written in BASIC. It generates native, relocatable 6502 or 6809 code. It comes with a 20 page manual and can be modified or augmented by the user. $24.95 on tape or disk for OSI or TRS-80 Color.

**LABYRINTH** — 16K EXTENDED COLOR BASIC — With amazing 3D graphics, you fight your way through a maze facing real time monsters. The graphics are real enough to cause claustrophobia. The most realistic game that I have ever seen on either system. $14.95. (8K on OSI)



**PROGRAMMERS!**
**SEE YOUR PROGRAM IN THIS SPACE!!** Aardvark traditionally pays the highest commissions in the industry and gives programs the widest possible coverage. Quality is the keyword. If your program is good and you want it presented by the best, send it to Aardvark.

**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.

**SPACE ZAPPER** — Protect your central Star Base from ships that attack from all four sides. Fast reflexes are required as the action speeds up. Great for kids or Dads. This game has high speed high resolution graphics and looks as if it just stepped out of the arcades. — 16K extended or 32K disk. BASIC TRS-80. Color only. $14.95.

*Please specify system on all orders*

**This is only a partial listing of what we have to offer. We have arcade and thinking games, utilities and business programs for the OS1 and TRS-80 Color. We add new programs every week. Send $1.00 for our complete catalog.**

**TRS 80 COLOR**

**AARDVARK - 80**
**2352 S. Commerce, Walled Lake, MI 48088**
**(313) 669-3110**

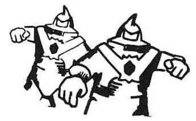
**OSI**

PRETZELLAND
SOFTWARE

**PROUDLY PRESENTS: AFFORDABLE ATARI® SOFTWARE**

ANDROID ATTACK

FIGHT YOUR WAY THROUGH LEVELS OF DEADLY ANDROID GUARDS TO SAVE THE RUNAWAY NUCLEAR REACTOR, THEN TRY TO SAVE YOURSELF!

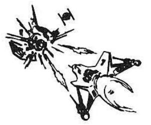


**AAARRRGGG!**

A FAST, FRANTIC CHASE AROUND THE SCREEN TRYING TO CATCH SOME CRAZY CREATURES. IF YOU CATCH THE *SUPER-AAARRRGGG!*, YOU'LL GET A SUPER BONUS, BUT DON'T GET POISONED!

STARBASE ASSAULT

HOW LONG CAN YOU PROTECT YOUR STARBASE FROM THE ATTACKING ALIEN ARMADA? EACH HIT WEAKENS YOUR FORCE FIELD RINGS AND NOW THEY'RE ATTACKING FIVE AT A TIME!



FOR 16K CASSETTE OR 24K DISK
*FAST* ACTION, SUPER GRAPHICS, ONLY $18.95 EACH
PLEASE ADD $2.00 PER ORDER FOR SHIPPING.

INTRODUCTORY OFFER:
MENTION THIS AD AND TAKE
$1.00 OFF EACH GAME!

**Pretzelland Software**
**2005 WHITTAKER**
**YPSILANTI, MI. 48197**

**FORTH for Ohio Scientific**

OSI-FORTH 3.0 is a full implementation of the FORTH Interest Group FORTH, for disk-based OSI systems (C1,C2,C3,C4,C8) Running under OS65D3, it includes a resident text editor and 6502 assembler. Over 150 pages of documentation and a handy reference card are provided. Requires 24K (20K C1P). Eight-inch or mini disk $79.95. Manual only, $9.95. "OSI-FORTH Letters" software support newsletter $4.00/year.

Other Software for Ohio Scientific Computers:

VIDEO EDITOR

Video Editor is a powerful full screen editor for disk-based C2, C4, C8 systems with the polled keyboard and color video boards (b&w monitor ok). Allows full cursor-control with insertion, deletion and duplication of source for BASIC or OSI's Assembler/Editor. Unlike versions written in BASIC, this machine-code editor is co-resident with BASIC (or the Assembler), autoloading into the highest three pages of RAM upon boot. Video Editor also provides single-keystroke control of sound, screen format, color and background color. Eight-inch or mini disk: $14.95. Specify amount of RAM.

SOFT FRONT PANEL

Soft Front Panel is a software single-stepper, slow-stepper and debugger-emulator that permits easy development of 6502 machine code. SFP is a fantastic monitor, simultaneously displaying all registers, flags, the stack and more. Address traps, opcode traps, traps on memory content and on port and stack activity are all supported. This is for disk systems with polled keyboard and color (b&w monitor ok). Uses sound and color capabilities of OSI C2/C4/C8 systems (not for C1P). Eight-inch or mini disk $24.95. Specify amount of RAM. Manual only, $4.95 (May be later credited toward software purchase). Six page brochure available free upon request.

TERMINAL CONTROL PROGRAM

OSI-TCP is a sophisticated Terminal Control Program for editing OS-65D3 files, and for uploading and downloading these files to other computers through the CPU board's serial port on OSI C2, C4 and C8 disk-based systems with polled keyboards. Thirteen editor commands allow full editing of files, including commands for sending any text out the terminal port and saving whatever text comes back. INDUTL utility included for converting between BASIC source and TCP file text. Eight-inch or mini disk $39.95. Manual only, $2.95.

WRITE FOR FREE CATALOG!
Prices shown are postpaid.
Specify computer model & RAM.

******* NEW ADDRESS ******
Technical Products Company
P.O. BOX 9053
Boone, NC 28608

**"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 $55.00
master diagnostics +plus $75.00
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
APPLESOFT CARD ROM TEST
INTEGER CARD ROM TEST
MOTHERBOARD RAM TESTS
PARALLEL CARD PROM TESTS
DISK DRIVE SPEED CALIBRATION
DISK HEAD READ/WRITE TEST
DISK WRITE PROTECT SWITCH TEST
DISK DRIVE HEAD CLEANING ROUTINES
DISK DRIVE MAINTENANCE
DC HAYES MICROMODEM II TEST
(EPROM & HARDWARD)
MONITOR SKEWING TESTS
MONITOR & MODULATOR CALIBRATION
MONITOR TEXT PAGE TEST
MONITOR TEST PATTERN
MONITOR & TV YOKE ALIGNMENT
LO-RES COLOR TESTS
HI-RES COLOR TESTS
RANDOM HI-RES COLOR GENERATOR
LISSAJOUS PATTERNS
SPEAKER FUNCTION TESTS
SQUARE WAVE MODULATION
PADDLE & SPEAKER TESTS
PADDLE & BUTTON TEST
PADDLE DRIFT TEST
INTERNAL MAINTENANCE
GENERAL MAINTENANCE
ON BOARD HELP
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: 1-800-362-2421
Add: $5.00 outside U.S. and Canada

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

# Tiny PILOT for the PET

### by Jim Strasma and John O'Hare

**This program offers PET owners a machine language PILOT, based on Nick Vrtis's program for the SYM (MICRO 16:41). In addition, a few commands have been added and PET screen editing has been implemented.**

**Tiny PILOT**
requires:

PET
8K or larger
O.S. 2.0

This PET version of PILOT, like its parallel program on the SYM, isn't as full a PILOT as the version sold by Apple Computer Co. However, it is a good start, and you can develop it into something very impressive, given a bit of time.

If you'd rather not type in this lengthy listing, and have either the ASM/TED or MAE assembler and a CBM disk, contact the ASM/TED Users' Group (ATUG) librarian, Brent Anderson, for disk 'UE'. If your copy of ASM/TED can't read MAE disk files, include your ASM/TED serial number from the front of your manual as proof of purchase, and ask for the disk containing "MOSER". Write to ATUG at 200 South Century, Rantoul, IL 61866.

## For the Adventurous

There is a way to disable the "crunch" routine in PET BASIC that makes the quotation mark necessary at the start of each line of PILOT. Like the routine that disables the PET's [stop] key, it merely jumps three bytes beyond the usual entry point in BASIC when inserting a new line into a program. In BASIC 4.0, the usual entry is at $B419 and the needed value is $B41C.

This technique is used in the Editor of CBM's 6502 development program disk. But the patch must be initialized before beginning work on a PILOT program. With that in mind, you may prefer to keep using the quote marks.

***Table 1:* PET TINY PILOT**

PET TINY PILOT is based on an original program for the SYM computer, by Nick Vrtis. The PET conversion and extensions are by John O'Hare, aided by Jim Strasma.

**PET TINY PILOT Program Statements**

| | |
|---|---|
| T:TEXT | Display the text on screen |
| A: | Input up to 40 characters into the answer field |
| ?: | Input up to 40 characters into both the name and answer fields |
| TEXT | Compare text to last input and set yes/no flag |
| J:N | Jump to label N for next line |
| | J:A means jump to last accept |
| | J:* means restart program |
| U:N | Use subroutine labeled N |
| S: | Stop program and return to edit mode |
| C: | Compute, performs "+", "−", and "=" on variables "A"-"Z" |
| R: | Remarks — are not executed |
| P:X | Put a random number between 1 and 99 in variable "X" (Call three times at the start of your program to initialize correctly. Otherwise, result may be outside correct range.) |
| I:X | Input number into variable "X" |
| L:F-Z | Call machine-language routine |
| L:A | Clear screen |
| L:B | Reverse screen foreground and background |
| L:C | Scroll up |
| L:D | Scroll down |
| L:E | Home cursor to top left of screen |
| L:F | Use machine-language program in second cassette buffer address 826) (Do *not* use "L:G" through "L:Z" as commands. They are reserved for future features, and will crash your program if used.) |
| F:U | Set to graphic mode (upper-case) |
| F:L | Set to text mode (lower-case) |
| D: | Delay about five seconds |
| W: | Wait for key to be pressed |

| | |
|---|---|
| Conditionals | (May precede any statement. Execution only if condition satisfied.) |
| Y | Execute if match flag = Y |
| N | Execute if match flag = N (i.e. YT:TEXT, NJ:N) |
| Label | |
| *A | Labels current location 'A' |
| Variables within Text | Causes contents of named variable to be matched or displayed |
| $? | Same as $X — applies to Name field |

## Notes on TINY PILOT

### Getting It Working

The interpreter's object code is located at addresses $7800 through $7F91 hex. This version is for PET or CBM computers with BASIC 2.0 (sometimes called 3.0) only. BASIC 1.0 and 4.0 versions are feasible, but have not yet been attempted. If you wish to do the conversion, we recommend getting a copy of the source code from the ASM/TED Users' Group.

Users with the upgrade BASIC 2.0 ROMs may either type in PET TINY PILOT from the hex listing in this article, or request it on disk from ATUG.

### Writing Programs

PET TINY PILOT programs are keyed in like a BASIC program, but each line number must be followed by a single quotation mark:

```
10 "T:TINY PILOT TEST
20 "S:
```

After loading TINY PILOT from BASIC like any other program, enter "new: sys32512". This is important! Without the "new", PET will think you are out of memory. And without the "sys" call, Tiny PILOT will not be protected from BASIC.

Once the TINY PILOT program has been loaded and initialized, TINY PILOT programs may be loaded, saved, edited, and listed just like PET BASIC programs, using the usual syntax. They may even be written, edited, and saved without PET TINY PILOT loaded. To run a program, simply type SYS30721.

### Using the Match (M:) Statement

"M:YE,SURE will match with "YES", "YEP","SURE", and "SURELY", but not with "YA", "SUROUND" , or "NO".

### Example Match Formats:

M:YES,,YA,OK
M:NO
M:- (Checks for a negative number)

PILOT has two special error messages:

1. ERR, with the specific error listed here;
2. X LABEL NOT FOUND, where X is a TINY PILOT label.

**Sample Program**

```
10 "R:GUESS PROGRAM
20 "P:R
30 "T:GUESS MY NUMBER (999 TO QUIT)
40 "*LT:YOUR GUESS
50 "I:G
60 "C:$=G
70 "M:999
80 "YT:OK
90 "YJ:E
100 "C:N=R-G
110 "C:$=N
120 "C:S=S+1
130 "M:0
140 "YT:YOU GOT IT!
150 "YJ:E
160 "M:-
170 "YT:TOO BIG
180 "NT:TOO SMALL
190 "T:
200 "J:L
210 "*ET:IT TOOK $S GUESSES
220 "T:
230 "T:MY NUMBER - $R
240 "S:
```

Contact Jim Strasma at P.O. Box 647, Pawnee, IL 62558. Write to John O'Hare at P.O. Box 157, Lemont, IL 60439.

*GOSUB International, Inc.* **presents**

The **CARDBOARD** for the VIC 20! The **CARDBOARD** is an expansion motherboard for use with Commodore's VIC 20 series computers. It has six slots that will accept any VIC-compatable cartridge in any configuration.

Increase RAM up to 40K and use several utility ROMs plus have several games online, all switch selectable!

The **CARDBOARD** can be daisy-chained, giving the user an almost unlimited number of available expansion slots.

A system reset switch has also been added to the **CARDBOARD** allowing the user to select and/or restart games on ROM without turning off the computer.

*All this for only $119.95 plus $1.50 S/H*

To order send check or money order to:

**GOSUB INTERNATIONAL, INC.**
**501 E. PAWNEE, SUITE 430**
**WICHITA, KANSAS 67211**
**(316) 265-9858**

*VISA and Master Card phone orders also accepted.*

**RPL**

**RPL** is a fast, space-efficient language, designed for the PET/CBM user who wants to develop high-speed, high-quality software with a minimum of effort. While ideal for programming games and other personal applications, it is primarily oriented toward real-time process control, utility programming, and similar demanding business and industrial uses.
R. Vanderbilt Foster, of Video Research Corporation, says he thinks that **"RPL is one HELL of a system!"** (capitals his). Ralph Bressler, reviewing the package in The Paper, says "I know of few language systems this complete, this well documented, for this kind of price." For more information, see the following:

**MICRO, Dec. '81, p. 35**
**MICROCOMPUTING, Feb. '82, p. 10**
**MICRO, Mar. '82, p. 29**
**BYTE, Mar. '82, p. 476**
**COMPUTE!, Mar. '82, pp. 45, 120.**

See also the article **"Basic, Forth and RPL"** in the June '82 issue of MICRO, and Mr. Bressler's review in the Jan./Feb. '82 issue of The Paper. Don't let our prices deceive you: **RPL** is a first-class, high performance language in every respect. We are keeping its price so low in order to make it accessible to the widest possible number of users. Only **$80.91**, postpaid, for both the **RPL** compiler and its associated symbolic debugger, complete with full documentation (overseas purchasers please add $5.00 for air mail shipping). Versions available for PET-2001 (Original, Upgrade or V4.0 ROM's), CBM 4032, and CBM 8032/8096, on cassette, 2040/4040, and 8050 disk.

**Order Anytime, Day or Night 7 Days A Week**
**800-327-8965**
(ask for extension 2)

**VISA**
**Master Charge**
**American Express**

**Samurai Software**
**P.O. Box 2902**
**Pompano Beach, Florida 33062**
**(305) 782-9985**

**Tiny PILOT Listing**

```
7800 EA A9 00 8D 8A 7D 20 ED
7808 7B A2 33 A9 00 8D 85 7D
7810 9D 42 7D CA 10 FA A5 97
7818 C9 04 D0 03 4C 89 C3 B1
7820 88 B1 88 C9 2A D0 04 C8
7828 C8 D0 EB C9 59 F0 04 C9
7830 4E D0 12 CD F1 7C F0 F0
7838 20 BF 7B AD 8A 7D F0 03
7840 4C 89 C3 B0 D1 8D 76 7D
7848 C8 C8 C9 4C D0 14 20 B6
7850 7C 20 02 7C 8A 4A AA E8
7858 E0 01 D0 09 20 29 E2 4C
7860 88 78 4C D3 78 E0 02 D0
7868 18 A2 80 A0 00 84 20 86
7870 21 B1 20 49 80 91 20 C8
7878 D0 F7 E8 E0 84 D0 F0 F0
7880 07 E0 03 D0 09 20 3F E5
7888 20 C8 7C 4C 38 78 E0 04
7890 D0 29 A0 28 84 22 A0 00
7898 84 20 A0 BF A2 83 86 21
78A0 86 23 B1 20 91 22 88 C0
78A8 FF D0 F7 CA E0 7F D0 EE
78B0 A0 27 A9 20 91 20 88 10
78B8 FB 30 CD E0 05 D0 08 A9
78C0 13 20 D2 FF 4C 88 78 E0
78C8 06 D0 06 20 3A 03 4C 88
78D0 78 A9 4C C9 50 D0 23 20
78D8 02 7C 8E 86 7D F8 38 A5
78E0 55 65 58 65 59 85 54 A2
78E8 04 B5 54 95 55 CA 10 F9
78F0 D8 AE 86 7D 9D 43 7D 4C
78F8 38 78 C9 49 D0 42 A9 3F
7900 20 AA 7C A9 00 8D 78 7D
7908 8D 79 7D 20 8C 7C C9 0D
7910 F0 19 38 E9 30 A2 04 18
7918 0E 79 7D 2E 78 7D CA D0
7920 F6 18 6D 79 7D 8D 79 7D
7928 4C 0B 79 20 02 7C AD 78
7930 7D 9D 42 7D AD 79 7D 9D
7938 43 7D 20 A1 7C 4C 38 78
7940 C9 46 D0 1D B1 88 8D 89
7948 7D C9 55 D0 08 A9 0C 8D
7950 4C E8 4C 38 78 AD 89 7D
7958 C9 4C D0 F6 A9 0E 4C 4F
7960 79 C9 44 D0 0E A9 00 85
7968 8F 85 8E A5 8E C9 01 D0
7970 FA F0 DF C9 57 D0 10 20
7978 B6 7C 20 E4 FF C9 00 F0
7980 F9 20 C8 7C 4C 38 78 C9
7988 3F D0 06 38 6E 76 7D D0
7990 0E C9 41 D0 39 A5 88 8D
7998 EF 7C A5 89 8D F0 7C A9
79A0 3F 20 AA 7C A2 27 20 8C
79A8 7C C9 14 D0 03 E8 D0 F6
79B0 C9 0D D0 02 A9 00 9D F2
79B8 7C 2C 76 7D 10 03 9D 1A
79C0 7D C9 00 F0 03 CA 10 DE
79C8 20 A1 7C 4C 38 78 C9 43
79D0 F0 03 4C 7D 7A 20 02 7C
79D8 8E 7C 7D A9 00 8D 7A 7D
79E0 8D 7B 7D C8 A2 2B D0 59
79E8 C8 B1 88 30 22 C9 2F 90
79F0 1E C9 3A B0 14 29 0F 6A
79F8 6A 6A 6A A2 04 2E 79 7D
7A00 2E 78 7D 0A CA D0 F6 F0
7A08 DF 20 0A 7C 4C E8 79 F8
7A10 AA AD 7E 7D C9 2D F0 16
7A18 18 AD 79 7D 6D 7B 7D 8D
7A20 7B 7D AD 78 7D 6D 7A 7D
7A28 8D 7A 7D 4C 41 7A 38 AD
7A30 7B 7D ED 79 7D 8D 7B 7D
7A38 AD 7A 7D ED 78 7D 8D 7A
7A40 7D D8 8E 7E 7D 8A F0 0C
7A48 30 0A A9 00 8D 78 7D 8D
7A50 79 7D F0 94 AE 7C 7D 10
7A58 15 A2 38 20 0D 7C 20 1D
7A60 7C A2 04 BD 7F 7D 9D 15
7A68 7D CA 10 F7 30 0C AD 7B
7A70 7D 9D 43 7D AD 7A 7D 9D
7A78 42 7D 4C 38 78 C9 4D D0
7A80 57 88 C8 A2 27 B1 88 F0
```

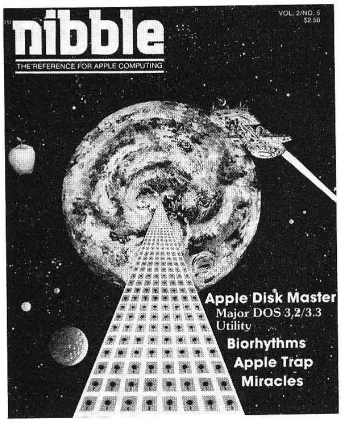
**Tiny PILOT Listing** *(continued)*

```
7A88 09 DD F2 7C D0 08 C8 CA
7A90 10 F3 A2 59 D0 3D C9 24
7A98 F0 14 C9 2C F0 F4 C8 B1
7AA0 88 F0 2E C9 2C F0 DB D0
7AA8 F5 AC 77 7D D0 F0 C8 8E
7AB0 7C 7D 20 1A 7C AE 7C 7D
7AB8 8C 77 7D A0 04 B9 7F 7D
7AC0 F0 09 DD F2 7C D0 E2 CA
7AC8 88 10 F2 AC 77 7D C8 D0
7AD0 B4 A2 4E 8E F1 7C D0 A2
7AD8 C9 55 D0 1B B1 88 48 20
7AE0 BF 7B AD 8A 7D F0 03 4C
7AE8 89 C3 A5 88 8D 84 7D A5
7AF0 89 8D 85 7D 68 D0 06 C9
7AF8 4A D0 50 B1 88 8D 76 7D
7B00 C9 2A F0 40 C9 41 D0 0E
7B08 AD EF 7C 85 88 AD F0 7C
7B10 85 89 A0 00 F0 68 20 F0
7B18 7B B1 88 C9 2A D0 08 C8
7B20 B1 88 CD 76 7D F0 20 20
7B28 BF 7B AD 8A 7D F0 13 AD
7B30 76 7D 20 AA 7C A0 7F A9
7B38 7E 20 1C CA 20 A1 7C 4C
7B40 89 C3 B0 D5 4C 06 78 C8
7B48 4C 7E 7B C9 53 D0 03 4C
7B50 89 C3 C9 45 D0 15 AD 85
7B58 7D F0 14 85 89 AD 84 7D
7B60 85 88 A0 00 A9 00 8D 85
7B68 7D F0 13 C9 52 D0 03 4C
7B70 38 78 C9 54 F0 05 88 88
7B78 20 D7 7C 20 89 7B AD 8A
7B80 7D F0 03 4C 89 C3 4C 16
7B88 78 B1 88 F0 2F C9 24 D0
7B90 24 C8 B1 88 C9 3F F0 10
7B98 20 1A 7C A2 04 BD 7F 7D
7BA0 F0 16 20 AA 7C CA 10 F5
7BA8 A2 27 BD 1A 7D F0 09 20
7BB0 AA 7C CA 10 F5 20 AA 7C
7BB8 C8 4C 89 7B 20 A1 7C B1
7BC0 88 F0 04 C8 4C BF 7B C8
7BC8 B1 88 C8 11 88 D0 06 A9
7BD0 01 8D 8A 7D 60 C8 C8 C8
7BD8 B1 88 C9 22 D0 01 C8 98
7BE0 18 65 88 85 88 90 02 E6
7BE8 89 A0 00 38 60 20 A1 7C
7BF0 A0 01 84 88 8C EF 7C A9
7BF8 04 85 89 8D F0 7C A0 FF
7C00 D0 C5 B1 88 38 E9 41 0A
7C08 AA 60 20 02 7C BD 43 7D
7C10 8D 79 7D BD 42 7D 8D 78
7C18 7D 60 20 0A 7C 10 1C A9
7C20 2D 8D 83 7D F8 38 A9 00
7C28 ED 79 7D 8D 79 7D A9 00
7C30 ED 78 7D 8D 78 7D D8 A2
7C38 03 D0 02 A2 04 18 6E 7D
7C40 7D AD 78 7D 20 63 7C AD
7C48 79 7D 4A 4A 4A 4A 20 63
7C50 7C AD 79 7D 20 63 7C 2C
7C58 7D 7D 30 01 CA A9 00 9D
7C60 7F 7D 60 29 0F 09 30 9D
7C68 7F 7D 2C 7D 7D 30 05 C9
7C70 30 D0 01 60 38 6E 7D 7D
7C78 CA 60 C9 03 D0 03 4C 89
7C80 C3 C9 93 D0 22 A9 3E 20
7C88 AA 7C A0 00 20 B6 7C 20
7C90 E4 FF 20 D2 FF 8D 89 7D
7C98 20 C8 7C C9 00 F0 ED D0
7CA0 D9 20 B6 7C 20 E2 C9 4C
7CA8 C8 7C 20 B6 7C 20 D2 FF
7CB0 8D 89 7D 4C C8 7C 8C 87
7CB8 7D 8E 86 7D 8D 89 7D 08
7CC0 68 8D 88 7D AD 89 7D 60
7CC8 AC 87 7D AE 86 7D AD 88
7CD0 7D 48 AD 89 7D 28 60 A9
7CD8 45 20 AA 7C A9 52 20 AA
7CE0 7C 20 AA 7C A9 2D 20 AA
7CE8 7C A9 01 8D 8A 7D 60 1D
7CF0 1D 1D 11 20 2D 32 2D 32
7CF8 1D 1D 1D 1D 1D 1D 11 11
7D00 20 2D 32 2D 32 1D 1D 1D
7D08 1D 1D 11 11 11 11 11 11
```

**Tiny PILOT Listing** *(continued)*

```
7D10 11 11 20 2D 32 2D 32 1D
7D18 1D 1D 1D 1D 11 20 2D 32
7D20 2D 32 1D 1D 11 11 11 11
7D28 20 2D 31 2D 31 1D 1D 1D
7D30 1D 1D 11 11 11 20 2D 31
7D38 2D 31 1D 1D 11 20 2D 31
7D40 2D 31 1D 1D 1D 1D 1D 1D
7D48 11 11 11 11 11 11 11 20
7D50 2D 31 2D 31 1D 1D 1D 1D
7D58 1D 1D 1D 11 11 11 20 2D
7D60 31 2D 31 1D 1D 1D 1D 11
7D68 11 11 11 11 11 11 11 20
7D70 20 30 20 30 1D 1D 1D 11
7D78 11 20 20 30 20 30 1D 1D
7D80 1D 11 11 11 20 20 30 20
7D88 30 1D 1D 1D 11 11 11 11
7D90 11 11 11 11 20 20 30 20
7D98 30 1D 1D 1D 1D 1D 1D 1D
7DA0 1D 11 11 20 20 30 20 30
7DA8 1D 11 11 11 11 11 11 20
7DB0 20 30 20 30 1D 1D 1D 1D
7DB8 1D 1D 1D 1D 11 11 11 11
7DC0 11 20 20 31 20 31 1D 1D
7DC8 1D 1D 1D 11 11 11 11 11
7DD0 11 20 20 31 20 31 1D 1D
7DD8 1D 1D 11 11 11 11 11 11
7DE0 11 20 20 31 20 31 1D 1D
7DE8 1D 1D 1D 1D 11 11 11 11
7DF0 11 20 20 31 20 31 1D 1D
7DF8 1D 11 20 20 31 20 31 1D
7E00 1D 1D 1D 1D 1D 1D 1D 11
7E08 11 11 20 20 32 20 32 1D
7E10 1D 1D 11 11 11 11 11 11
7E18 11 20 20 32 20 32 1D 1D
7E20 11 11 11 11 11 11 11 11
7E28 20 20 32 20 32 1D 1D 1D
7E30 1D 1D 1D 11 11 11 11 20
7E38 20 32 20 32 1D 11 11 11
7E40 20 20 32 20 32 1D 1D 11
7E48 11 11 11 11 11 20 20 32
7E50 20 32 1D 1D 1D 1D 1D 1D
7E58 1D 1D 11 11 11 11 11 11
7E60 11 20 20 33 20 33 1D 1D
7E68 11 11 11 11 11 11 11 20
7E70 20 33 20 33 1D 11 11 11
7E78 11 11 20 20 33 20 33 1D
7E80 1D 1D 1D 11 20 20 33 20
7E88 33 1D 1D 1D 1D 1D 1D 1D
7E90 11 11 11 11 20 20 33 20
7E98 33 1D 1D 11 11 20 20 34
7EA0 20 34 1D 11 11 11 11 20
7EA8 20 34 20 34 1D 1D 1D 1D
7EB0 11 11 11 11 11 11 20 20
7EB8 34 20 34 1D 1D 1D 1D 1D
7EC0 11 11 11 11 20 20 34 20
7EC8 34 1D 1D 1D 1D 1D 1D 1D
7ED0 1D 11 20 20 35 20 35 1D
7ED8 1D 1D 11 11 11 11 11 20
7EE0 20 35 20 35 1D 1D 1D 1D
7EE8 1D 1D 11 20 20 35 20 35
7EF0 1D 11 20 20 35 20 35 1D
7EF8 1D 1D 1D 11 11 11 11 11
7F00 A9 4C 85 70 A9 29 85 71
7F08 A9 7F 85 72 A9 78 85 35
7F10 C6 35 A0 7F A9 46 20 1C
7F18 CA A2 00 B5 53 10 02 49
7F20 80 95 53 E8 E0 07 D0 F3
7F28 60 E6 77 D0 02 E6 78 8C
7F30 87 7D A0 00 B1 77 AC 87
7F38 7D C9 8A F0 03 4C 76 00
7F40 20 01 78 4C 89 C3 93 23
7F48 23 23 20 50 45 54 20 50
7F50 49 4C 4F 54 20 23 23 23
7F58 0D 0D 20 42 59 2D 20 4A
7F60 4F 48 4E 20 4F 27 48 41
7F68 52 45 0D 0D 20 41 4E 44
7F70 20 4A 49 4D 20 53 54 52
7F78 41 53 4D 41 0D 00 20 4C
7F80 41 42 45 4C 20 4E 4F 54
7F88 20 46 4F 55 4E 44 3F 00
7F90 11 AA AA AA AA AA AA AA
```

# "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.

# Microcomputer Interfacing: FORTH *vs.* BASIC

*by Mark Bernstein*

**BASIC and FORTH versions of programs to control a digitizer interface are compared line by line. The FORTH versions are not only faster and more space-efficient, but they are also easier to understand.**

The routines, as written, require:

PET (all versions) with FORTH
Houston Instruments 'HIPAD' Digitizer
However, with slight modification, the program will run on any computer with a parallel port.

FORTH, widely considered an unusual and peculiar language, is not difficult to learn. FORTH interpreters are available for all major personal computers and, since FORTH is easy to implement, these interpreters are usually inexpensive. FORTH programs run substantially faster than programs written in microcomputer BASIC, making the language especially attractive for system programming and language implementation.

Critics object that FORTH programs are difficult to understand or modify. It has been called a write-only language, unsuitable for significant programming tasks.

If FORTH programs were especially difficult to read, FORTH's usefulness in most applications would be questionable. Careful and considerate programming, however, can produce FORTH programs that are *more* legible than their BASIC/assembly language equivalents. Indeed, many FORTH programs are easier to debug and modify than their BASIC kindred.

To demonstrate a typical FORTH application, we will discuss an interface between a Commodore PET computer and a Houston Instruments HIPAD digitizer (figure 1). *(Editor's Note: PET and AIM interfaces to Summagraphics' BIT PAD and BIT PAD ONE were discussed in the July, 1981 issue of MICRO.)* This popular and inexpensive device allows an operator to transfer information from pictures, drawings, charts, or photographs to a small computer. To use the digitizer, the operator simply places the digitizer's pointer at any point of the pad's 11'' × 11'' *active area.* A built-in 8748 microcomputer measures the pointer's position, converts it into either inches or millimeters, and transmits the pointer's coordinates to the master computer.

Scientists and engineers use the digitizer to translate charts and spectra into computer-readable form. Designers and planners can use digitizers to make and revise graphics, plans, and diagrams. Since digitizers are durable and easy to understand, they are popular in schools. Digitizers may be especially important to handicapped people, making computers accessible to those who cannot conveniently use a keyboard.

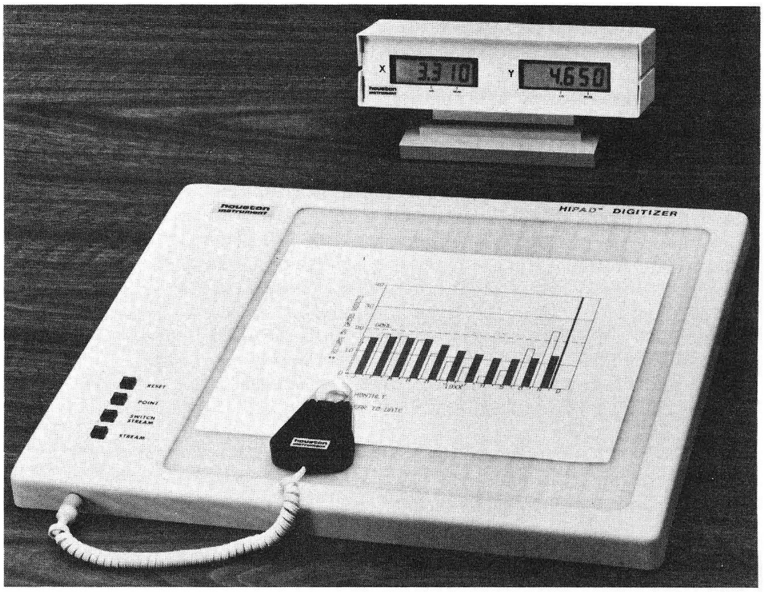
Details of the PET-HIPAD interface are covered under ''The Digitizer Interface'' on page 84.

### Initializing the Input Port

The computer's first task is the proper initialization of the PET's user port. The VIA is a complicated device; we won't attempt to explain all its functions and capabilities here. To properly initialize the VIA for *this* task, we must perform the following steps:

1. Define Port A as an input port by storing the value 00 in the *data direction register.*
2. Set CB2 (ACK) low by storing the binary value 110 in bits five to seven of the *control register.*
3. Request that the port observe the digitizer's STROBE signal, and report low-to-high transitions. To request this function, we set bit zero of the *control register* to one.

*(Continued on page 79)*



## The Digitizer Interface

Houston Instruments designed the HIPAD to be compatible with several different types of interfaces. ''Parallel binary with handshaking'' is the fastest option available, and allows us to connect the digitizer directly to the PET's parallel user port *via* the cable specified in figure A. Any other computer with a VIA could be used.

Each time the digitizer makes a measurement, it sends a coordinate pair (X,Y) to the computer using the protocol shown in figure B. The two coordinates are transmitted in five individual bytes. The first byte declares the start of a transmission and identifies the pad's present operating mode. Bytes two and three contain the measured X coordinate and bytes four and five contain the measured Y coordinate.

The computer can identify the initial control byte (byte one) since only this first byte has its most significant bit set to 1. If the computer tries to start listening to the digitizer in the middle of a transmission, an unexpected control byte will warn it of its mistake.

Two handshake control signals regulate the transfer of each byte from the digitizer to the computer (figure C). When the digitizer wants to send a byte to the computer, it transmits a pulse over the (normally low) STROBE line. The computer responds in turn by sending a pulse over the (normally low) ACK (ACKNOWLEDGE) line.

The digitizer transmits a STROBE pulse to tell the computer that new data are ready and waiting at the input port. The computer responds by sending an ACK pulse, which tells the digitizer the computer has read the data sent and is ready for more.

The PET's user port, a 6522 Versatile Interface Adapter (VIA) is programmed by storing numbers into its sixteen control registers (table 1). The digitizer's STROBE signal is connected to the PET's CA1 input, which we program to set a flag bit in the port's *interrupt flag register* whenever it observes an *active transition* from low-to-high. The digitizer's eight data lines are connected to the VIA's Port A, and may be read by examining the contents of the port's *data register*. Finally, the PET's CB2 output generates the ACK pulse to acknowledge a successful data transfer.

The digitizer's timing specifications require that the ACK signal's duration be between 20 and 50 microseconds. We generate this signal by using the VIA's shift register. To send an ACK pulse, we store the binary value 1000 0000 into the shift register. The shift register's timer is then set to shift a new bit out on CB2 every 40 microseconds. The shift register begins with the leftmost (most significant) bit, so it sends a 1 pulse on CB2. 40 microseconds later, it sends the next bit, a 0 pulse. After another 40 microseconds, the computer sends the next bit, also a 0. The process continues until all eight bits have been sent, at which point the operation stops. The entire operation thus sends a single 40-microsecond pulse on the CB2 (ACK) line (figure D).

Since the VIA shift register operates autonomously, the PET is free to perform other computations while the ACK pulse is sent. The length of the ACK pulse, moreover, is not in any way affected by the language used to implement the interface, but is determined only by the value stored in the VIA timer register.

**_Figure A:_ PET to HIPAD Interface**

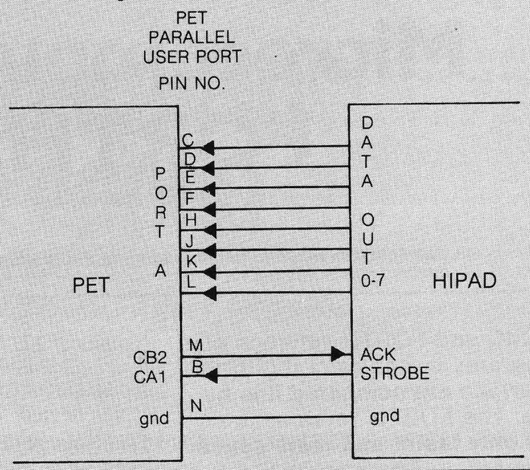


**_Figure B:_ Transmission Sequence**

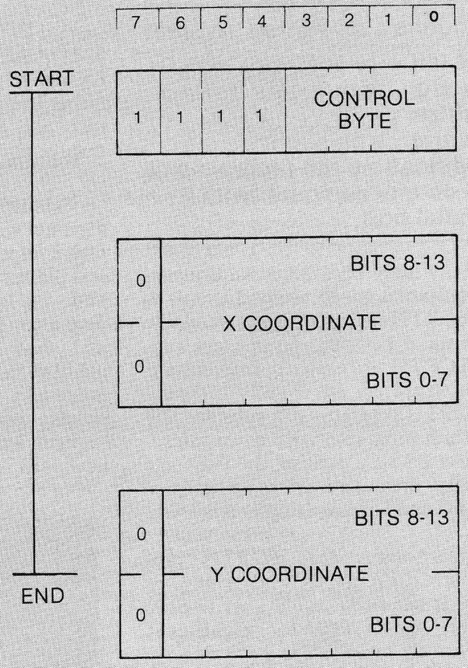


**_Figure C:_ Control Signal Timing**

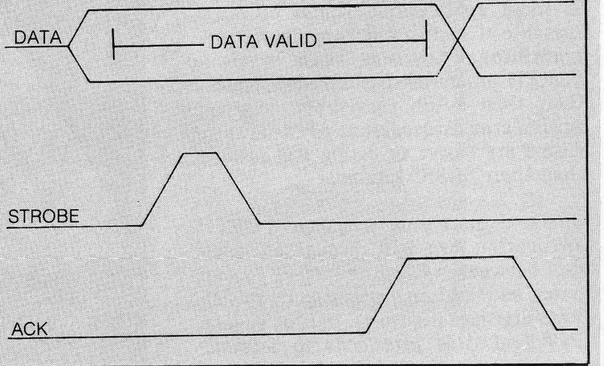


***Table 1:* The PET computer's user I/O port is controlled by storing values into sixteen memory locations, called registers, located inside the VIA chip. The eight registers used by the digitizer interface are described above.**

| Address | Name | Purpose |
|---|---|---|
| E841 | data register | Data from the digitizer appear here. |
| E843 | data direction register | Specifies whether each bit of the data register is to be an input or an output bit. Each bit is configured as an input (set to zero). |
| E84A | shift register | Data to be transmitted serially over the CB2 (ACK) output are stored here. |
| E848 | timer #2 low-order | The rate at which the shift register operates is specified here. |
| E849 | timer #2 high-order | This byte specifies the high-order 8 bits for the shift register rate (zero here). Writing to this register starts the shift register. |
| E84B | auxiliary control | Setting bit 0 of this register enables inputs to the data register |
| E84C | control register | Various bits in this register specify the behavior of the port's control inputs and outputs. For the digitizer interface, we must configure CA1 (STROBE) and CB2 (ACK). |
| E84D | interrupt flag register | When the digitizer transmits a STROBE pulse, the port automatically sets bit 1 of this register to 1. Reading data from the data register clears this bit to 0. Hence, bit 1 indicates whether new data from the digitizer are ready to be read. |

*(Continued from page 77)*

4. Activate the input port by writing a one to bit zero of the *auxiliary control register.*

Simple subprograms in FORTH and BASIC that perform these tasks are given in listing 1. In BASIC, we define the constant VI for the base address of the VIA, and must be careful that this value is *never changed*. In FORTH, we can define fixed and unchangeable CONSTANTs, which cannot be changed at some other point in the program. FORTH, moreover, can handle names of up to 31 characters, helping to clarify the program. CTRL-REG is clearly more suggestive than VI+12, just as the FORTH command VIA-SET is cleaner than the BASIC equivalent

```
GOSUB 1000 : REM INITIALIZE VIA
```

Notice, too, that FORTH moves easily between different numerical bases. By allowing programmers to express themselves in binary, hex, or octal when appropriate, FORTH makes programs easier to decipher and debug.

## Sending the ACK Signal

Next we turn to the task of generating the ACK signal. As discussed above, we use the VIA's shift register to produce a 40-microsecond pulse. To generate this pulse, we must perform the following steps:

1. Enable the shift register. Set it to transfer one bit every time the VIA's Timer #2 reaches zero, by setting bits two and four of the VIA *auxiliary control register.* Recall that bit zero was set to one by the VIA initialization routine; the auxiliary control register should now contain the binary value of 0001 0101.

2. Store the binary value 1000 0000 into the *shift register.*

3. Set Timer #2 to shift one bit onto the CB2 output every 40 microseconds by storing a decimal 20 (one-half the number of microseconds) into the Timer #2 *low-order byte,* and then storing a zero into the Timer #2 *high-order byte.* This last step automatically begins the transmission sequence; no further intervention is required.

Routines to create the ACK pulse are shown in listing 2. Once more, the FORTH program is at least as clear as the BASIC code. In FORTH, we define several new constants for various VIA addresses, including SHIFT-REG, TIMER-LO, and TIMER-HI. We also define the variable ACK-TIME, which specifies the duration of the ACK pulse. Clearly, "ACK-TIME" is better than AT — its BASIC equivalent — just as SHIFT-REG is better than either SR or VI+8. ACK! is more suggestive than

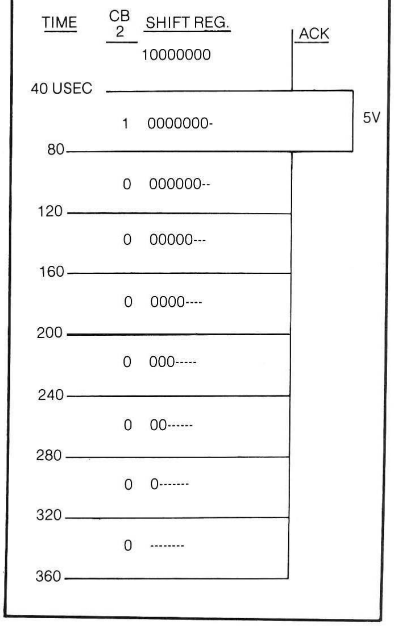
```
GOSUB 1500: REM SEND ACK PULSE
```

## The STROBE Signal

The digitizer transmits a STROBE pulse to indicate that data are available and ready to be read. The STROBE signal is connected to the PET's CA1 input which, in turn, controls a flag bit in the VIA *interrupt flag* register. If the computer is waiting for data from the digitizer, it simply needs to wait for the appropriate interrupt flag. When the flag is set, and only then, the computer may read a data byte from the *data register.*

FORTH and BASIC routines that wait for the STROBE signal are given in listing 3. The two languages handle this task in similar ways, producing

***Figure D:* Timing of the ACK Pulse**



similar programs. The FORTH program's structure is clearer, since its BEGIN...UNTIL structure eliminates the need for an undesireable GOTO. However, the BASIC program specifies more clearly the conditions for terminating the process.

## Reading Bytes

Next we combine these routines to read data bytes from the digitizer. To read any byte, the computer must first await the digitizer's STROBE pulse. It then reads a byte from the data register and acknowledges receipt by sending out an ACK pulse.

Listing 4 shows BASIC and FORTH routines to read bytes from the digitizer. The first routine reads a single byte from the digitizer. The second routine hunts for the start of a digitizer transmission, reading (and discarding) data until it finds a control byte. The third routine uses these two subroutines to read an entire transmission, including one control byte and four data bytes.

Note that the FORTH versions are substantially more concise. SYNC's BEGIN...UNTIL loop is quite clear; SYNC explicitly waits *until* it finds a control byte. PAD@, which reads an entire five-byte transmission from the HIPAD, uses FORTH's DO...LOOP structure, equivalent to BASIC's FOR ...NEXT loop:

| **BASIC** | **FORTH** |
|---|---|
| FOR I = 1 TO 4 | 5 1 DO |
| ... | ... |
| NEXT I | LOOP |

## The Digitizer Device Driver

In listing 5, we finally come to the routine that connects the user's programs with the digitizer interface. Calling this subroutine reads one point from the digitizer, storing the coordinates in the variables X% and Y%.

Here, for the first time, BASIC is clearer than FORTH. Only in this routine does FORTH's Reverse Polish Notation cause any real obscurity; fortunately, the problem is not very serious. The FORTH phrase

```
HEX
80 * +
```

is simply equivalent to the BASIC

```
( )*128 + ( )
```

***Listing 1:*** **Initializing the VIA input port.**

```
FORTH                                BASIC
HEX ( VIA ADDRESSES)                 1000 REM === VIA INITIALIZATION
E843 CONSTANT DIR-REG                1010 VI=59456 :REM VIA BASE ADDRESS
E84C CONSTANT CONTROL-REG
E841 CONSTANT DATA-REG
E84B CONSTANT AUX-REG

( INITIALIZE VIA )
BINARY
: VIA-SET
   00000000 DIR-REG  C!              1020 POKE VI+3,0 : REM DATA DIRECTION
   11001101 CTRL-REG C!              1030 POKE VI+12,205 : REM CONTROL REG
   00000001 AUX-REG  C! ;            1040 POKE VI+11,1 : REM AUX. REG
                                     RETURN
```

***Listing 2:*** **Using the VIA's shift register to transmit a 40 usec ACK pulse.**

```
FORTH                                BASIC
HEX                                  1500 REM == TRANSMIT ACK SIGNAL
( ADDITIONAL VIA REGISTERS)
E84D CONSTANT IRQ-REG
E848 CONSTANT TIMER-LO
E849 CONSTANT TIMER-HI
E84A CONSTANT SHIFT-REG
DECIMAL
20 VARIABLE ACK-TIME ( 40 USEC)      1510 AT=20 : REM 40 USEC
HEX
: ACK!
   15 AUX-REG C!                     1520 POKE VI+11, 21  : REM AUX REG.
   80 SHIFT-REG C!                   1530 POKE VI+10, 128 : REM SHIFT REG.
   ACK-TIME C@  TIMER-LO C!          1540 POKE VI+8, AT : REM SET TIMER
   0 TIMER-HI C! ;                   1550 POKE VI+9,0 : REM START PULSE
```

***Listing 3:*** **BASIC and FORTH routines to await data from the digitizer.**

```
FORTH                                BASIC
HEX                                  2000 REM === WAIT FOR STROBE
( STROBE SETS BIT 1 OF )             2010 REM STROBE SETS BIT 1 OF
( IRQ FLAG REGISTER )                2020 REM IRQ FLAG REGISTER
                                     2030 REM
2 CONSTANT STROBE-FLAG               2040 SF=2
                                     2050 REM
: AWAIT
  BEGIN
   IRQ-REG C@ STROBE-FLAG AND        2060 A1 = PEEK (VI+13) AND SF
  UNTIL ;                            2070 IF A1=0 THEN GOTO 2060
                                     2080 RETURN
```

***Listing 4:*** **Three subroutines to read data from the digitizer. The first reads a single byte of a multiple-byte transmission. The second reads bytes from the digitizer, discarding all until it finds a control byte. The third uses the other two to read a complete 5-byte transmission.**

```
FORTH                                BASIC
HEX
( FETCH 1 BYTE )                     2500 REM ===FETCH 1 BYTE
: CPAD@ AWAIT                        2510 GOSUB 2000 : REM WAIT FOR STROBE
   DATA-REG C@                       2520 BY=PEEK(VI+1)  : REM READ DATA
   ACK! ;                            2530 GOSUB 1500 : REM TRANSMIT ACK
                                     2540 RETURN
( WAIT FOR CONTROL BYTE)
: SYNC BEGIN                         3000 REM === WAIT FOR CONTROL BYTE
   CPAD @                            3010 GOSUB 2500 : REM READ A BYTE
   80 AND UNTIL ;                    3020 IF (BY AND 128)=0 THEN GOTO 3010
                                     3030 RETURN
( READ 5-BYTE TRANSMISSION )         3500 REM ===READ 5-BYTE TRANSMISSION
: PAD@   SYNC                        3510 GOSUB 3000 : REM GET FIRST BYTE
  4 0 DO                             3520 FOR I=1 TO 4
    CPAD @                           3530 GOSUB 2500
  LOOP ;                             3540 BY(I)=BY
                                     3550 NEXT I
                                     3560 RETURN
```

***Listing 5:*** **The master digitizer control routine. Users invoke this routine each time they want to read data from the digitizer.**

```
FORTH                                BASIC
HEX
0 VARIABLE X
0 VARIABLE Y

: POINT@                             4000 REM === DIGITIZER DEVICE DRIVER
   VIA-SET                           4010 GOSUB 1000 : REM INITIALIZE
   PAD@                              4020 GOSUB 3500 : REM GET DATA
   SWAP 80 * + Y !                   4030 Y%=BY(3)*128+BY(4)
   SWAP 80 * + X ! ;                 4040 X%=128*BY(1)+BY(2)
                                     4050 RETURN
```

That is, the first (leftmost) item is multiplied by $2^7$, then added to the second (rightmost) item. This procedure, repeated for both coordinates, reduces the coordinates to standard 16-bit integers.

## Performance

When we compare the BASIC and FORTH routines in listings 1-5, we find the BASIC listings are not substantially easier to read or interpret. At times, it is true, the BASIC text conforms more closely to our expectations or to conventional notation, but elsewhere the FORTH text is clearer and more direct. The FORTH text is sometimes longer, since it explicitly declares constants, but these declarations substantially improve program clarity.

If all comments were removed, the BASIC program would be terribly difficult to understand. The FORTH program is fairly easy to interpret without comments. We know, for example, that VIA-SET must have something to do with setting up a VIA port, that ACK! must send or transmit something called ACK, and that AWAIT waits for something. GOSUB 1000, GOSUB 1500, and GOSUB 2000 are far less revealing!

Moreover, FORTH (unlike BASIC) does not discourage generous use of comments within the program. FORTH comments take no extra memory space, and don't require any execution time in the finished program.

The BASIC program can accept a point roughly every 150 milliseconds; the FORTH program can accept a point every ten milliseconds — the maximum speed at which the digitizer operates. The BASIC routines occupy 903 bytes; the FORTH routines occupy only 385 bytes. The FORTH routine is half the size of its BASIC equivalent, and runs almost 15 times faster!

## Using FORTH with Other Languages

This demonstration dramatically illustrates FORTH's superiority for controlling common devices. The FORTH code is not only faster and smaller, but it is probably easier to understand.

However, even this small example demonstrates FORTH's weakness; FORTH's Reverse Polish Notation can make algebraic expressions very obscure indeed. RPN can be very powerful; after all, many people prefer Hewlett Packard calculators. Nevertheless, few people would argue that

```
4 5 2 1 + * /.
```

is clearer or more easily understood than

$$\text{PRINT} \quad \frac{(\,(1 + 2) * 5\,)}{4} \qquad !$$

An excellent answer to this deficiency is to implement an algebraic language (such as BASIC or Pascal) in FORTH. A subset of Pascal is, in fact, already available. This compiler converts source text into FORTH-like object code, interpretable by FORTH's efficient inner interpreter. Since the language produces the same output as FORTH itself, FORTH and Pascal programs are free to call each other. Control and interface tasks can be handled in FORTH, while mathematical and algebraic problems can be handled using the more sophisticated Pascal parser.

For the coming decade or so, microcomputer users must continue to endure a shortage of computer power. Eventually, personal computers will be fast enough and have enough memory to exceed most people's needs, but that time is not here yet. For the present, small computers need efficient programming systems and languages with legible source code that produces fast, concise programs. While this computer power shortage endures, FORTH and its derivatives offer a promising solution.

---

FORTH for PET is available from

AB Computers
252 Bethlehem Pike
Comar, PA 18915

FSS
1903 Rio Grande
Austin, TX 78705

Microtech
P.O. Box 102
Langhorne, PA 19047

For more information on FORTH, see MICRO's FORTH Feature in the February, 1982 issue.

---

Contact the authors at the Department of Chemistry, Harvard University, 12 Oxford St., Cambridge, MA 02138.

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.

# MICRO Short Subjects

**Apple Disk Drive Repair** *by Patrick Schwab*
**Horizontally Scrolled Messages** *by Chris Williams*
**Breakpoint Utility for OSI C1P** *by John Seybold*

## Apple Disk Drive Repair — Do It Yourself?

Patrick Schwab, 805 Valleywood Dr.
S.E., Salem, Oregon 97306

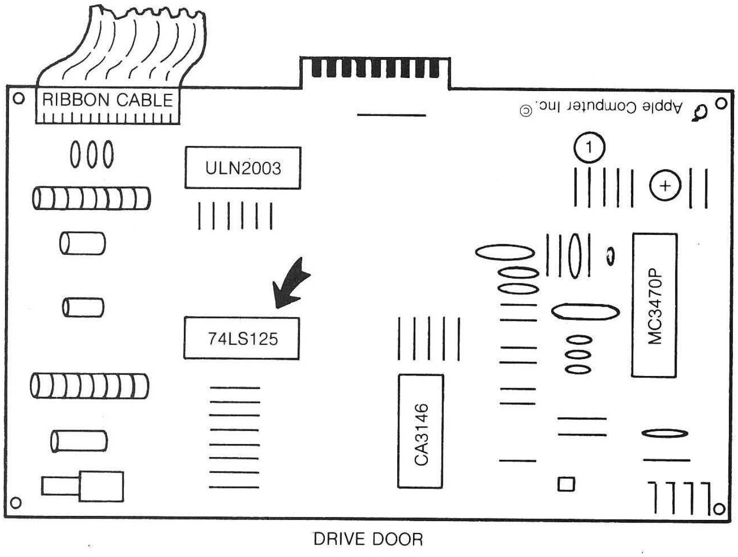
If you've ever plugged your disk drive in wrong, take heart. Except for the noise and smoke, there is remarkably little damage. In fact, if you took your drive into your Apple dealer he would only charge you the minimum service charge. What magic does he perform to resurrect a burnt out disk drive?

Actually, only one chip, the Tri-State Buffer, has been damaged and it's easy to get at. You can remove the IC with the IC puller from your 3.3 DOS upgrade kit. The damage to that chip can be extensive. The crater caused by the melting IC is reminiscent of Mt. St. Helens. Even if you do not see any apparent evidence, you can be assured your chip has suffered damage. For $1.50 from your Apple dealer, you can replace the 74LS125N.

### Replacing the 74LS125N

1. Make sure the computer is off.
2. Touch the power supply to discharge any static electricity.
3. Unplug the drive from the controller card.
4. Turn the drive upside down.
5. Remove the four black screws on the bottom of the drive.
6. Turn drive rightside up.
7. Slide the drive cover back and remove it.
8. Locate IC chip #74LS125N — refer to drawing below.
9. Use IC puller to lift the chip straight up.
10. Insert the new chip with writing as in drawing.
11. Slide the cover back over the drive.
12. Check the cable to make sure it is not pinched or twisted.
13. Turn the drive upside down.
14. Replace the screws.
15. Plug the drive back on the card. Make sure you do not off-set the plug to the second row of pins; if you do, go to step 1.

***Schwab Figure:***
**IC Chip 74LS125N is shown on circuit board in the left center of drawing.**



## Horizontally Scrolled Messages

Chris Williams, 5676 S. Meadow La.
#101, Ogden, Utah 84403

Horizontally scrolled ("ticker-tape") messages capture and hold the interest of onlookers. Any good marketing professional will tell you that getting the attention of a consumer is over half the job of selling.

Here is a program that will scroll a user-defined message across the screen at a user-defined rate. It's written in Applesoft and will run without crowding in a 16K machine. The total number of characters (including trailing periods) allowed in the message is 256 in each of the two DATA statements for a total of 512.

The two DATA statements at lines 45 and 55 are the source of the message and can be changed. Line 85 is the internal delay loop; changing its index's maximum value (currently 60) will vary the scrolling speed.

The heart of the program is in the rest of the "I" loop. Both the HTAB and MID$ commands print out the correct number of characters from the string at a decrementing horizontal cursor location, while incrementing the character number within the string where printing begins. These three simultaneous selections result in the right-to-left horizontal scrolling effect.

You can use the program as a whole, or strip out the "I" loop if you want to add it to another program. But I suggest you keep my convention of trailing periods at the end of each part of the message. I've found that without them you lose continuity and, worse, the watcher's interest.

**Williams Listing**

```
10  REM  *** ADVERTISE ***
20  REM  *** BY C WILLIAMS ***
30  HOME : VTAB 10: PRINT "THIS P
     ROGRAM WILL ALLOW YOU TO INP
     UT AN ADERTISEMENT AND HAVE
     IT SCROLL HORIZONTIALLY ACRO
     SS THE SCREEN AT VARYING RAT
     ES."
33  REM  THIS LOOP DELAYS TO KEEP

34  REM  PROGRAM DESCRIPTION ON T
     HE SCREEN FOR LONG ENOUGH TO
      READ
35  FOR Y = 1 TO 3000: NEXT Y: HOME

40  READ AD$
44  REM  DATA STATEMENT CONTAINS
     ADVERTISEMENT
45  DATA       "HELLO!  I AM AN A
     PPLE COMPUTER AND I CAN DO A
      WHOLE LOT MORE THAN JUST AD
     VERTISE MYSELF.  COME ON IN
     AND ASK ABOUT ME............
     ............................
     .. "
47  REM  70 SUBROUTINE CONTAINS S
     CROLLER
48  GOSUB 70
50  READ AD$
54  REM  DATA STATEMENT CONTAINS
     ADVERTISEMENT
55  DATA         "THE PEOPLE HER
     E WILL BE DELIGHTED TO SHOW
     YOU HOW I CAN HELP YOU.  AND
      REST ASSURED THAT I ALMOST
     CERTAINLY CAN...............
     .......................... "
57  REM  70 SUBROUTINE CONTAINS S
     CROLLER
58  GOSUB 70
59  REM  RESET READ STATEMENTS AN
     D LOOP BACK
60  RESTORE : GOTO 40
65  REM  SET LOOP INDEX = STRING
     LENGTH
70 L =  LEN (AD$)
75 II = 1
77  REM  VARIABLE SET FOR SPEED
78 FT = 40:UNO = 1:ZR = 0
80  FOR I = 1 TO L
84  REM  J LOOP CONTROLS SCROLL S
     PEED, CHANGE INDEX IF YOU WI
     SH
85  FOR J = 1 TO 60: NEXT J
87  REM  VTAB CENTERS SCROLLED "T
     ICKER-TAPE"
90  VTAB 14
95 TNUM = FT - I: IF TNUM < ZR THEN
     II = II + UNO
76  IF TNUM <  = ZR THEN TNUM = U
     NO
97  REM  TNUM CONTROLS HTAB WHICH
      IS HEART OF SCROLLER, WHEN
     < 0 MOVE STARTING POINT TO R
     IGHT IN STRING
100  HTAB TNUM
105 M = I: IF I > FT THEN M = FT
110  PRINT  MID$ (AD$,II,M)
115  REM  LOOP BACK FOR NEXT CHAR
     ACTER
120  NEXT I
124  REM  DONE
125  RETURN
```

**Seybold Listing**

```
                ;BREAKPOINT UTILITY
                ;
                ;BY JOHN SEYBOLD
                ;
                WARM     EQU $A274
                ;
                         ORG $1C0
                         OBJ $800
                ;
01C0 48                  PHA            ;SAVE ACCUMULATOR
01C1 8D 88 D3            STA !54152     ;PUT IT ON THE SCREEN
01C4 A9 41               LDA 'A
01C6 8D 86 D3            STA !54150     ;PRINT 'A' LABEL
01C9 A9 58               LDA 'X
01CB 8D 8D D3            STA !54157     ;PRINT 'X' LABEL
01CE A9 59               LDA 'Y
01D0 8D 94 D3            STA !54164     ;PRINT 'Y' LABEL
01D3 8E 8F D3            STX !54159     ;PUT X ON THE SCREEN
01D6 8C 96 D3            STY !54166     ;PUT Y ON THE SCREEN
01D9 20 00 FD            JSR $FD00      ;POLL KEYBOARD UNTIL A CHAR. R
EC'D
01DC C9 53               CMP 'S         ;IS IT A 'S'?
01DE D0 03               BNE RET        ;IF NOT, THEN RETURN
01E0 4C 74 A2            JMP WARM       ;OTHERWISE WARMSTART
01E3 68         RET      PLA
01E4 40                  RTI
```

**Seybold Figure**

```
A] X+  Y!
```

# Breakpoint Utility for OSI C1P

John S. Seybold, 2130 University Ave. #74, Madison, Wisconsin 53705

Recently, while debugging a user subroutine, I became aware of the value of a breakpoint utility. I was unable to read the contents of the registers and it was impossible to tell what the processor was doing. Since I could not afford an extended monitor, I wrote this short program to do the trick. The program makes use of the fact that the 6502 sees a BRK instruction as a software interrupt and jumps to the subroutine that has its address at $FFFE and $FFFF. In the C1P and Superboard, that address is $1C0.

If you are familiar with the C1P's memory map, you will have noticed that this address is at the top of the BASIC stack area. If you call your machine code program from a BASIC program that uses a large amount of stack, then you will have to relocate the utility. This poses no problem since the code is entirely relocatable, but you must key in a JMP instruction at $1C0. For example, if you want to put the utility in page two, simply key in $4C 22 02 into locations $1C0 through $1C2 and then put the utility in, starting at $222. (Beware of overwriting your user routine if it is also in page two.)

To use the utility, just add "BRK NOP" to your program wherever desired. The NOP is to allow for the fact that the 6502 saves the contents of the program counter, plus two, when it executes a break instruction which is only one byte long. Once the breakpoints have been added and the utility is in memory, you may run your program.

When a breakpoint is encountered, you will see a display similar to the one shown in the figure. A, X, and Y stand for the respective registers. The figure next to each letter is the graphics character of the code in that register. To resume execution of your program, simply hit any key on the keyboard except "S". Hitting "S" will stop the program completely and warm start the machine to allow you to check memory locations. Once you have stopped, you must run the program from the beginning again, but when you get to the same breakpoint, you will know what is in each memory location, if you haven't made any changes.

The program is heavily commented so as to be self-explanatory. I hope that it will be as much help to you as it was to me.

MICRO

**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 froms. **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.**

---

CSE means OSI

Custom After Market Software
for C1P and C4P machines

*Basic Enhancer:
Renumber, Auto Sequencer, Screen Control functions, and tape 1/0 system that is faster and has file names
C1P .................... $21.95
C4P .................... $29.95

Modified Monitor Rom Chip:
Now get indirect jump-capabilities just like those in the C1P and for no extra charge CSE will burn in your machines serial number ............... $16.95

*NOTE: The C4P version of the Basic Enhancer includes the modified monitor Rom chip required for proper program functioning.

This is only a partial listing of our products. Write us for information on new disk programs or send $2 for catalog. Please include $2.00 shipping and handling with orders.

**Computer**
**Science**
**Engineering**
Box 50 • 291 Huntington Ave. Boston 02115

---

**BREAK THE TIME BARRIER!**

**Speed-up and expand Apple ][ 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 ][ or ][+, 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.*

Box 5 Manvel ND 58256
(701) 696-2574

**S&H Software**

## 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...$380** **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** **$180**
OSI-Type floppy disk controller and real time clock. Will Support 5¼" or 8", Single or double-sided drives. Requires drives with separated data and clock outputs.

**BIO-1600 BARE IO CARD** **$50**
Super 1/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.

**BMEM-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. 1/16" epoxy glass P.C. board.

**C1P-EXP EXPANSION INTERFACE** **$65**
Expansion for C1P 600 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%.

## APPLE-LISP

**Excellent implementation of LISP 1.5**
**for the Apple II Computer!**

Apex Company Limited introduces APPLE-LISP which is designed for symbolic data processing and used for the fields of artificial intelligence.

Features of APPLE-LISP are as follows:

- APPLE-LISP has *three systems.*
  - — ST LISP (accepts S-expression programs)
  - — META LISP (accepts both S-expression programs and meta-like language programs)
  - — MINI ED (edits the source program)
- APPLE-LISP accepts not only S-expression programs but also meta-like language programs.
- *142* machine language functions are available in disk version.
- *Fast* and *efficient* Garbage Collector.
- Full support of *low resolution* and *high resolution* graphics (gr, plot, hlin, vlin, hplot, hcolor, etc.).
- Full support of *real* number and *arithmetic functions* (sin, cos, log, rnd, abs, quotient, sqrt, sign, plus, max, etc.).
- *Tracing* by trace functions and interruption.
- Runs on 48K Apple II or Apple II Plus with *disk* or *cassette.*
- Extensive 140-page user's manual.
- Includes *PROG* feature, *WHILE* structure, extended COND expression.
- Many *library* functions.
- The following attractive programs are included:
  EXPRES (converts infixed expression to prefixed),
  HANOI (tower of HANOI),
  DET (calculates determinant of the matrix with symbolic formula),
  HILBERT (draws two recursive graphic patterns), etc.

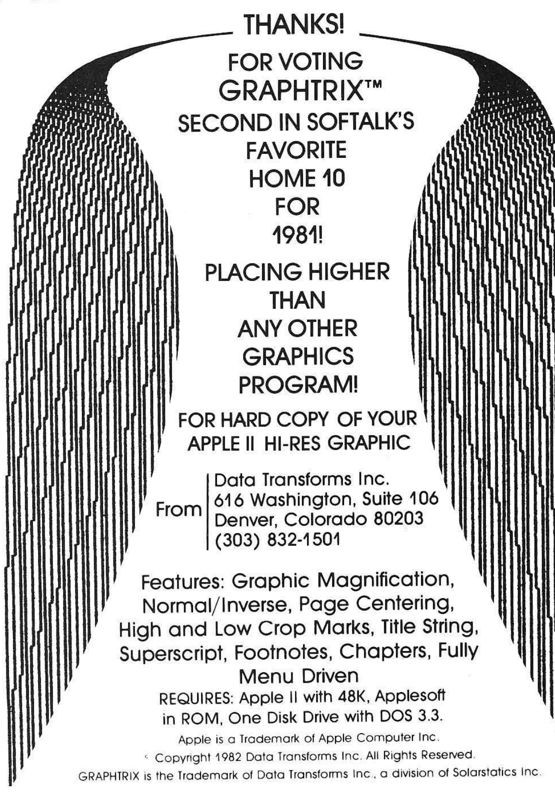
| | |
|---|---|
| (A) APPLE-LISP Disk Version with DOS 3.3 | $252.00 |
| (B) APPLE-LISP Disk Version without DOS 3.3 | $192.00 |
| (C) APPLE-LISP Cassette Version | $189.00 |

*Additional $6.00 for postage fee required.*

To Order: Send the order form with your check or money order.

To Apex Company Limited Date: ____________
I enclose my check/money order $________ for ____________
unit(s) of item ______________________________
Name ______________________________
Address ______________________________

**Apex Company Limited**
5-11-3, Yagumo, Meguro-Ku, Tokyo, 152 Japan


THANKS!
FOR VOTING
GRAPHTRIX™
SECOND IN SOFTALK'S
FAVORITE
HOME 10
FOR
1981!

PLACING HIGHER
THAN
ANY OTHER
GRAPHICS
PROGRAM!

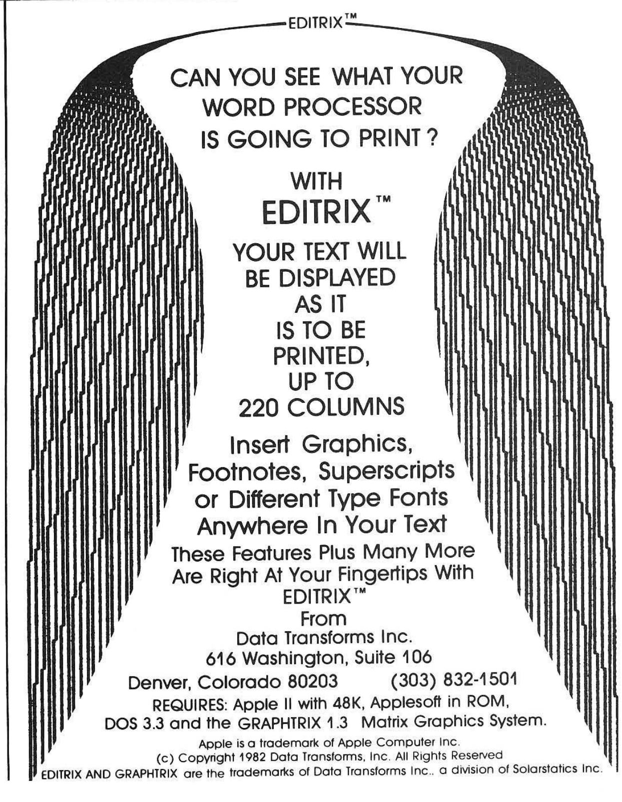
FOR HARD COPY OF YOUR
APPLE II HI-RES GRAPHIC

From
Data Transforms Inc.
616 Washington, Suite 106
Denver, Colorado 80203
(303) 832-1501

Features: Graphic Magnification, Normal/Inverse, Page Centering, High and Low Crop Marks, Title String, Superscript, Footnotes, Chapters, Fully Menu Driven

REQUIRES: Apple II with 48K, Applesoft in ROM, One Disk Drive with DOS 3.3.

Apple is a Trademark of Apple Computer Inc.
© Copyright 1982 Data Transforms Inc. All Rights Reserved.
GRAPHTRIX is the Trademark of Data Transforms Inc., a division of Solarstatics Inc.


EDITRIX™

CAN YOU SEE WHAT YOUR
WORD PROCESSOR
IS GOING TO PRINT?

WITH
EDITRIX™
YOUR TEXT WILL
BE DISPLAYED
AS IT
IS TO BE
PRINTED,
UP TO
220 COLUMNS

Insert Graphics, Footnotes, Superscripts or Different Type Fonts Anywhere In Your Text

These Features Plus Many More Are Right At Your Fingertips With EDITRIX™

From
Data Transforms Inc.
616 Washington, Suite 106
Denver, Colorado 80203 (303) 832-1501

REQUIRES: Apple II with 48K, Applesoft in ROM, DOS 3.3 and the GRAPHTRIX 1.3 Matrix Graphics System.

Apple is a trademark of Apple Computer Inc.
(c) Copyright 1982 Data Transforms, Inc. All Rights Reserved
EDITRIX AND GRAPHTRIX are the trademarks of Data Transforms Inc., a division of Solarstatics Inc.

# PRINT AT for OSI Systems

*by Matt Asay*

**By loading this routine, the AT keyword is recognized in a PRINT statement. The routine is designed for an OSI C1P but should work (with the modifications suggested) on a C4P or C8P. The method used here could probably be adapted to add recognition of new statements or keywords to other Microsoft BASICs such as the PET. A useful hex object code loader is also provided.**

**PRINT AT**
requires:
OSI C1P
May be modified easily for C2P.

The Microsoft BASIC on an Ohio Scientific C1P has most of the features found on other versions. One feature that (unfortunately) is lacking is the ability to print at a selected location on the screen. There are some ways to get around this by using POKE but you are limited to POKEing one character at a time. It is slow and cumbersome to do.

I have developed a program to remove these limitations by adding an AT option to the PRINT statement. Once this program is installed you can print anything anywhere on the screen with ease. The program hides itself at the top of your available memory on any size system, using only 166 bytes of permanent storage. After it has been entered you can write, save, load, and run programs using the new PRINT AT statement. Programs which do not use AT in their PRINTs should function the same as always.

The three forms of the statement are:

PRINT AT *location; print-list;*

where *location* is:

**BASIC Program to Load and Initialize PRINT AT**

```
(See text for description of the Relative Hexadecimal Loader)

    1 REM   ((PRINT AT))
    2 REM   BY MATT ASAY
    3 REM   ((REL HEX LOADER))
    5 GOSUB 10: GOTO 1000
   10 DEF FNA(DX)=ASC(MID$(H$,DX,1))
   20 DEF FNHD(DX)=FNA(DX)-48+(FNA(DX)>64)*7
   30 DEF FNB(DX)=FNHD(DX)*16+FNHD(DX+1)
   40 DEF FNHA(DX)=((FNHD(DX)*16+FNHD(DX+1))*16+FNHD(DX+2))*16+FNHD(DX+3)
   45 READ H$: RO=PEEK(134)*256+PEEK(133)-FNHA(1)
   50 FOR HA=RO TO 32767: READ H$: ON LEN(H$) GOTO 51,52,53,54,55: GOTO 54
   51 RETURN
   52 POKE HA,FNB(1): NEXT: STOP
   53 RA=RO+FNB(2): GOTO 56
   54 POKE HA,FNB(1): FOR I=3 TO LEN(H$) STEP 2: HA=HA+1: POKE HA,FNB(I):
   NEXT I: NEXT: STOP
   55 RA=RO+FNHA(2)
   56 IF LEFT$(H$,1)="H" THEN POKE HA,RA/256: NEXT: STOP
   57 POKE HA,RA AND 255: IF LEFT$(H$,1)="R" THEN HA=HA+1: POKE HA,RA/256
   58 NEXT: STOP

Size of code in hex
  100 DATA 00FD
Code for USRX
  110 DATA A9,L57,A0,H57,8581848285838484858584 86A207
  120 DATA BD,R4F,95C5CA10F8AD1A02AC1B028D,RE6,8C,RE7
  130 DATA A9,LE1,A0,HE1,8D1A028C1B02AD1C02AC1D02
  140 DATA 8D,RFB,8C,RFC,A9,LF6,A0,HF6,8D1C028C1D02A988A0AE
  150 DATA 850B840C60C920F0F34C,R57,00
Code for PSPLIC
  160 DATA 24CC1014C941D00E489848A001B1C3C954F013
  170 DATA 68A86806CCC997D00285CCC93AB0034CCD0060
Code for PR.AT
  180 DATA 46CC68A86820BC0020BC00C9A5D00620BC0038
  190 DATA B04120C1AA2008B420C200C92CD023A5110A0A
  200 DATA 0A0A26120A8511A5122A48A5114820C9AA2008B4
  210 DATA 6818651185116865128512A5118D,RE9,A5122903
  220 DATA 09D08D,REA,20C200C93BD0034CBC00A91C85CC4C4EA2
Code for OSPLIC
  230 DATA 24CC70034C00008D00D0EE,RE9,D003EE,REA,C60E60
Code for CSPLIC
  240 DATA A90085CC4C0000
End of load data marker
  250 DATA *

This code initializes PRINT AT while preserving any previous USR function
 1000 UL=PEEK(11)
 1010 UH=PEEK(12)
 1020 POKE 11,RO-INT(RO/256)*256
 1030 POKE 12,RO/256
 1040 X=USR(X)
 1050 POKE 11,UL
 1060 POKE 12,UH
The following code is a short demonstration of the use of PRINT AT
 1200 PRINT: PRINT
 1210 PRINT "TEST PROGRAM"
 1220 FOR I=1 TO 20: PRINT: NEXT
 1230 PRINT AT 10*32+5;"PRINT";
 1240 PRINT AT *;" AT";
 1250 PRINT AT 12,5;"HAS BEEN";
 1260 PRINT "WORKS !!!";
 1270 PRINT AT *;" LOADED...";
 1280 A$="AND IT"
 1290 PRINT AT 14,20-LEN(A$);A$;
 1300 FOR I=1 TO 500: NEXT
```

1. A numeric expression. Printing starts at sc + INT(expression), where sc is the address of the screen.
2. Two numeric expressions separated by a comma. Printing starts at sc + INT(expr1) * 32 + INT (expr2). This allows specification of location by row and column.
3. An asterisk ("*"). Printing continues with the position immediately after the last character printed by the last PRINT AT.

*print-list* is any allowable list of items to be printed, separated, by semicolons. The trailing semicolon is necessary since the carriage return and linefeed that BASIC tags on will print as their corresponding graphics characters. This was done intentionally to allow the printing of all graphics characters using CHR$( ).

### Examples

PRINT AT 200;CHR$(248);" < - A tank";

PRINT AT X,Y; "PRINT AT ROW X, COLUMN Y";

PRINT AT 15,7; "PRINT AND ";

A$ = "ADD"

PRINT AT *; A$ + " MORE";

PRINT "PRINT ON BOTTOM AND SCROLL"

### How to Install

Once I had developed this program I needed an easy way to install it on a system. I considered and rejected making a tape that the monitor could read. It would be difficult to modify, error-prone on input, and would work only if loading to a fixed absolute address. I did not want to use a BASIC program that POKEd in several DATA statements of decimal values since I think in hex when programming in assembly. For this reason I created a BASIC program that reads hex strings and converts them to binary and loads them into memory. To make it adaptable it calculates a starting load address from the size of the program and the address of the top of memory.

Enter the program shown on listing 1, save it to tape and then run it. After it is through loading (about 15 seconds) it will print "PRINT AT HAS BEEN LOADED... AND IT WORKS !!!"

### Assembly Listing of PRINT AT Routine

```
Assembly listing of PRINT AT routines
(Underlined values entered as relative addresses in DATA statements)

                   *       =     $2100
2100               USRX    =     *               INITIALIZATION VIA X=USR(X)
2100  A9 57                LDA   #PS.LB          RESERVE MEMORY FOR SPLICES
2102  A0 21                LDY   #PS.UB
2104  85 81                STA   $81
2106  84 82                STY   $82
2108  85 83                STA   $83
210A  84 84                STY   $84
210C  85 85                STA   $85
210E  84 86                STY   $86
2110  A2 07                LDX   #7              PUT SPLICE INTO PARSE ROUTINE
2112  BD 4F 21     USRX1   LDA   PATCH,X
2115  95 C5                STA   $C5,X
2117  CA                   DEX
2118  10 F8                BPL   USRX1
211A  AD 1A 02             LDA   $021A           GET OLD OUTPUT VECTOR
211D  AC 1B 02             LDY   $021B
2120  8D E6 21             STA   OS.0+1          STORE INTO OUTPUT SPLICE
2123  8C E7 21             STY   OS.0+2
2126  A9 E1                LDA   #OS.LB          SPLICE INTO OUTPUT VECTOR
2128  A0 21                LDA   #OS.UB
212A  8D 1A 02             STA   $021A
212D  8C 1B 02             STY   $021B
2130  AD 1C 02             LDA   $021C           GET OLD CONTROL-C VECTOR
2133  AC 1D 02             LDY   $021D
2136  8D FB 21             STA   CS.0+1          STORE INTO CONTROL-C SPLICE
2139  8C FC 21             STY   CS.0+2
213C  A9 F6                LDA   #CS.LB          SPLICE INTO CONTROL-C VECTOR
213E  A0 21                LDY   #CS.UB
2140  8D 1C 02             STA   $021C
2143  8C 1D 02             STY   $021D
2146  A9 88                LDA   #$88            RESTORE USR VECTOR TO DEFAULT
2148  A0 AE                LDY   #$AE
214A  85 0B                STA   $0B
214C  84 0C                STY   $0C
214E  60                   RTS
                   ;
214F  C9 20        PATCH   CMP   #'              PATCH PUT AT $C5-$CC
2151  F0 F3                BEQ   *-11
2153  4C 57 21             JMP   PSPLIC
00CC               ATFLG   =     *-PATCH+$C5     AT FLAG AT $CC
                   ; BIT 0 SET MEANS PRINT TOKEN FOUND ON LAST FETCH
                   ; BIT 1 SET MEANS PRINT AT CURRENTLY ACTIVE
2156  00                   .BYTE *-*
                   ;
2100               T       =     PSPLIC/256*256
0057               PS.LB   =     PSPLIC-T        PSPLIC LOW BYTE
0021               PS.UB   =     PSPLIC/256             UPPER BYTE
2100               T       =     OSPLIC/256*256
00E1               OS.LB   =     OSPLIC-T        OSPLIC LOW BYTE
0021               OS.UB   =     OSPLIC/256             UPPER BYTE
2100               T       =     CSPLIC/256*256
00F6               CS.LB   =     CSPLIC-T        CSPLIC LOW BYTE
0021               CS.UB   =     CSPLIC/256             UPPER BYTE
                   ;
2157               PSPLIC  =     *               PARSER SPLICE
2157  24 CC                BIT   ATFLG           PRINT TOKEN FOUND ?
2159  10 14                BPL   SPL1            BRANCH IF NOT
215B  C9 41                CMP   #'A             CHECK FOR "AT"
215D  D0 0E                BNE   SPL0            NO
215F  48                   PHA                   SAVE A,Y
2160  98                   TYA
2161  48                   PHA
2162  A0 01                LDY   #1
2164  B1 C3                LDA   ($C3),Y
2166  C9 54                CMP   #'T             (NO BLANKS ALLWD BETWEEN A AND T)
2168  F0 13                BEQ   PR.AT           BRANCH IF "AT" FOUND
216A  68                   PLA                   RESTORE Y,A
216B  A8                   TAY
216C  68                   PLA
216D  06 CC        SPL0    ASL   ATFLG           CLEAR "PRINT FOUND" BIT
216F  C9 97        SPL1    CMP   #PRTOK          IS CHARACTER PRINT TOKEN ?
2171  D0 02                BNE   SPL2            BRANCH NO
2173  85 CC                STA   ATFLG           SET PRINT FOUND, CLEAR AT FOUND
2175  C9 3A        SPL2    CMP   #':             SET STATUS AND RETURN CHARACTER
2177  B0 03                BCS   SPL3
2179  4C CD 00             JMP   $00CD
217C  60           SPL3    RTS
                   ;
217D               PR.AT   =     *               "PRINT AT" FOUND
217D  46 CC                LSR   ATFLG           CLEAR PRINT FLAG, SET AT FLAG
217F  68                   PLA                   RESTORE Y,A
2180  A8                   TAY
2181  68                   PLA
2182  20 BC 00             JSR   $00BC           SKIP OVER "T"
2185  20 BC 00             JSR   $00BC           GET NEXT CHARACTER
2188  C9 A5                CMP   #ASTTOK         "*" TOKEN ?
218A  D0 06                BNE   PR.A0           BRANCH IF NOT
218C  20 BC 00             JSR   $00BC           GET NEXT CHAR
218F  38                   SEC
```

**Assembly Listing** *(continued)*

```
2190  B0 41             BCS   PR.A3        BRANCH ALWAYS
2192  20 C1 AA   PR.A0  JSR   $AAC1        COLLECT EXPRESSION 1
2195  20 08 B4          JSR   $B408        CONVERT TO INTEGER
2198  20 C2 00          JSR   $00C2        FOLLOWED BY "," ?
219B  C9 2C             CMP   #',
219D  D0 23             BNE   PR.A2        BRANCH NO
219F  A5 11             LDA   $11          PUSH INT(EXPR1)*32 ONTO STACK
21A1  0A                ASL   A
21A2  0A                ASL   A
21A3  0A                ASL   A
21A4  0A                ASL   A
21A5  26 12             ROL   $12
21A7  0A                ASL   A
21A8  85 11             STA   $11
21AA  A5 12             LDA   $12
21AC  2A                ROL   A
21AD  48                PHA
21AE  A5 11             LDA   $11
21B0  48                PHA
21B1  20 C9 AA          JSR   $AAC9        COLLECT 2ND EXPRESSION
21B4  20 08 B4          JSR   $B408        CONVERT TO INTEGER
21B7  68                PLA                ADD INT(EXPR1)*32
21B8  18                CLC
21B9  65 11             ADC   $11
21BB  85 11             STA   $11
21BD  68                PLA
21BE  65 12             ADC   $12
21C0  85 12             STA   $12
21C2  A5 11      PR.A2  LDA   $11          ADD $D000, STORE AS "AT" ADDRESS
21C4  8D E9 21          STA   ATLO
21C7  A5 12             LDA   $12
21C9  29 03             AND   #$03
21CB  09 D0             ORA   #$D0
21CD  8D EA 21          STA   ATHI
21D0  20 C2 00          JSR   $00C2        GET NEXT CHARACTER
21D3  C9 3B      PR.A3  CMP   #';          MUST BE ";"
21D5  D0 03             BNE   B00B00       BRANCH IF NOT
21D7  4C BC 00          JMP   $00BC        GET NEXT CHAR AND RETURN TO
                 ;                            PRINT ROUTINE

21DA  A9 1C      B00B00 LDA   #28          LOAD OFFSET OF "ST" ERR MSG
21DC  85 CC             STA   ATFLG        RESET PRINT AND AT FLAGS
21CE  4C 4E A2          JMP   $A24E        PRINT ERROR MESSAGE
                 ;
0097             PRTOK  =     $97          BASIC TOKEN FOR "PRINT"
00A5             ASTOK  =     $A5          BASIC TOKEN FOR "*"
                 ;
                 OSPLIC =     *            OUTPUT VECTOR SPLICE
21E1  24 CC             BIT   ATFLG        "AT FLAG" SET ?
21E3  70 03             BVS   OS.1         BRANCH IF YES
21E5  4C 00 00   OS.0   JMP   *-*          DO NORMAL OUTPUT AND RETURN
21E8  8D 00 D0   OS.1   STA   $D000        STORE CHARATER ON SCREEN
21E9             AT.LO  =     *-2          LOCATION TO STORE IN (LOW BYTE)
21EA             AT.HI  =     *-1                              (HIGH BYTE)
21EB  EE E9 21          INC   AT.LO        INCREMENT SCREEN ADDRESS
21EE  D0 03             BNE   OS.2
21F0  EE EA 21          INC   AT.HI
21F3  C6 0E      OS.2   DEC   $0E          DON'T LET CHAR COUNT OVERFLOW
21F5  60                RTS                RETURN
                 ;
21F6             CSPLIC =     *            CONTROL-C VECTOR SPLICE
21F6  A9 00             LDA   #0           END OF STATEMENT,
21F8  85 CC             STA   ATFLG          SO RESET PRINT, AT FLAGS
21FA  4C 00 00   CS.0   JMP   *-*          DO NORMAL CONTROL-C STUFF
21FD                    END

        Addresses and Subroutines used in PRINT AT
$000B Address of USR subroutine vector
$000E Current "characters printed" count.  Incremented by BASIC before
      output routine is called.
$0011 Integer part (low,high form) of number in floating point accumulator
      ($00AC-$00AF) after a call to "fix" routine  at $B408.
$0081 Address of start of string storage    Set to start of PSPLIC by USR
$0083 Address of string scratchpad          call (initialization) to protect
$0085 Address of end of memory+1            PRINT AT routine.
$00BC Increment current character pointer and fetch next character of BASIC
      program.  Return with Z set if end of line or ":", C clear if "0"-"9",
      blanks skipped over.
$00C2 Fetch current character, setting status as above.
$00C3 Address of current character.
$021A Address of print routine.
$021C Address of routine to check for control-C. (Called at the end of each
      BASIC statement).
$A24E BASIC error message routine.  Use A as offset into error message table.
$AAC1 Collect expression starting at current character of program. Put result
      into floating point accumulator ($AC-$AF).
$AAC9 Collect expression starting with next characterof program.
$AE88 Default address for USR routine. Prints "FC" error message.
$B408 Converts number in floating accumulator to integer and stores it at
      $11,$12 (low,high form).
$D000 Address of start (upper left corner) of screen.
```

across several lines of the screen. Then you may type NEW and enter or LOAD any program you like, using PRINT AT.

If an error occurs in the middle of a PRINT AT statement the "AT flag" can be turned off by typing any valid BASIC statement (i.e., LIST or "?" for PRINT, etc.) at the keyboard.

### Relative Hexadecimal Loader

The loader reads strings from data statements and loads a program into high memory. The program consists of four parts:

1. Program size:

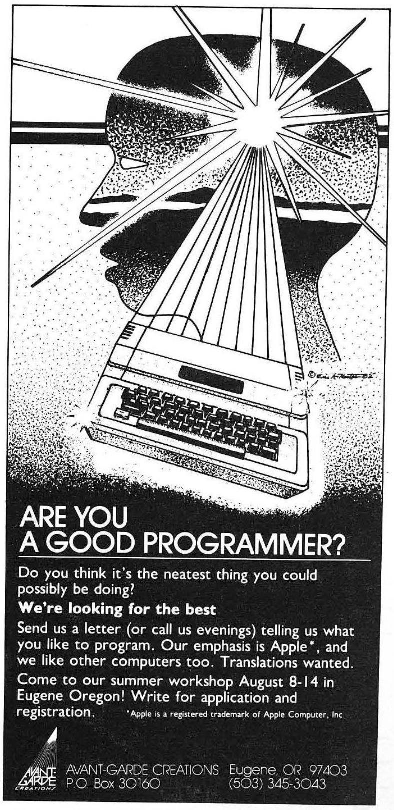
   A 4-digit hex number. This value is subtracted from the end-of-memory address at $0085 to get the starting address for the program.

2. Non-relocatable hex data:

   A string of any number of bytes in hex form.

3. Relocatable addresses:

   A prefix character R, H, or L followed by two or four hex characters. The hex number is added to the starting



address of the program. The resulting address is stored as follows:

R: Store both bytes (low, high form).
H: Store high byte.
L: Store low byte.

4. End of program marker:

   Any single character ("*" is used here).

You can use the loader program for your own machine language routines. Use lines 1-58 as shown. Replace 100-999 with DATA statements for your code in the format shown. When the program has finished loading it will jump to 1000 with R0 set to the starting load address. Your statements here should protect your program if desired and perform any other initialization needed.

## How the Program Works

The program has four parts: a USR call for initialization, "splices" into the BASIC parse, output, and control-C routines.

The USR routine changes the top of memory address to protect the permanent part of the program (not including this initialization). It patches the other three pieces into their respective vectors. The code at line 1000 saves and restores the previous USR address, so this routine can be loaded after another USR routine without messing it up.

The second piece is spliced into the parse routine at $BC-$D3. This routine fetches the program for the BASIC interpreter a character/token at a time. When not in a PRINT statement this routine works normally. Otherwise it checks for an AT following the PRINT token. If it is found, the routine collects and interprets the location specification. It then returns the character following the first semicolon to the print routine as if the "AT location;" had not been there.

The third piece is spliced into the output vector. Any time the "AT flag" (bit 1 of $CC) is on, instead of going to the normal print routine, it outputs to the current screen location and then increments the location. It then decrements the character count (which the routine that calls it increments) to prevent overflow and returns to the caller.

The last piece is spliced into the control-C vector. This vector is called at the end of each statement (to check if control-C is depressed). The spliced routine unconditionally resets the "AT flag" before going to the normal control-C routine. This prevents an error, control-C, or END of the program from leaving the "PRINT AT" on when control returns to the user.

This program takes 253 bytes to load, but after initialization requires only 166 bytes. If you wish to preserve the initialization code also just change the "L57" in line 110 to "L00".

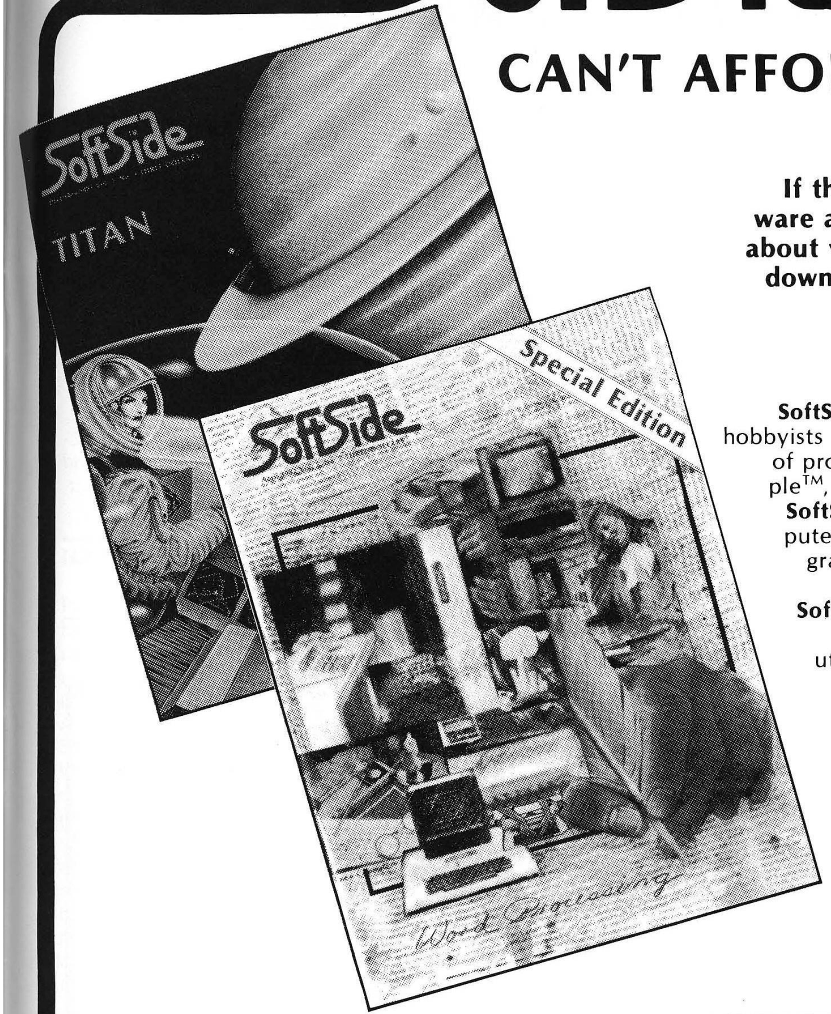
The only change which should be necessary to use this routine on a C2P is to change the code at 219F-21B0 to multiply by 64 instead of 32.

---

Matt Asay is a senior analyst at Queue Systems Incorporated where he develops process control and data acquisition systems. He holds a degree in Computer Science from CSU, Sacramento. His home computer is an OSI C1P with 20K RAM and a floppy disk. He may be contacted at 2925 Janet Drive, West Sacramento, CA 95691.

MICRO™

# 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.

Use coupon to order. Mail to: **SoftSide** Publications, 6 South St., Milford, NH 03055

**As you can see, you'll receive pages and pages of information and entertainment from SoftSide. 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 Product Catalogs

Our Software and Hardware Catalogs list the newest 6502/6809-based products. (Turn to pages 109 and 117.) The easy-to-read entries offer quick information for our readers. Use these catalogs to present your products. Just write to MICRO and ask for the appropriate form.

**Software (Hardware) Catalog**
**MICRO**
**P.O. Box 6502**
**Chelmsford, MA 01824**

## MICRObits *(continued)*

**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

**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

*Deadline for MICRObits: 20th of second month before publication. Send typewritten copy with $25 per insertion. (Subscribers: first ad at $10.)*

MICRO™

SIGNALMAN MARK I DIRECT CONNECT MODEM – $89.50

Standard 300-baud, full duplex, answer/originate. Powered by long lasting 9-volt battery (not included). Cable and RS-232 connector included.

EPROMS – HIGH QUALITY, NOT JUNK

Use with PET, APPLE, ATARI, SYM, AIM, etc. 450 ns. $6.50 for 2716, $12.50 for 2532. We sell EPROM programmers for PET and ATARI

5¼ INCH SOFT SECTORED DISKETTES

Highest quality. We use them on our PETs, APPLEs, ATARIs, and other computers. $22.50/10 or $44.50/20

NEW! C. ITOH STARWRITER F-10 DAISY WHEEL PRINTER

Letter quality, flawless copy at 40 char/sec. Bidirectional printing, 15-inch carriage, uses standard Diablo ribbons and print wheels.

PARALLEL – $1495, RS-232 – $1680, TRACTORS – $210

MAE SOFTWARE DEVELOPMENT SYSTEM FOR PET, APPLE, ATARI $169.95

"The Compatible Assembler"

- Professional system for development of Machine Language Programs. 31 Characters per label.
- Macro Assembler/Text Editor for Disk-based systems.
- Includes Word Processor for preparation of Manuals, etc.
- Standard Mnemonics – Ex.: LDA (LABEL), Y
- Conditional Assembly, Interactive Assembly.
- Editor has string search/search and replace, auto line numbering, move, copy, delete, uc/lc capability.
- Relocating Loader to relocate object modules.
- Designed with Human Factors Considerations.

FLASH!! EHS Management has decided to allow $50.00 credit to ASM/TED owners who want to upgrade to MAE. To get this credit, return ASM/TED manual with order for MAE.

BEFORE YOU BUY THAT OFF-BRAND ASSEMBLER, WRITE FOR OUR FREE DETAILED SPEC SHEET.

SMARTERM 80 COLUMN CARD FOR APPLE – $279

Upper/lower case and 80 columns. Includes 5x7 matrix character set, full ASCII keyboard, and true shift key operation.

TYMAC PARALLEL PRINTER INTERFACE FOR APPLE – $119.95

For use with Centronics, Starwriter, Prowriter, etc.

TRAP65 – $69.95 (was $149.95)

Traps 6502 unimplemented opcodes and helps prevents accidental system crashes. Also useful to extend the 6502's instruction set. Imagine a DIVIDE or MULTIPLY instruction, or PHX, PHY, etc. Manual describes how you can add new instructions.

EPROM PROGRAMMER FOR PET AND ATARI COMPUTERS

The BRANDING IRON is an EPROM programmer especially designed for PET and ATARI computers. Programs 2716 and 2532 type EPROMs. The PET version plugs into the cassette and I/O port and comes with software which adds the programmer commands to the PET monitor. The ATARI version plugs into controller jacks and comes with a full fledged machine language monitor which provides 30 commands for interacting with the computer and the BRANDING IRON. PET – $75.00 ATARI – $119.95

C. ITOH PROWRITER DOT MATRIX

Near letter quality printing at 120 char/sec with tractor and friction feed and graphics capabilities (some say it is better than the MX-80).

PARALLEL – $499.95 , RS-232 – $660.00

Eastern House

3239 Linda Dr.
Winston-Salem, N.C. 27106
(919) 924-2889 (919) 748-8446
Send for free catalog!

VISA® MasterCard

# Symbol Table Lister for the OSI

### by Rolf Johannssen

**Use this routine to list the symbol table generated by the OSI C1P Assembler.**

**Symbol Table Lister**
requires:

OSI C1P
OSI C1P Assembler

## Introduction

Programming in assembly rather than a high level language (BASIC, Pascal) may be preferred for one of three main reasons: speed, economy of memory usage, and the ability to do things not available in the higher level languages. Small sections of code can be assembled by hand and entered using a simple monitor. However, this is a tedious process and prone to error. For any serious assembly language coding an assembler program must be used. An assembler will read source code, check for errors, generate all necessary cross-references, and produce the desired assembled code. A listing may optionally be produced by the assembler.

## The OSI C1P Assembler-Editor

The OSI C1P assembler does all of these things and has editing capability as well so that the user can conveniently enter source code from keyboard or tape and edit it before assembly. One desirable option lacking in the OSI assembler is the ability to list or print out a *symbol table* following the listing. This is a table listing all symbols and labels together with their assigned values. A symbol table is a valuable adjunct in reading a program listing. When modifying a program, it helps you avoid inadvertent duplication of symbols or labels. A complete cross-reference program would be even more useful, and would not be difficult to write. For my own use, the extra effort and extra memory required did not

**Symbol Table Lister**

```
                 ;      SYMBOL TABLE LISTING PROGRAM
                 ;      LAST REVISION 14 JAN 82
                 ;      PAGE ZERO LOCATIONS
                 LL   = $0A         LAST LINE USED IN SYMBOL TABLE
                 BC   = $10         BYTE COUNTER
                 CC   = BC+1        CHAR. COUNTER
                 CSV  = CC+1        SAVED CHARACTER
                 MCTR= CSV+1        MULT. CHAR. COUNTER
                 XP   = MCTR+1       X POINTER
                 XSV  = XP+1        X REG. SAVE
                 YSV  = XSV+1       Y REG. SAVE
                 LN   = YSV+1       LINE NUMBER
                 LW   = LN+2        LAST WORD
                 PTR  = LW+2        POINTER
                 PTR2= PTR+2        SECOND POINTER
                 BFR  = PTR2+2      BUFFER
                 DEST= BFR+8        DESTINATION BUFFER
                 M    = DEST+8      MINIMUM SYMBOL VALUE
                 MP   = M+4         MINIMUM IN CURRENT LOOP
                 CTRO= $64          CTRL/O FLAG AS IN BASIC
                 ;      ADDRESS EQUATES
                 BCB  = $0100       BASIC ASCII BUFFER
                 PFL  = $0205       SAVE-TO-TAPE FLAG
                 STMEM=$12C9        START OF MEM FOR SOURCE
                 STS  = $12CB       TOP OF STORAGE
                 NL   = $12FE       NEXT LOCN FOR SOURCE
                 CRL  = $A86C       BASIC RETURN-LINE FEED
                 BFF  = $B7E8       BASIC HEX-DEC CONVERSION
                 BPF  = $B96E       BASIC DEC-ASCII CONVER
                 EXM  = $E800       EXTENDED MONITOR
                 PHEX= $EAAC        PRINT HEX CHAR.
                 DVD  = $EE6C       16-BIT DIVIDE ROUTINE
                 PRINT=$FFEE        PRINT VECTOR
                 ;      PROGRAM STARTS HERE
350 1391                *=$1391
360 1391 A900    STRT LDA #0
370 1393 852F        STA M          INITIALIZE MINIMUM
380 1395 8530        STA M+1        TO ZERO
390 1397 8531        STA M+2
400 1399 8532        STA M+3
410 139B 8564        STA CTRO       ALLOW PRINTING
420 139D 38          SEC
430 139E A50A        LDA LL         SET POINTER LW TO LAST
440 13A0 E904        SBC #4
450 13A2 8519        STA LW
460 13A4 A50B        LDA LL+1       LOCN IN SYMBOL TABLE
470 13A6 E900        SBC #0
480 13A8 851A        STA LW+1
490 13AA 206EB9      JSR BPF         INITIALIZE BASIC ROUTINE
500 13AD 206CA8      JSR CRL
510 13B0 A9FF    LOOP1 LDA #$FF     MAKE MP > ANY POSSIBLE
520 13B2 8534        STA MP+1       SYMBOL
530 13B4 ACCB12  LOOP2 LDY STS      SET PTR+Y TO TOP
540 13B7 ADCC12      LDA STS+1      OF SYMBOL TABLE
550 13BA 851C        STA PTR+1      DECREMENT Y AS TABLE
560 13BC A900        LDA #0         IS READ
570 13BE 851B        STA PTR
```

*(Continued)*

seem to be worthwhile. This article presents a symbol table lister for the OSI C1P (cassette version).

## Operation of the Assembler-Editor

In the OSI assembler, source code is stored in memory as it is read in, beginning at the location following the end of the assembler. Numbered lines are inserted at their correct position. Each line begins with two bytes containing the line number in hex, in the order low, high. The line ends with a return ($0D). Line feeds are not stored in the source text but are added during printing after each return. There is no special signal to indicate end-of-text as in BASIC; rather the editor keeps the next location available for text in a table (see below.) When an assembly is requested, a symbol table is built which begins at the last location in RAM and moves to successively lower addresses as more symbols are added. Each symbol requires six locations for storage: four bytes for the symbol itself (encoded) and two bytes for the value of the symbol. A symbol may be from one to six characters in length. It must begin with an alphabetic and the remaining characters must be in the set A-Z, 0-9, :, ., or $. The symbol table is not sorted, nor is a hash table used; the symbols are simply entered in the order in which they are encountered. A forward reference causes an entry to be made in the symbol table with a value which appears to be random. When the symbol is subsequently defined, its value is adjusted at that time.

## Operation of the Symbol Table Lister

The assembler maintains pointers to the start and end of source code and the start and end of the symbol table. These are shown as STMEM, NL, STS, and LL in the accompanying listing. Let me define "equivalence" as the numerical representation in which the symbol is stored, "value" as the value assigned to the symbol. E.g., "LABEL" always has the equivalence $4B2A2120; its value may be anything from $0000 to $FFFF.

The lister program begins by zeroing a 4-byte memory location, M. It then scans the symbol table to find the smallest equivalence greater than or equal to M (the smallest symbol numerically is also the earliest alphabetically.) The value of the found minimum equivalence is incremented by one and stored in M before the table is searched again. Thus the table is

**Symbol Table Lister** *(continued)*

```
 580 13C0 C010    LOOP3 CPY #$10       WHEN Y GETS BELOW $10
 590 13C2 B00E          BCS TRN        ADD $80 AND DECREMENT
 600 13C4 98            TYA            PTR BY $80 TO AVOID
 610 13C5 0980          ORA #$80       ADDRESSING ERRORS IF
 620 13C7 A8            TAY            Y DECREMENTS FROM
 630 13C8 A51B          LDA PTR        00 TO FF
 640 13CA 4980          EOR #$80
 650 13CC 851B          STA PTR
 660 13CE 1002          BPL TRN
 670 13D0 C61C          DEC PTR+1
 680 13D2 98      TRN TYA              COMPARE PTR+Y TO LW
 690 13D3 38            SEC            TO SEE IF SEARCH ENDED
 700 13D4 E903          SBC #3
 710 13D6 A8            TAY
 720 13D7 18            CLC
 730 13D8 651B          ADC PTR
 740 13DA 08            PHP
 750 13DB C519          CMP LW
 760 13DD D018          BNE CONT
 770 13DF 28            PLP
 780 13E0 A51C          LDA PTR+1
 790 13E2 6900          ADC #0
 800 13E4 C51A          CMP LW+1
 810 13E6 D00E          BNE CM1        IF MP+1=$FF THEN
 820 13E8 A534          LDA MP+1       SYMBOL TABLE EXHAUSTED
 830 13EA C9FF          CMP #$FF        SO QUIT   BUT IF
 840 13EC D048          BNE PRNT       MP+1<$FF THEN A SYMBOL
 850 13EE A900          LDA #0         HAS BEEN FOUND  PRINT IT
 860 13F0 8D0502        STA PFL        TURN OFF SAVE FLAG
 870 13F3 4C00E8        JMP EXM
 880 13F6 08      CM1 PHP
 890 13F7 28      CONT PLP             DOUBLE LOOP FOR 32-BIT
 900 13F8 A200          LDX #0         SUBTRACT
 910 13FA 38      CLOOP SEC            WHEN X=0, COMPARE
 920 13FB B11B          LDA (PTR),Y    CURRENT VALUE IN SYMBOL
 930 13FD F531          SBC M+2,X      TABLE WITH  M  IF VALUE
 940 13FF C8            INY            IS <M THEN OMIT 2d LOOP
 950 1400 B11B          LDA (PTR),Y    IF VALUE=>M THEN
 960 1402 F532          SBC M+3,X      COMPARE CURRENT VALUE
 970 1404 88            DEY            WITH MINIMUM (THIS LOOP)
 980 1405 88            DEY            IN MP  IF VALUE=>MP THEN
 990 1406 88            DEY            CONTINUE SEARCH    BUT
1000 1407 B11B          LDA (PTR),Y    IF VALUE<MP THEN
1010 1409 F52F          SBC M,X        REPLACE MP BY
1020 140B C8            INY            NEW MINIMUM
1030 140C B11B          LDA (PTR),Y
1040 140E F530          SBC M+1,X
1050 1410 08            PHP
1060 1411 E000          CPX #0
1070 1413 D008          BNE TMP
1080 1415 28            PLP
1090 1416 9019          BCC NXWORD
1100 1418 C8            INY
1110 1419 A204          LDX #4
1120 141B D0DD          BNE CLOOP
1130 141D 28      TMP PLP
1140 141E B011          BCS NXWORD
1150 1420 A200          LDX #0
1160 1422 88            DEY
1170 1423 B11B    MVMP LDA (PTR).Y     COPY SYMBOL (CODED) AND
1180 1425 9533          STA MP,X       ITS VALUE FROM PTR+Y
1190 1427 C8            INY            INTO MP
1200 1428 E8            INX
1210 1429 E006          CPX #6
1220 142B D0F6          BNE MVMP
1230 142D 98            TYA
1240 142E E905          SBC #5
1250 1430 A8            TAY
1260 1431 88      NXWORD DEY
1270 1432 88            DEY
1280 1433 4CC013        JMP LOOP3
1290 1436 A208    PRNT LDX #8          FILL PRINT BUFFER
1300 1438 A920          LDA #$20       WITH SPACES
1310 143A 951E    STB STA BFR-1,X
1320 143C CA            DEX
1330 143D D0FB          BNE STB
1340 143F B533    CPM LDA MP,X         COPY CURRENT MINIMUM TO
1350 1441 952F          STA M,X        GLOBAL MINIMUM
1360 1443 E8            INX
1370 1444 E004          CPX #4
1380 1446 D0F7          BNE CPM
1390 1448 E631          INC M+2        INCREMENT GLOBAL MIN.
1400 144A D002          BNE LOOP3.     FOR NEXT PASS
1410 144C E632          INC M+3
1420 144E A000    LOOP3. LDY #0        NOTE LOOP3. NOT= LOOP3
```

**Symbol Table Lister** *(continued)*

```
1430 1450 8414         STY XP
1440 1452 A901         LDA #1             LOAD BYTE COUNTER TO
1450 1454 8510         STA BC             SIGNAL FIRST PAIR OF
1460 1456 A203   LOOP4 LDX #3             BYTES BEING DECODED
1470 1458 B93300 LOOP4P LDA MP,Y
1480 145B 85DC         STA $DC
1490 145D C8           INY
1500 145E B93300       LDA MP,Y
1510 1461 85DD         STA $DD
1520 1463 C8           INY
1530 1464 BDDC15 LOOP5 LDA DVS,X
1540 1467 85DE         STA $DE
1550 1469 CA           DEX
1560 146A BDDC15       LDA DVS,X
1570 146D 85DF         STA $DF
1580 146F CA           DEX
1590 1470 8615         STX XSV
1600 1472 8416         STY YSV
1610 1474 A204         LDX #4
1620 1476 A900         LDA #0
1630 1478 95D7   STRZER STA $D7,X
1640 147A CA           DEX
1650 147B D0FB         BNE STRZER
1660 147D A210         LDX #$10           DECODE INTO ASCII
1670 147F 206CEE       JSR DVD            QUOTIENT RETURNED IN X
1680 1482 8A           TXA                IF QUOTIENT=0,END; EXIT
1690 1483 F04C         BEQ GADR           IF QUOTIENT=01 TO $1A THEN
1700 1485 C91B   NXCHR CMP #$1B           ALPHABETIC  ADD $40
1710 1487 900A         BCC ALPH           IF QUOTIENT=$1B TO $24 THEN
1720 1489 C925         CMP #$25           NUMERIC     ADD $15
1730 148B 9008         BCC NUM            IF QUOTIENT>$24 THEN : . OR $
1740 148D AA           TAX                TABLE LOOK-UP
1750 148E BDBB15       LDA CHR-$25,X
1760 1491 D004         BNE PP
1770 1493 692B   ALPH ADC #$2B
1780 1495 6915   NUM ADC #$15
1790 1497 A614   PP    LDX XP
1800 1499 951F         STA BFR,X          PUT ASCII CHAR INTO BFR
1810 149B E614         INC XP
1820 149D E002         CPX #2
1830 149F D006         BNE TSR
1840 14A1 C610         DEC BC
1850 14A3 A416         LDY YSV
1860 14A5 D0AF         BNE LOOP4
1870 14A7 A5D8   TSR LDA $D8
1880 14A9 D004         BNE TSTX
1890 14AB A5D9         LDA $D9
1900 14AD F022         BEQ GADR
1910 14AF A615   TSTX LDX XSV
1920 14B1 1008         BPL LPREP
1930 14B3 A5D8         LDA $D8
1940 14B5 A000         LDY #0
1950 14B7 84D8         STY $D8
1960 14B9 F0CA         BEQ NXCHR
1970 14BB A5D8   LPREP LDA $D8
1980 14BD 85DC         STA $DC
1990 14BF A5D9         LDA $D9
2000 14C1 85DD         STA $DD
2010 14C3 A416         LDY YSV
2020 14C5 D09D         BNE LOOP5
2030 14C7 C610   XIT DEC BC
2040 14C9 3006         BMI GADR
2050 14CB A615         LDX XSV
2060 14CD A416         LDY YSV
2070 14CF D087         BNE LOOP4P
2080 14D1 A200   GADR LDX #0              PRINT 8 CHARS FROM BFR
2090 14D3 B51F   GB$ LDA BFR,X
2100 14D5 20EEFF       JSR PRINT
2110 14D8 E8           INX
2120 14D9 E008         CPX #8
2130 14DB D0F6         BNE GB$
2140 14DD A205         LDX #5
2150 14DF B533         LDA MP,X           PRINT SAVED VALUE OF
2160 14E1 20ACEA       JSR PHEX           SYMBOL (CURRENT LOOP)
2170 14E4 CA           DEX                IN HEX
2180 14E5 B533         LDA MP,X
2190 14E7 20ACEA       JSR PHEX
2200 14EA A000         LDY #0             SET UP SEARCH OF ASCII
2210 14EC A200         LDX #0             SYMBOL FOR DUPLICATE
2220 14EE 8613         STX MCTR           CHARACTERS
2230 14F0 8612         STX CSV
2240 14F2 B91F00 LOOP6 LDA BFR,Y
2250 14F5 C8           INY
2260 14F6 C920         CMP #$20
2270 14F8 F01C         BEQ CXIT
```

searched once for each symbol to be printed. This method is not as efficient as a true sort, but it requires less memory. For a table of 100 symbols, the output is only slightly slower than the rate at which characters are written to the screen. After the minimum equivalence has been found in a particular pass (lines 510-1280), the symbol is decoded into its ASCII value (lines 1290-2070). The ASCII representation of the symbol is searched for multiple characters and converted to the form used by the assembler for source code (e.g., L666 = $4C363636 ➛$4C36FE) (lines 2080-2530).

Next, the source file is searched for the line defining the symbol (lines 2540-2950). If the symbol is not defined (and this will have caused an assembler error) the lister program prints a ? instead of a line number. Additionally, if the symbol is more than two characters long, the fourth character will be an embedded ?. Finally, the symbol, its value, and the line number where defined are all printed out (lines 2960-3250). This process is repeated until all symbols have been found and printed out.

To conserve memory space, the program uses routines stored in BASIC ROM and in an extended monitor EPROM (EXMON, DVD, and PHEX). The jump to EXMON at the end of the program (line 870) may be replaced by a jump to the monitor ($FE00) or to the assembler ($1300). The routines for 16-bit division (DVD) and printing a hex character (PHEX) are listed for the benefit of those who do not have the extended monitor. Since the assembler begins at $0240, PHEX can be relocated to $0222-0238 and DVD to the end of memory, provided the contents of STS are changed from $FF to $DD.

The program as written here begins at $1391 and runs to $15E3. The value in STMEM has accordingly been changed to $15E4. Note that this change must be made as soon as the assembler is loaded, before any source code is entered. This reduces the space available for an assembler source file by $253 (595 decimal) locations. If this reduction in space turns out to be crucial, the lister could be relocated to overlay part of the assembler. If this is done, the part of the assembler to be overlaid should be stored on tape. The assembler can then be reused by loading only the short overlay file rather than the entire program. The lister uses some page-zero locations for storage, but does not change any values required by the assembler, so the assembler can be re-run after running the lister if so desired. Output goes to the print vector at $FFEE which is a JMP (indirect) to $021A, 021B. These locations are initialized by the monitor to send output to the screen or tape, depending on the value in $0205. They can, of course, be changed to point to a print routine if a printer is available.

## Summary

This article presents a symbol table listing program written specifically for the cassette-based assembler furnished for the OSI C1P. The attached listing is followed by a symbol table printed out by this program. The assembler instruction manual appears to apply to the OS-65D version as well, though some addresses would need to be changed to run this lister program with it.

---

The author may be contacted at 13917 Congress Drive, Rockville, MD 20853.

**Symbol Table Lister** *(continued)*

```
2280 14FA C512           CMP CSV
2290 14FC F014           BEQ DUPL
2300 14FE 48             PHA
2310 14FF A513           LDA MCTR
2320 1501 F007           BEQ STOR
2330 1503 9527           STA DEST,X
2340 1505 E8             INX
2350 1506 A900           LDA #0
2360 1508 8513           STA MCTR
2370 150A 68        STOR PLA
2380 150B 9527           STA DEST,X
2390 150D E8             INX
2400 150E 8512           STA CSV
2410 1510 D0E0           BNE LOOP6
2420 1512 C613      DUPL DEC MCTR           DECREMENT MCTR FOR EACH
2430 1514 D0DC           BNE LOOP6          MULTIPLE CHARACTER
2440 1516 A513      CXIT LDA MCTR           IF NO DUPLICATE THEN
2450 1518 F003           BEQ CRTN           EXIT
2460 151A 9527           STA DEST,X         STORE NEGATIVE VALUE IN
2470 151C E8             INX                DEST IF DUPLICATE CHAR
2480 151D 8611      CRTN STX CC             NOW DEST IS IN ASM
2490 151F A920           LDA #$20           SOURCE FORMAT
2500 1521 9527           STA DEST,X
2510 1523 E8             INX
2520 1524 E008           CPX #8
2530 1526 D0F7           BNE CRTN+2
2540 1528 ACC912         LDY STMEM          SET UP SEARCH OF SOURCE
2550 152B ADCA12         LDA STMEM+1        CODE FOR SYMBOL
2560 152E 851E           STA PTR2+1
2570 1530 A900           LDA #0
2580 1532 851D           STA PTR2
2590 1534 A200      GORD LDX #0
2600 1536 CCFE12         CPY NL             IF SOURCE EXHAUSTED
2610 1539 D00A           BNE GORD.          AND NO MATCH FOUND
2620 153B A51E           LDA PTR2+1         THEN PRINT ?
2630 153D CDFF12         CMP NL+1
2640 1540 D003           BNE GORD.
2650 1542 4CC415         JMP QUEST
2660 1545 20D215    GORD. JSR INCY
2670 1548 8517           STA LN
2680 154A 20D215         JSR INCY
2690 154D 8518           STA LN+1
2700 154F 20D215    LS   JSR INCY
2710 1552 30FB           BMI LS             SKIP LEADING BLANKS
2720 1554 C920           CMP #$20           BOTH SINGLE AND MULT.
2730 1556 F0F7           BEQ LS
2740 1558 D003           BNE TNC
2750 155A 20D215    NC   JSR INCY
2760 155D D527      TNC  CMP DEST,X         COMPARE SOURCE CODE
2770 155F D00A           BNE NXLN$          TO SAVED SYMBOL
2780 1561 E8             INX
2790 1562 E411           CPX CC
2800 1564 F00E           BEQ FOUND          MATCH OF CORRECT #
2810 1566 D0F2           BNE NC             OF CHARACTERS
2820 1568 20D215    NXLN JSR INCY
2830 156B C90D      NXLN$ CMP #$0D
2840 156D F0C5           BEQ GORD
2850 156F 20D215         JSR INCY
2860 1572 D0F7           BNE NXLN$
2870 1574 20D215    FOUND JSR INCY          IF FOLLOWED BY TERMINATOR
2880 1577 C920           CMP #$20           THEN TRUE FIND
2890 1579 F00C           BEQ TRFIND         ELSE BURIED IN LONGER
2900 157B C90D           CMP #$0D           SYMBOL  CONTINUE SEARCH
2910 157D F008           BEQ TRFIND
2920 157F C92A           CMP #'*
2930 1581 F004           BEQ TRFIND
2940 1583 C93D           CMP #'=
2950 1585 D0E4           BNE NXLN$
2960 1587 A517      TRFIND LDA LN
2970 1589 85AE           STA $AE            SET UP CALL TO BASIC
2980 158B A518           LDA LN+1           CONVERSION ROUTINES
2990 158D 85AD           STA $AD
3000 158F A290           LDX #$90
3010 1591 38             SEC
3020 1592 20E8B7         JSR BFF            CONV HEX TO DECIMAL
3030 1595 206EB9         JSR BPF            CONV DEC TO ASCII
3040 1598 A000           LDY #0             STORED STARTING AT $0100
3050 159A B90001    TZ   LDA BCB,Y          SEARCH BCB FOR TERMINATOR
3060 159D F003           BEQ NX             ADJUST LEADING SPACES SO
3070 159F C8             INY                NUMBER IS RIGHT-JUSTIFIED
3080 15A0 D0F8           BNE TZ
3090 15A2 98        NX   TYA
3100 15A3 49FF           EOR #$FF
3110 15A5 18             CLC
```

**Symbol Table Lister** *(continued)*

```
3120 15A6 6908        ADC #8
3130 15A8 A8          TAY
3140 15A9 A920    SB: LDA #$20
3150 15AB 20EEFF  SB  JSR PRINT
3160 15AE 88          DEY
3170 15AF D0FA        BNE SB
3180 15B1 A000        LDY #0
3190 15B3 B90001  SN  LDA BCB,Y      PRINT LINE NUMBER
3200 15B6 F006        BEQ PXIT
3210 15B8 20EEFF      JSR PRINT
3220 15BB C8          INY
3230 15BC D0F5        BNE SN
3240 15BE 206CA8  PXIT JSR CRL
3250 15C1 4CB013      JMP LOOP1      CONTINUE
3260 15C4 A93F    QUEST LDA #'?      SYMBOL NOT FOUND IN
3270 15C6 8D0001      STA BCB        SOURCE   PRINT ?
3280 15C9 A900        LDA #0
3290 15CB 8D0101      STA BCB+1
3300 15CE A006        LDY #6
3310 15D0 D0D7        BNE SB:
3320 15D2 B11D    INCY LDA (PTR2),Y
3330 15D4 C8          INY
3340 15D5 D002        BNE IXT
3350 15D7 E61E        INC PTR2+1
3360 15D9 48      IXT PHA
3370 15DA 68          PLA
3380 15DB 60          RTS
3390 15DC         ; DIVISORS FOR CODED LABELS
3400 15DC 00      DVS .BYTE 0,$28,6,$40
3400 15DD 28
3400 15DE 06
3400 15DF 40
3410 15E0         ; NON-ALPANUMERICS ALLOWED IN LABELS
3420 15E0 3A      CHR .BYTE ':.$?'
3420 15E1 2E
3420 15E2 24
3420 15E3 3F
3430 EAAC             *=$EAAC
3440 EAAC         ;   PHEX
3450 EAAC 48          PHA
3460 EAAD 4A          LSR A
3470 EAAE 4A          LSR A
3480 EAAF 4A          LSR A
3490 EAB0 4A          LSR A
3500 EAB1 20B5EA      JSR PH1
3510 EAB4 68          PLA
3520 EAB5 290F    PH1 AND #$0F
3530 EAB7 0930        ORA #$30
3540 EAB9 C93A        CMP #$3A
3550 EABB 9002        BCC PH2
3560 EABD 6906        ADC #6
3570 EABF 4CEEFF  PH2 JMP PRINT
3580 EE61             *=$EE61
3590 EE61         ;   DIVIDE ROUTINE
3600 EE61 26DC    DIVIDE ROL $DC
3610 EE63 26DD        ROL $DD
3620 EE65 CA          DEX
3630 EE66 3017        BMI DV1
3640 EE68 26D8        ROL $D8
3650 EE6A 26D9        ROL $D9
3660 EE6C         ; DVD              ENTRY TO DIVIDE ROUTINE
3670 EE6C 38          SEC
3680 EE6D A5D8        LDA $D8
3690 EE6F E5DE        SBC $DE
3700 EE71 A8          TAY
3710 EE72 A5D9        LDA $D9
3720 EE74 E5DF        SBC $DF
3730 EE76 90E9        BCC DIVIDE
3740 EE78 85D9        STA $D9
3750 EE7A 98          TYA
3760 EE7B 85D8        STA $D8
3770 EE7D B0E2        BCS DIVIDE
3780 EE7F A4DD    DV1 LDY $DD
3790 EE81 A6DC        LDX $DC
3800 EE83 60          RTS
```

**Sample Symbol Table Listing**

```
ALPH     1493   1770
BC       0010     50
BCB      0100    220
BFF      B7E8    280
BFR      001F    160
BPF      B96E    290
CC       0011     60
CHR      15E0   3420
CLOOP    13FA    910
CM1      13F6    880
CONT     13F7    890
CPM      143F   1340
CRL      A86C    270
CRTN     151D   2480
CSV      0012     70
CTRO     0064    200
CXIT     1516   2440
DEST     0027    170
DIVIDE   EE61   3600
DUPL     1512   2420
DVD      EE6C    320
DVS      15DC   3400
DV1      EE7F   3780
EXM      E800    300
FOUND    1574   2870
GADR     14D1   2080
GB$      14D3   2090
GORD     1534   2590
GORD,    1545   2660
INCY     15D2   3320
IXT      15D9   3360
LL       000A     40
LN       0017    120
LOOP1    13B0    510
LOOP2    13B4    530
LOOP3    13C0    580
LOOP3.   144E   1420
LOOP4    1456   1460
LOOP4P   1458   1470
LOOP5    1464   1530
LOOP6    14F2   2240
LPREP    14BB   1970
LS       154F   2700
LW       0019    130
M        002F    180
MCTR     0013     80
MP       0033    190
MVMP     1423   1170
NC       155A   2750
NL       12FE    260
NUM      1495   1780
NX       15A2   3090
NXCHR    1485   1700
NXLN     1568   2820
NXLN$    156B   2830
NXWORD   1431   1260
PFL      0205    230
PHEX     EAAC    310
PH1      EAB5   3520
PH2      EABF   3570
PP       1497   1790
PRINT    FFEE    330
PRNT     1436   1290
PTR      001B    140
PTR2     001D    150
PXIT     15BE   3240
QUEST    15C4   3260
SB       15AB   3150
SB:      15A9   3140
SN       15B3   3190
STB      143A   1310
STMEM    12C9    240
STOR     150A   2370
STRT     1391    360
STRZER   1478   1630
STS      12CB    250
TMP      141D   1130
TNC      155D   2760
TRFIND   1587   2960
TRN      13D2    680
TSR      14A7   1870
TSTX     14AF   1910
TZ       159A   3050
XIT      14C7   2030
XP       0014     90
XSV      0015    100
YSV      0016    110
```

## COLOR COMPUTER USERS

### THE POWERFUL FLEX DISK OPERATING SYSTEM WITH HUNDREDS OF SOFTWARE PACKAGES IS NOW AVAILABLE!

Now you can run FLEX, OS-9 and Radio Shack disk software on your Color Computer. 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 (40¢ stamps) and we'll send it to you.

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.

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.

MOVEROM 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 plus an online help system are 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 disk 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**

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,' A/BASIC Compiler, plus more.

Application packages include; A/R, G/L, 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 $1,275. 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)474-7856

---

## PLAY THE NUMBERS GAME — AND WIN

### With ELF and TWG/ARIMA STATISTICAL SOFTWARE

**TWG/ARIMA** is invaluable to users interested in time series. This Box-Jenkins package identifies
- seasonal and non-seasonal differencing
- Box-Cox transformations estimates
- correlation between coefficients
- Box-Pierce Q statistics and their significance
- t statistics
- probabilities and predicts
- various lead times
- starting points
- confidence levels for time series forecast data

$300.00

**ELF** performs
- factor analysis
- multiple regression
- stepwise, discriminant analysis
- cross tabulations
- 1 and 2 way ANOVA

as well as all basic statistics, including skewness, kurtosises, etc. and Chi-Square. ELF will also create and edit a data base as well as visually depict data in scattergrams and histograms (bar graphs)...and more

$200.00

Mastercard/VISA

Each program comes with database manager, numeric software keypad and requires an Apple II with Applesoft, 48 K, and DOS 3.3

To order write to

The Winchendon Group  
3907 Lakota Road  
P.O. Box 10114 M  
Alexandria, VA 22310

(703) 960-2587

**THE WINCHENDON GROUP**

---

## 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, decompiling, 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.*

---

## PRINT ][

### THE ULTIMATE PRINT STATEMENT FOR THE APPLE ][ COMPUTER.

| Standard Apple Output | PRINT ][ Output |
|---|---|
| $1234.56 | $1234.56 |
| $12 | $ 12.00 |
| $1.9 | $ 1.90 |
| $.75 | $ .75 |
| $765 | $ 765.00 |
| *Print "$"; I (without Print ][)* | *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 ][.**

Available at your local dealer for $24.95 or write to:

VISA  
MasterCard

**Computer Systems Design**  
2139 Jackson Blvd.  
Rapid City, SD 57701  
(605) 341-3662

MICRO™

Dr. William R. Dial
438 Roslyn Avenue
Akron, Ohio 44320

# Resource Update

**A list of magazines which contain information about the 6502/6809 microprocessors on a reasonably regular basis.**

Did you ever wonder just what magazines are rich sources of information on the 6502/6809 microprocessors, 6502/6809-based microcomputers, accessory hardware and software? For several years this writer has been assembling a bibliography of 6502/6809 references related to hobby and small business systems. The accompanying list of magazines has been compiled from this bibliography. An attempt has been made to give up-to-date addresses and subscription rates for the magazines cited. Subscription rates are for the U.S. Rates to other countries normally are higher.

## GENERAL 6502/6809

**MICRO: The 6502/6809 Journal**
$24.00 per year, 12 issues
P.O. Box 6502
Chelmsford, MA 01824

**Compute!**
$20.00 per year, 12 issues
515 Abbott Drive
Broomall, PA 19008

## GENERAL COMPUTER

**Byte**
$19.00 per year, 12 issues
Byte Subscriptions
P.O. Box 590
Martinville, NJ 08836

**Computer Shopper**
$10 per year, 12 issues
Glenn Patch, Editor
P.O. Box F
Titusville, FL 32780

**Computing Today**
£ 15.15, 12 issues
Argus Specialists Publications, Ltd.
145 Charing Cross Road
London WC2H 0EE
England

**Creative Computing**
$20.00 per year, 12 issues
P.O. Box 789-M
Morristown, NJ 07960

**CSRA Computer Club Newsletter**
$6.00 per year
P.O. Box 284
Augusta, GA 30903

**Dr. Dobb's Journal**
$21.00 per year, 12 issues
People's Computer Co.
P.O. Box E
1263 El Camino Real
Menlo Park, CA 94025

**Epson Information Exchange Newsletter**
$12.00 per year, 12 issues
Epson Users Group
c/o Frank Barden
136 Candlewick Drive
Wendell, NC 27591

**IEEE Micro**
$23.00 per year, quarterly
IEEE Computer Society
10662 Los Vaqueros Circle
Los Alamitos, CA 90720

**Interface Age**
$18.00 per year, 12 issues
McPheters, Wolfe and Jones
16704 Marquardt Ave.
Cerritos, CA 90701

**Microcomputing**
$25.00 per year, 12 issues
Wayne Green, Inc.
80 Pine Street
Peterborough, NH 03458

**Microcomputer Index**
$30.00 per year, quarterly
Microcomputer Information Services
2646 El Camino Real, Box 247
Santa Clara, CA 95051

**Personal Computer World**
£ 8.00, 12 issues
Sportscene Publishers (PCW) Ltd.
14 Rathbone Place
London W1P 1DE
England

**Personal Computing**
$18.00 per year, 12 issues
Hayden Publishing Co.
4 Disk Drive, Box 1408
Riverton, NJ 08077

**Popular Computing**
$15.00 per year, 12 issues
Byte Publications
P.O. Box 307
Martinville, NJ 08836

**Practical Computing**
£ 10.00, 12 issues
IPC, Business Press, Ltd.
Quadrant House
The Quadrant, Sutton
Surrey SM2 5AS
England

**Purser's Magazine**
$12.00 per year, 4 issues
c/o Robert Purser
P.O. Box 466
El Dorado, CA 95623

**SoftSide**
$30.00 per year, 12 issues
P.O. Box 68
Milford, NH 03055

**Spreadsheet**
$25.00 per year
Visigroup—Visicalc User Group
P.O. Box 254
Scarsdale, NY 10583

## APPLE-RELATED PUBLICATIONS

**The Abacus II Newsletter**
$18.00 per year, 6 issues
5302 Camino Alto Mira
Castro Valley, CA 94546

**Adam and Eve Newsletter**
$12.00 per year
Adam and Eve Apple II Users' Group
11 South Hancock Street
Madison, WI 53703

**Apple**
$2.00 per issue, quarterly
Apple Computer Co.
10260 Bandley Drive
Cupertino, CA 95014

**Apple Assembly Line**
$15 per year, 12 issues
S-C Software
P.O. 280300
Dallas, TX 75228

**Apple Barrel**
$18.00 per year, 9 issues
c/o Lee E. Gilbreth
Houston Area Apple Users Group
3609 Glenmeadow Dr.
Rosenberg, TX 77471

**Apple-Dayton Newsletter**
$18.00 per year
39 Mello Ave.
Dayton, Ohio 45410

*(continued)*

## Resource Update *(continued)*

**Apple Bits**
$20.00 per year, 10 issues
$2.00 application fee
NEO Apple Corps
c/o Norm Henke
21750 Miller Ave.
Euclid, OH 44119

**The Apple-Dillo**
$15.00 per year, 12 issues
c/o Lenard Fein
River City Apple Corps
2015 Ford St.
Austin, TX 78704

**Applegram**
$15.00 per year
Apple's British Columbia Computer Society
316-8055 Anderson Rd.
Richmond, B.C.,
Canada V6Y 1S2

**Apple Gram**
$15.00 per year, 12 issues
The Apple Corps of Dallas
P.O. Box 5537
Richardson, TX 75080

**Apple Mug Newsletter**
Apple Medical User's Group
2914 Katella, Suite 208
Orange, CA 92667

**The Apple Orchard**
$10.00 per year, 6 issues
International Apple Core
910 A George St.
Santa Clara, CA 95050

**Apple Peel**
$20.00 per year, 12 issues
Jerry Jenkins, Editor
The Birmingham Apple Corps
774 Twin Branch Drive
Birmingham, AL 35226

**Apple/Sass**
$12.00 per year, 12 issues
Honolulu Apple User's Society
P.O. Box 91
Honolulu, HI 96810

**Applesauce**
$12.00 per year, 6 issues
Original Apple Corps
P.O. Box 598
Venice, CA 90291

**AppleSeed Newsletter**
$15.00 per year, 12 issues
P.O. Box 12455
San Antonio, TX 78212

**ByteLines**
$12.00 per year, 12 issues
Hi Desert Apple Computer Club
P.O. Box 2702
Lancaster, CA 93534

**Call —A.P.P.L.E.**
$20.00 per year, 12 issues
$25.00 application fee
304 Main Ave. S., Suite 300
Renton, WA 98055

**The Cider Press**
$20.00 per year, 12 issues
San Fransisco Apple Core
1515 Sloat Blvd., Suite 2
San Fransisco, CA 94132

**The C.I.D.E.R. Press**
$10.00 per year
Apple Computer Information
and Data Exchange of Rochester
P.O. Box 2359
Rochester, NY 14623

**Erie Apple Crunchers Express**
$12.00 per year
P.O. Box 1575
Erie, PA 16507

**From The Core**
$12.00 per year, 12 issues
Carolina Apple Core
P.O. Box 31424
Raleigh, NC 27622

**FWAUG Newsletter**
$15.00 per year, about 9 issues
Lee Meador, Editor
Fort Worth Area Apple User Group
1401 Hillcrest Drive
Arlington, TX 76010

**The G.R.A.P.E. Vine**
$7.50 per year, 12 issues
Group for Religious Apple
Programming Exchange
c/o Stephen Lawson
P.O. Box 283
Port Orchard, WA 98366

**Hardcore Computing**
$20.00 per year, quarterly
P.O. Box 44549
Tacoma, WA 98444

**The Harvest**
$12.00 per year, 10 issues
Northern Illinois Apple User Group
1015 S. Ridge Rd.
Arlington Heights, IL 60005

**L.A.U.G.H.S.**
$15.00 per year
c/o Sam Ward
Louisville Apple User Group
8002 Canna Dr.
Louisville, KY 40258

**Maple Orchard**
$25.00 (Canadian) per year, 6 issues
Apple Users Group of Toronto
(Apple-Can)
P.O. Box 696
Willowdale, Ont.,
Canada M2K 2P9

**The Michigan Apple-Gram**
$12.00 per year, 10 issues
The Michigan Apple Computer Club
P.O. Box 551
Madison Heights, MI 48071

**Mini'App'Les Newsletter**
$12.00 per year
Mini'App'Les Apple
Computer User Group
Box 796
Hopkins, MN 55343

**Newsletter**
$10.00 per year
Apple Bytes of Buffalo
c/o Hank Kolk
171 Tree Haven Road
Buffalo, NY 14215

**Nibble**
$19.95 per year, 8 issues
S.P.A.R.C.
P.O. Box 325
Lincoln, MA 01773

**OKC Apple Times**
$10.00 per year, 6 issues
OKC Apple User Group
c/o Glenn H. Rodkin
3728 No. Frankford
Oklahoma City, OK 73112

**Peelings II**
$21.00 per year, 9 issues
The Peelings Co.
P.O. Box 188
Las Cruces, NM 88004

**Poke-Apple**
$15.00 per year
Apple-Siders
5707 Chesapeake Way
Fairfield, OH 45014

**Rubber Apple Newsletter**
12.00 per year, 10 issues
c/o Ken Gabelman
849 Russel Ave.
Akron, OH 44307

**The Scarlet Letter**
Big Red Apple Club
1301 North 19th
Norfolk, NE 68701

**The Seed**
$18.00 per year, 12 issues
P.O. Box 17467
Denver, CO 80217

**Softline**
Gratis, 6 issues per year
11021 Magnolia Blvd.
North Hollywood, CA 91601

*(Continued on page 110)*

# MICRO™
## Software Catalog

Name: **Payroll System**
System: OS65U
Memory: 48K
Language: BASIC
Hardware: OSI C-2 or C-3 series

Description: This integrated portion of EIS General Accounting System prepares payroll for hourly and salaried employees while accumulating information for tax reporting purposes. It includes all tax tables and optional payroll check writing.

Price: $800.00
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: **The Home Accountant**
System: Apple II, 1 disk drive (2nd recommended)
Memory: 48K
Language: Applesoft
Hardware: Applesoft in ROM, printer (132-column optional but recommended)

Description: Package offers a 100-budget category, keeps track of up to 5 checking accounts, cash and credit cards. Prints checks (if desired), prints a personal balance sheet and net worth statement. Allows multiple diskettes, fast bank reconciliation, and automatic transfers. Custom search and retrieval, and graphics for any category by bar graph, line graph and trend analysis.

Price: $74.95 (Retail)
Includes one program diskette, manual and binder

Authors: Robert Schoenburg, Stephen Pollack, Larry Grodin

Available:
Continental Software
16724 S. Hawthorne Blvd.
Lawndale, CA 90260
(213) 371-5612

---

Name: **Stock Forecasting System**
System: Apple II Plus or equivalent
Memory: 48K
Language: BASIC
Hardware: One disk drive minimum. Can use 2 drives, printer modem and graphics tablet.

Description: Fifteen menu-driven programs are used to determine buy and sell points for individual stocks. *System* provides technical analysis of stock prices. Complete data file control. Programs are locked, but user may make unlimited copies for his own use.

Price: $175
Includes program diskette, data disk, hardware lock and manual

Author: C. Edward Walter

Available:
Urban Aggregates, Inc.
6431 Brass Knob
Columbia, MD 21044

---

Name: **Elements of Music**
System: Micro Plato/Apple II Plus
Memory: 48K
Language: Micro Tutor, Applesoft BASIC
Hardware: One disk drive

Description: This computer program was developed for use with children and non-music majors who wish to learn the elements in music at an entry level. Content lessons include: pitch names, pitches on the keyboard, key signatures. Lesson disk includes student records for 50 users with data collection for all lessons on the disk.

Price: $175 Apple
$225 Micro Plato
Includes instructor manual, student instruction sheet, floppy disk lesson.

Author: John M. Eddins, Robert L. Weiss, Jr.

Available:
Electronic Courseware Systems, Inc.
P.O. Box 2374, Station A,
Champaign, IL 61820

---

Name: **Arith-Magic**
System: PET
Memory: 16K
Language: BASIC
Hardware: Diskette and Tape

Description: These three mathematics programs — Diffy, Tripuz and Magic Squares — are highly interactive and present motivated practice in whole number operations along with exploration of concepts. They are appropriate for grades 3-8 and come with thorough teacher documentation.

Price: $35.00
Includes postage and handling

Author: Joanne B. Rudnytsky

Available:
Quality Educational Designs
P.O. Box 12486
Portland, OR 97212

---

Name: **Russian Disk**
System: Apple
Memory: 32K
Hardware: Disk Drive

Description: The *Russian Disk* package contains two sets of programs: the first set teaches the Russian (or Cyrillic) Alphabet by first introducing the letters that most resemble their English counterparts and then the more exotic ones bit by bit. The second set of programs develops reading ability in Russian and expands the user's Russian vocabulary. It also provides a chance to practice typing in Russian using the Cyrillic characters learned in part one.

Price: $24.95

Author: Constance Curtin

Available:
Instant Software, Inc.
Peterborough, NH 03458
(603) 924-9471

---

Name: **Waterloo microPascal**
Hardware: Commodore SuperPET, Volker-Craig 2900, 3900, 4900, Northern Digital microWAT

Description: *Waterloo microPascal* is an extensive implementation of Pascal, corresponding very closely to draft proposals being produced by the International Standards Organization (ISO) Pascal committee. The ISO draft language is a refinement of the language originally defined by Wirth, varying only in minor aspects. This implementation includes sophisticated features such as text file support, pointer variables, and multi-dimensional arrays. A significant feature of Waterloo microPascal is its powerful interactive debugging facility.

Available:
Waterloo Computing Systems Limited
158 University Ave. W.
Waterloo, Ontario, Canada
N2L 3E9

---

Name: **Liquid and Gas Flow Calculation**
System: Apple II, Apple II Plus
Memory: 32K
Language: Applesoft BASIC
Hardware: DOS 3.2 II or DOS 3.3 II with controller card

Description: Menu-driven flow calculation programs, to find physical properties on flow of gases or liquid for tube data engineering. Two main programs calculate quantity, inlet pressure, outlet pressure, flow coefficient, specific gravity, temperature, selected tubing material's outer diameter's allowable wall thickness. Also includes tube volume calculation program. Considers the elevated temperature in high-pressure condition of unknown medium. Utilizes several sophisticated exponential calculation routines to find ultimate pressure ratings of both tubing data (aluminum, copper, carbon steel and two kinds of stainless steel) are stored in data file.

Price: $40.00
Includes seven programs on diskette.

Available:
American Avicultural Art & Science Inc.
3268 Watson Road
St. Louis, MO 63139
(314) 645-4431

*(Continued on next page)*

Name: **Escape From Arcturus**
System: Apple II
Memory: 48K
Language: Applesoft/ Machine Code
Hardware: DOS 3.3

Description: This machine-language program has colorful high-resolution animation along with sound efects. The two-part player action begins with commanding the Space Fortress. Surrounded by attacking Griplems, you fend off photon torpedoes, fighters, cruisers, and deal with their force fields — all at the same time. Moving to the escape pilot mode, you repel a variety of landing craft while saving the population of Arcturon. Requires an Apple II with Applesoft firmware, 48K, DOS 3.3 system with paddles, and quick reflexes.

Price: $35

Available:
Synergistic Software
5221 120th Ave. S.E.
Bellevue, WA 98006

---

Name: **Job Control System™**
System: Apple III
Memory: 64K
Language: Pascal
Hardware: Apple II with Pascal language and a 132-column printer

Description: High Technology Software Products, Inc. brought computer-assisted job control to the small-to-medium-size companies in manufacturing, construction, and service industries by introducing the *Job Control System™* for the Apple II early in 1981. JCS is now available for the Apple III. JCS provides management with reliable measures of productivity by furnishing up-to-the-minute job status data for determining the real cost of producing a product or providing a service.

Price: $750.00
Includes program diskette, 3-ring binder with complete documentation

Author: Mark Nettlingham

Available:
High Technology Software Products, Inc.
P.O. Box 14665
2201 N.E. 63rd St.
Oklahoma City, OK 73113
and other computer retailers worldwide

Name: **Absolute Security**
System: Apple II and Apple II Plus, DOS 3.3
Memory: 48K Bytes
Language: Applesoft BASIC, Machine Code (6502)
Hardware: Small audio amplifier (any cassette recorder), Micromodem II

Description: *Absolute Security* protects confidential modem communications by encoding uppercase ASCII text files with an *unbreakable* code. The security of the system relies on user-generated keys. It is, therefore, possible for every Apple user to own and use *Absolute Security* with equal total privacy and protection. Based on the "One-Time Pad" concept, *Absolute Security* is invulnerable to decoding by even Government superpowers.

Price: $79.95 ppd. in the USA
Includes operation/ configuration manual, 3.3 diskette, 2-system license

Author: Dann McCreary

Available:
Dann McCreary Software
Box 16435-MI
San Diego CA 92116

*Note: Absolute Security* is currently available only for shipment within the U.S., per U.S. State Department directive. Foreign inquiries only accepted.

---

Name: **Vigil**
System: VIC-20
Memory: 4K
Language: Assembler
Hardware: VIC with 3K memory expander

Description: A powerful new language for programming interactive games. Allows use of all features available on VIC (color, sound, light pen, game paddles and joysticks). With *Vigil* you can create action-packed games that rival machine-language coded games in speed, but in a fraction of the development time. Comes with nine preprogrammed games to get you started immediately.

Price: $35.00
Includes 80-plus-page manual and nine full length sample games

Author: Roy Wainwright

Available:
Abacus Software
P.O. Box 7211
Grand Rapids, MI 49510
(616) 241-5510

Name: **Waterloo microSUPERVISOR**
Hardware: Commodore SuperPET, Volker-Craig 2900, 3900, 4900, Northern Digital microWAT

Description: The *Waterloo microSUPERVISOR* is an operating system designed for single-user microcomputer environments. It includes monitor, library and serial line communications support. The monitor program supports loading of linker-produced program files into bank-switched RAM memory or normal RAM memory. The monitor also provides facilities which are useful for debugging machine-language programs. A library of functions and procedures is included for general use by other programs included in the package. The library includes support functions for input/output operations to the keyboard, screen and peripheral devices. A serial line setup program is included which permits the selection of programmable characteristics, such as baud rate, of RS-232 serial lines. In addition, this program includes support for establishing communication with a host computer, through a serial line, for the purpose of accessing its files and peripheral devices.

Available:
Waterloo Computing Systems Limited
158 University Ave. West
Waterloo, Ontario
Canada N2L 3E9

---

Name: **Math Drill Gamepac**
System: OSI C1P, Superboard; TRS-80 Model I, III
Memory: OSI - 8K
TRS-80 - 16K
Language: BASIC
Hardware: Cassette

Description: *Math Drill Gamepac* consists of three separate games for making math drills fun. Math Wars provides random drill problems in BASIC addition, subtraction, multiplication, and division with a choice of difficulty from one- to three-digit numbers. Fraction-Action follows the dreaded Gator-Hator at feeding time while drilling at all aspects of fractions. Decimal Tic-Tac-Toe requires correct answers to decimal problems in order to play the popular game. Excellent for grades K through 6.

Price: $11.95 per game
$29.95 for complete Gamepac
Add $2.00 for shipping
Includes separate tape for each game, instructions

Author: Doug Jenkins and Garry Taylor

Available:
Tripod Productions
Box 71, Route 11
Bowling Green, KY 42101

---

Name: **BASXR**
System: 0S65U
Memory: 48K
Language: BASIC
Hardware: Ohio Scientific C-2 or C-3 series

Description: Helps with debugging and modification. Lists all variables and/or commands and their line number locations. Locates specific lines on entry of decimal value of BASIC commands.

Price: $45 - program
$50 put on your OS65U disk
Includes program listing and documentation manual.

Available:
Electronic Information Systems, Inc.
P.O. Box 5893
Athens, GA 30604
(404) 353-2858

---

Name: **Electronics III #26014**
System: Apple II or Apple II Plus
Memory: 32K RAM
Language: Applesoft

Description: The programs in this package are used to analyze both periodic and aperiodic waveforms along with various circuits: average and RMS values of a periodic function, fourier series expansion of a periodic function; fourier transform and spectrum plot, analysis of damped excillations, impedance matching pads, and PI-TEE (delta-wye) transforms.

Price: $39.95
Includes documentation.

Available:
Advanced Operating Systems
450 St. John Rd.
Suite 792
Michigan City, IN 46360
(219) 879-4693

*(Continued on next page)*

NEW 6809 SYSTEM!

Now, for about the same price as you would expect to pay for the memory capacity alone, you can have a complete single board computer with these features:

- ✲6809 CPU, 1MHz clock
- ✲192KB RAM included, sockets for 64KB more
- ✲84X24 display of a 7X12 character font
- ✲Keyboard interface for an un-encoded switch matrix
- ✲Floppy controller for two 5" drives, single or double sided, up to 80 tracks
- ✲Parallel printer port
- ✲Serial I/O port
- ✲General purpose 8-bit parallel I/O port
- ✲Parallel expansion port
- ✲Dimensions: 8.6 by 10.3 inches

The FLEX operating system is supported by our device drivers. BASIC, PASCAL, and C are available for FLEX. The device drivers (in EPROM) include advanced features like auto-repeat for the keyboard, and track buffering for the disks. Commented source code of all EPROM contents is supplied.

For more information, send a stamped self-addressed envelope and we will send you a configuration guide that explains how to set-up a system. An assembled board is purchased by sending check or money order for $735 per board. (California add 6% sales tax).

Chandler Microsystems
22051 COSALA
MISSION VIEJO, CA 92691

FLEX, trademark Technical Systems Consultants, Inc.

***COMING SOON...***

The fastest, most comprehensive, most responsive software ever written for microcomputers...

**FEDDER SOFTWARE** by Interactive Computer Systems, for CP/M multi-user hard disk environment...

**ACU-LEDGER** by National Software, for the Apple II without CP/M...

Both of these flexible, high-speed file access systems will outperform their competition in respective categories...

Both will be distributed by

EASTERN SOFTWARE DISTRIBUTORS INC.
17 COMMERCE ST. BALTIMORE, MD 21202

Write to Eastern for details and the name of the Eastern dealer nearest you.

APPLE GRAPHICS USERS – SEE YOUR PICTURES IN A FLASH!

APPLE FLASHER™

High Speed Search And Display Of Hires Screens.
1.5-2 Second Display Time!
Uses Standard Files.
Slide Projector Simulation.
Continuous Display Modes.
Requires Applesoft, DOS 3.3.

$34⁵⁰

N.Y. State Residents Add Applicable Sales Tax.
We Accept Master Charge and Visa.
Include $1.00 handling.

CROW RIDGE ASSOCIATES, INC.
P.O. BOX 90: NEW SCOTLAND, N.Y. 12127
(518) 765-3620

the PRIME FACTOR, inc.

**PRIME FACTOR BASIC 2.2**

the PRIME FACTOR, Inc. presents PRIME FACTOR BASIC, a fast and powerful Applesoft™ compatible, 500 digit machine language math system.

PRIME FACTOR BASIC provides the tools to play the 3000 year old adventure game of large numbers. With the added commands you may explore for large mersenne primes, compute the natural logarithm e to 500 places, factor large numbers, experiment with a 129 digit public key trap door function and much more.

PRIME FACTOR BASIC is delivered on an Apple II DOS 3.3 diskette with numerous demonstration programs. The user manual takes you step by step through the easy commands which form a natural extension to Applesoft. Simple and array variables are supported. The functions +, −, ×, /, ^, ABS(X), SGN(X), MOD(X/Y), MOD(X^Y/Z), GCD(X,Y), SQR(X), SQREM(X), LEN(X) and IF . . . THEN testing are some of the tools available in this mathematics adventure package.

**PRIME FACTOR BASIC . . . . . . . . . . . . . . . . . . . . . $79.95**

*California residents add 6% tax*

Apple II, Applesoft and DOS 3.3 are trademarks of Apple Computer Inc.
PRIME FACTOR BASIC and PRIME FACTOR LOCK are trademarks of the PRIME FACTOR, Inc.

**14713 OXNARD STREET • VAN NUYS, CA 91411 • (213) 908-1838**

## Software Catalog *(continued)*

Name: **Videotrek**
System: OSI Cassette Systems
Memory: 8K
Language: BASIC
Description: Command the Starship "Challenger" on a high-speed chase through the galaxy as you try to destroy the Klingon invasion fleet. You must navigate through stars, black holes and planets, and watch out for the doomsday machine! From the author of *Time Trek*, this is an action graphics arcade version of *Star Trek*, with several levels of difficulty. Uses machine code and has sound for all C1Ps and color and sound for C4Ps. Cassette only. Specify your system!
Price: $9.95
Author: Bob Retelle
Available:
Pretzelland Software
2005 Whittaker Rd.
Ypsilanti, MI 48197
(313) 483-7358

Name: **Time Dungeon -World #26052**
System: Apple II or Apple II Plus
Memory: 32K RAM
Language: Applesoft
Description: This game package offers the user three different games, each focusing on a different time period. The operator must answer questions correctly in order to map his way out of a dungeon with as many pieces of gold as possible. The time period focused upon depends upon the game chosen. The three programs are: Ancient History 4,000 BC-6BC; World History WW I 1894-1919; World History WW II 1933-1945.
Price: $29.95
Includes documentation
Available:
Advanced Operating Systems
450 St. John Road
Suite 792
Michigan City, IN 46360

Name: **Color Pac Attack**
System: Radio Shack Color Computer, TRS-80C
Memory: 16K
Language: Assembly
Hardware: TRS-80C, joysticks
Description: *Pac Attack* brings the fun of the arcades to the Color Computer. Three little muggers chase your man relentlessly around a maddening maze as you furiously try to build up points. This game's great graphics and sound effects offer continuous action at three levels of difficulty for computer buffs of all ages.
Price: $24.95
Includes cassette and instructions
Author: Computerware
Available:
Computerware
P.O. Box 668
Encinitas, CA 92024

Name: **Tax-Manager**
System: Apple II or Apple II Plus
Memory: 48K
Language: Applesoft in ROM
Hardware: Disk II
Description: Get help preparing your federal income taxes and printing the schedules. This easy-to-use program includes the latest tax laws and will remain current with our Extended Warranty option.
Price: $150.00
Author: Taso
Available:
Micro Lab
2310 Skokie Valley Rd.
Highland Park, IL 60035

Name: **TWERPS**
System: Apple II or Apple II Plus
Memory: 48K
Language: Assembly
Hardware: Disk II
Description: Now, from the company that brought you *Sneakers, Beer-Run* and a host of other blockbusters, comes *TWERPS* — another whimsical game of skill with eight levels of play and a cast of top-starring characters. Meet Captain Twerp! Thrill at the shooting Oribters! Be amazed by the swooping Glingas! Gasp in terror at the drooling Gleepnoks! Sit on the edge of your seat as you try to get back to the mother ship before your fuel runs out! Now at your neighborhood stores.
Price: $29.95
Includes diskette and instruction booklet
Author: Dan Thomson
Available:
Your local computer store

Name: **Spelltest**
System: Any 6809 system running FLEX or OS-9 operating system
Memory: 32K to 56K total RAM (including the 8K used by operating system)
Language: 6809 machine code
Hardware: SWTPC, GIMIX, or any other 6809 system. Two disk drives recommended
Description: Menu driven spelling checker. User friendly, designed to be used by a secretary. *Spelltest* is interactive, letting you check each word in context. It writes a corrected file for you after checking. Also allows you to build an additional dictionary of new words. You may accept, accept and save, or replace each suspect word. Suspected misspelled words may be listed on terminal or printer, checked individually or checked in context. Also will print list of valid words. Fast operation. 22,000+ dictionary.
Price: $199.95
Includes 22,000 word dictionary and binary code and instructions. Source $100 additional.
Author: Dale L. Puckett
Available:
Frank Hogg Laboratory
130 East Water
Syracuse, NY 13210
(315) 474-7856
Master Card and VISA accepted

Name: **W7AAY RAE to BASIC File Transfer Program**
System: Synertek SYM-1
Memory: 8K minimum
Language: BAS-1 and RAE-1
Description: This 512-byte ROMable program allows RAE text files to be transferred to BASIC and for BASIC programs to be transferred to RAE. Now you can use the powerful RAE editor to create and modify BASIC programs. Fully documented source code in RAE format supplied on cassette tape with instructions.
Price: $15.00 ppd. in USA
$17.00 foreign
Author: John M. Blalock
Available:
Blalock and Associates
P.O. Box 39356
Phoenix, AZ 85069

Name: **Entertainment Software for Ohio Scientific**
System: OSI C2-4P or C4P micros
Memory: 8K
Language: BASIC
Hardware: Cassette, 5¼" disk, or 8" disk
Description: Micronics Computerware is introducing a line of software beginning with the following full feature games: *Breakout, Box-In Hangman, Crossball,* and *Battleship.* All the games take full advantage of OSI's sound, color and graphics features. Interested parties should write or phone for complete information.
Price: $6.95 (cassette)
$9.95 (disk) - write for more information on disks.
Includes cassette or disk, full documentation
Available:
Micronics Computerware
750 Auburn Avenue
Buffalo, NY 14222

Name: **DOW2000 & Option43**
System: Apple II
Memory: 48K
Language: Applesoft
Hardware: Disk 3.3/3.2 and printer option
Description: Stock Market Analysis will determine price projections based on a stock's BETA coefficient or Relative Strength # and the Dow Jones Average. Projections are made as you vary the DOW (What if...); on one stock or entire portfolio with single scan, quick scan, or variable scan of values. The option program will give you the percent of increase of the option months to determine which month and strike price option to buy for a given stock. Included is the booklet "The Art of Timing Your Stock's Next Move." Author in market 17 years and former Registered Investment Advisor with S.E.C.
Price: $29.95
Includes booklet (booklet alone $5.95)
Author: CIAC: Patrick and David Calabrese
Available:
Bit 'n Pieces Series
P.O. Box 7035
Erie, PA 16510

*(Continued on next page)*

## Software Catalog *(continued)*

Name: **FORTH Programming Aids**
System: fig-FORTH model (FORTH-79 in preparation)
Memory: free dictionary space: 3K minimum, 13K recommended
Language: High level FORTH
Hardware: Any with the above FORTH systems

Description: These routines aid in development and debugging and complement cross/meta compilers with commands to perform the following: (1) decompile words into structured FORTH code (it generates IF, ELSE, etc.), optionally to disk; (2) find (and decompile) all words called *by* a specified word; (3) find all calls *to* a specified word; (4) create a one-to-one translation of FORTH run-time code.

Price: $150.00 (California residents add 6.6%. Foreign air add $15.)
Includes 40-page manual and all source code

Author: T.E. Wempe, R.E. Curry

Available:
Curry Associates
P.O. Box 11324
Palo Alto, CA 94306

---

Name: **Small-C for 6809**
System: 6809 with FLEX9 or DOS 69D (OS-9-planned)
Memory: 48K recommended
Language: C
Hardware: Any which will run above operating systems

Description: Proper subset of C except for #ASM extension. Based originally on Small-C by Ron Cain, with a few extensions. Generates relocatable code, special loader supplied for TSC's absolute assembler. SSB users should furnish description of hardware and OS revision level. Version 1.0 now, 2.0 July.

Price: $52.50 (5") version 1.0
Includes source for run-time library, compiler-tester, and FLEX loader.

Author: Allan Batteiger, Bill Knight, Howard Harkness

Available:
Word's Worth
P.O. Box 28954
Dallas, TX 75228

---

Name: **Fun With Math Vol. 1**
System: Apple II, Apple II Plus
Memory: 48K with DOS 3.3 and FP in ROM
Language: Applesoft and Integer (RAM Integer on disk)
Hardware: Apple II or Apple II Plus computer, DOS 3.3, game controllers *not* required.

Description: Educational programs present drill in the four basic operations in a highly motivational game format using the Apple's graphics and sound capabilities. All programs offer immediate reinforcement and have two levels of difficulty, which is under teacher control. These programs are designed to be "childproof" and are almost impossible to "crash." The programs were written by a teacher who has had 17 years classroom experience. Three games (Bomber, Saucer Math and Lone Ranger Fast Draw) furnish drill in the four basic operations. The other programs (Anti-Aircraft, Sub Commander, Fraction Gunfight, Place Value Tank, and Talking Subtraction) provide drill in place value, equivalents in measurement, fraction identification, and subtraction with regrouping.

Price: $44.95
Additional copies $15.00 ea.
$19.95 for Bomber, Saucer Math and Lone Ranger Fast Draw only

Author: M.C. Henderson III

Available:
Learn-A-Lot Software
711 Ahrens
Houston, TX 77017
(713) 643-2064

---

Name: **Library On-Line Circulation System**
System: Apple II Plus
Memory: 48K
Language: Applesoft and Machine
Hardware: One or two disk drives; "Paper Tiger" printer (optional)

Description: Uses A.B.T. barwand™ and barcoded labels to circulate up to 40,000 titles for 2,000+ borrowers in schools and small libraries. Produces overdue notices; handles unlimited "holds" and "reserves." Programmed by a librarian/programmer and tested in several library environments. Other library software available.

Price: $250.00 in U.S.
Includes manual

Author: Bob Stevens

Available:
Richmond Micro Software
Box 94088
Richmond, B.C.
Canada V6Y 2A2

---

Name: **Dental Insurance Form Writer**
System: Apple II
Memory: 48K RAM
Language: Applesoft, DOS 3.3
Hardware: Firmware Card, disk drive, printer

Description: Dentists can now prepare universal ADA insurance claim forms on a computer. Each form can be prepared (preauthorization), saved to disk, reloaded, edited and then printed (billing) as many times as you desire. *Dental Insurance Form Writer* allows rapid billing and claims submittal with a minimum of effort. User-definable, up to 10 practitioners. Many more features. Can be copied and is user-modifiable.

Price: $100.00
Includes disk, manual, forms

Author: J. McFarland

Available:
Andent, Inc.
1000 North Ave.
Waukegan, IL 60085

---

Name: **Transfer III**
System: Apple III
Memory: Standard
Language: Applesoft
Hardware: Built-in disk drive

Description: *Transfer III* is a new and valuable utility for the Apple III computer. It moves sequential text files either way between an Apple II disk (DOS 3.3) and an Apple III. It can be used, for example, to transfer VisiCalc data files, word-processor text files, BASIC programs converted to text files, and laboratory-data files. All actions required are performed easily and automatically after you select options from menus.

Price: $60
Includes diskette and manual.

Available:
Mind Systems Corporation
P.O. Box 506
Northampton, MA 01061
(413) 586-6463

---

Name: **Spooler**
System: Apple II
Language: Pascal
Hardware: Printer, Pascal Speedup Kit

Description: For use with the Pascal Speedup Kit, *Spooler* allows the user to continue using the entire Pascal system while producing printed reports. Key features: works in any slot with any printer and virtually any printer interface card; automatically picks up user's workfile.

Price: $45.00

Available:
Stellation Two
P.O. Box 2342
Santa Barbara, CA 93120
(805) 966-1140

---

Name: **polyFORTH**
System: Motorola EXORset 30, EXORcisor I & II, Omni-Byte 68000, Intel 8080, 8085, 8086, RCA 1802, LSI-11/02, 23, PDP 11/20-11/70, IBM Series 1, Z-80
Memory: 8K Bytes
Language: FORTH
Hardware: Many disk subsystems, printers, specialized control hardware, etc.

Description: *polyFORTH* from FORTH, Inc. is a multi-tasking, multi-programming environment which includes editor, file handling, virtual memory, language and utilities. It is widely used as a total approach to professional systems development by software and hardware engineers, product designers, educators and scientists. Users of this uniquely powerful technology achieve greatly reduced development time and memory size without sacrificing processor speed or flexibility. *polyFORTH* is the latest and most advanced implementation of the extensible FORTH language developed by Charles H. Moore of FORTH, Inc. in 1972.

Price: $5100-$8200 depending upon configuration
Includes all features, full source and documentation.
Options include: graphics, file system, educational courses, consulting.

Available:
FORTH, Inc.
2309 Pacific Coast Highway
Hermosa Beach, CA 90254
(213) 372-8493

*(Continued on next page)*

## Software Catalog *(continued)*

Name: **Pegasus**
System: Apple II Plus (at present — CP/M, Apple III and IBM PC in future)
Memory: 64K
Language: Apple Pascal
Hardware: Two disk drives (one for program must be 5", other can be 5", 8" or a hard disk)

Description: *Pegasus* was designed to be easy to learn and use by the novice computer user, and also allow the experienced programmer to write applications programs. *Pegasus* is a general data base management program. Input of data can be from the keyboard or from disk files. Output from the system may be to the screen, printer or to disk files.

Price: $199.95
Includes disk (5¼" format at present), manual and update service

Author: Sunil Subbakrishna and J. David Lehman of Shakti Systems, Inc.

Available:
Powersoft, Inc.
P.O. Box 157
Pitman, NJ 08071
(609) 589-5500

---

Name: **PTERM-103**
**PTERM-232**
**PTERM-8010**
**XPTERM-103**
**XPTERM-232**
**XPTERM-8010**
System: Commodore PET and CBM
Memory: 8K for PTERM; 32K for XPTERM
Language: Commodore BASIC and 6502 assembly language
Hardware: CBM disk for XPTERM. Serial interface on telephone modems.

Description: *PTERM-103* provides auto-dial, auto-answer capabilities. Phone numbers are stored and dialed by mnemonic. The baud rate is software selectable (110, 150 or 300 bits per second). *XPTERM-103* allows the system to automatically answer calls from other XPTERM users (103, 232 or 8010 versions). After the remote user provides a password, CBM disk files can be transferred in either direction. *PTERM 232* and *XPTERM 232* work with either the TNW 232D or the TNW-2000, connected directly to the host computer or *via* an accoustical coupler. With the TNW 232D, BREAK can be sent from the keyboard, while the TNW-2000 requires a separate switch modification for BREAK. *PTERM 232* will not support 1200 bits per second operation due to the PET scroll delay.

Price: $19 PTERM-103
$49 XPTERM-103
$19 PTERM-232
$49 XPTERM-232
$29 PTERM-8010
$59 XPTERM-8010
$5 extra for program on 2040 or 8050 disk instead of tape.
PTERM-103 free with TNW 103 modem.
Includes documentation.
Updating provided free of charge for one year.

Available:
TNW Corporation
Dept. MJ
3444 Hancock Street
San Diego, CA 92110
(714) 296-2115
TWX 910-335-1194

---

Name: **SofTeach for UCSD Pascal™**
System: The UCSD p-System™
Memory: 48K runtime environment; 64K development environment
Language: UCSD Pascal
Hardware: 8086, Z-80, 8080, 8085, 6502, 9900, 6809, 68000, and LSI-11/PDP-11

Description: *SofTeach* is a computer-aided instruction package that helps the novice learn to use and understand UCSD Pascal. *SofTeach* is accompanied by *The UCSD Pascal Handbook* which provides a complete description of the concepts of structured programming embodied in the UCSD Pascal.

Price: $125.00
(Prices subject to change)
Includes object code, SofTeach User Manual, and the UCSD Pascal Handbook

Available:
SofTech Microsystems, Inc.
9494 Black Mountain Rd.
San Diego, CA 92126
(714) 578-6105

(Continued on next page)

## Software Catalog *(continued)*

Name: **Piper**
System: VIC-20 and PET/CBM
Memory: 2K
Language: Assembly
Hardware: Standard VIC or PET/CBM with CB2 speaker
Description: This unique program allows you to compose, save, recall and play back music using a standard VIC or a PET with a CB2 speaker. You enter music using alpha notation: A F# C D. Rests and note duration are just as easy. You may vary volume, tempo, print pictures or text and automatically load and play additional compositions from cassette or diskette.

Price: $25

Author: Roy Wainwright

Available:
Abacus Software
P.O. Box 7211
Grand Rapids, MI 49510
(616) 241-5510

---

Name: **Memory II**
System: C4P and C8P
Memory: 8K
Language: BASIC
Hardware: Amplifier
Description: This is just like the game of *Simon Says* with four color blocks on the screen that the computer will turn on randomly. The computer will start off by lighting one color, then two, three, etc. You will have to copy what the computer plays. The longer the sequence the faster it plays. There are three options: color and sound, color only, and sound only. There are four levels of play — the higher the level the longer the sequence!

Price: $7.95
Includes instructions.

Author: Mark A. Dickenson

Available:
Compu-U-Gamer Software
P.O. Box 802
Nevada, MO 64772

---

Name: **The Planetary Guide**
System: Apple II
Memory: 48K
Language: Applesoft
Hardware: DOS 3.3
Description: *The Planetary Guide*, which runs on the Apple II computer, puts the solar system at your finger tips. Graphic displays, along with text data and detailed tables, provide an enjoyable education for all ages. All major solar system members are displayed in detail on hi-res color screens. Moon phases and planetary movement are also animated on hi-res screens. Planet satellites as well as comets and asteroids are also covered. *The Planetary Guide* is menu driven, and single keystroke commands allow rapid access to the many general purpose or detailed programs.

Price: $30.00

Available:
Synergistic Software
5221 120th Ave. S.E.
Bellevue, WA 98006
(206) 226-3216

---

Name: **LexiCom**
System: Apple II or Apple II Plus
Memory: 48K
Language: Applesoft
Description: A word processing utility that allows nearly universal transfer of word processing files. Converts Applewriter to Supertext, Supertext to Applewriter, and either of these may be converted to or created from sequential text files. Allows modem transfer of word processing files, editing of BASIC programs.

Price: $49.95
Includes documentation

Author: David Szetela

Available:
Micro-SPARC Systems Div.
Dept. P
P.O. Box 325
Lincoln, MA 01773

---

Name: **Doctor's Office Companion™**
System: Apple II
Memory: 48K
Language: Applesoft
Description: *Doctor's Office Companion* provides office efficiency through automation of patient billing, completion of insurance forms, accounts receivable ageing, and account history tracking, freeing the doctor's office from billing drudgery and expense.

Price: $995.00 special introductory price
Includes program and complete documentation

Available:
High Technology Software Products, Inc.
P.O. Box 14665
2201 N.E. 63rd St.
Oklahoma City, OK 73113

## Software Catalog *(continued)*

Name: **The Triangulator**
System: Apple II Plus
Memory: 48K
Language: Assembly
Hardware: Disk and Printer
Description: Solves oblique and right triangles and saves results for use in calculation of next triangle. Up to nine results can be saved and used to solve another triangle, or printed in tabular form.
Price: $39.95
Author: David P. Talich
Available:
Arrow Data Systems
1224 E.Harmont
Phoenix, AZ 85020

---

Name: **Property Management System (PMS)**
Memory: 48K
Language: CBASIC
Hardware: Dual diskette drives (220K each), 80×24 screen, 80-column printer
Description: A general ledger system that keeps track of all income (tenants') and expenses providing financial reports and management information necessary to control income properties. This software package includes a checkwriter, budgeting, and exception reports like a list of delinquent tenants. All information and reports are available immediately from the system either on the screen or printed.
Price: $795.00
Includes one year of maintennance
Available:
Realty Automation, Inc.,
221 N. Lois
La Habra, CA 90631
(213) 947-2762

---

Name: **Outpost**
System: OSI
Memory: 8K
Language: Machine Code
Hardware: C1P/Superboard, Cassette
Description: Take the ship movement from *Asteroids*, the swarming aliens from *Invaders*, the rubber, bouncing ship from *Omega System*, and the constantly rotating fortress from *Star Castle*, and you're beginning to get an idea of what this high speed, original arcade game looks like. You have to recover the lost fuel cell and bring it back into your outpost, while dodging blocks, reproducers and the ever-present aliens (up to 20 of them swarming around at once!) *Outpost* can be played by one or two players.
Price: $11.95
Author: Dave Edson
Available:
Pretzelland Software
2005 Whittaker Rd.
Ypsilanti, MI 48197

---

Name: **The Tool™**
System: Apple II
Memory: 48K
Language: Applesoft
Hardware: One disk II disk drive. Optional equipment includes one to four Corvus disk drives or up to eight disk II drives and a printer.
Description: *The Tool* is designed to save programmers 80% to 90% of development time. *The Tool* generates program code... machine language code. It provides an entry screen generator, a database manager, and a report formatter. *The Tool Operating System* (TOS) allows files to span multiple disk drives. Whether you have four 20-megabyte drives or eight floppy drives and 254 diskettes, TOS can use the storage space as one large file or several.
Price: $395.00 special introductory price
Includes program and complete documentation
Available:
High Technology Software Products, Inc.
P.O. Box 14665
2201 N.E. 63rd Street
Oklahoma City, OK 73113

---

Name: **Foosball**
System: Apple II or Apple II Plus
Memory: 48K
Language: Assembly
Hardware: One disk drive
Description: No more dressing up and braving the elements to get your kicks in a smoky, noisy arcade. Now you and up to three of your friends can play *Foosball* at home. Divide into Grud and Robot teams and bash the ball around the screen. Two can play with paddles or up to four can play with Sirius Software's Joyport™ and two sets of paddles.
Price: $29.95
Includes diskette and documentation
Author: Keithen
Available:
Your local computer store

---

Name: **AmperSoft**
System: Apple II or Apple II Plus
Memory: 48K
Language: Applesoft
Hardware: RAM Card - 16K or 32K
Description: Provides four extensions to Applesoft: print using, machine language sort, automatic disk storage and retrieval of arrays, and automatic matrix operations. Moves DOS onto RAM card to provide 10K additional program space. All extensions accessed by simple commands within Applesfot programs.
Price: $49.95
Includes extensive documentation.
Author: Cornelis Bongers
Available:
Micro-SPARC Systems Division, Dept. P
P.O. Box 325
Lincoln, MA 01773

---

Name: **CommuniTree — First Edition**
System: Apple II, Apple II Plus
Memory: 48K
Language: FORTH
Hardware: Up to six disk drives; Hayes Micromodem II
Description: *CommuniTree — First Edition* is a versatile telecommunications software package that can create a wide variety of on-line computer conferencing and videotext facilities. The software makes a highly "intelligent" messaging system — its unique, tree-structured database format puts new messages directly and immediately where they belong. (No lengthy, time-consuming sorts to organize your information after the fact!) By connecting the computer with the telephone lines (*via* modem), the software lets many people share the information. Users of the system can call into the host computer with their own computer or terminal from remote locations. The Turing Test, seven level password protection, and "private" section of the Conference Tree allow the host operator to choose just who uses the system. It can emulate a public bulletin-board-like system, or a private professional exchange between colleagues, or both!
Price: $120..00
Includes manual, program disk, data disk.
Available:
The CommuniTree Group
470 Castro
Suite 207-3002
San Francisco, CA 94114
(415) 474-0933

---

Name: **Print Spooler**
System: The UCSD p-System™
Memory: 48K runtime environment; 64K development environment
Language: Written in UCSD Pascal™
Hardware: 8086, 8080, 8085, Z80, 68000
Description: *The Print Spooler* is a user-executable program which sends one or more text files to a printer while the user continues normal UCSD p-System operations, such as text editing or data entry.
Price: $50.00
(subject to change)
Includes object code for the *Print Spooler.*
Available:
SofTech Microsystems, Inc.
9494 Black Mountain Rd.
San Diego, CA 92126
(714) 578-6105

---

Name: **Capitalization Order #0339AD-C10**
System: Apple II or Apple II Plus
Memory: 32K
Language: BASIC
Description: *Capitalization* is designed for students, writers, reporters — anyone who wants to learn or review the relevant laws quickly and effectively. Each of the 12 fundamental rules is concisely explained on the screen, then examples are given, followed by exercises. The computer keeps score and reports the level of mastery at the end of each set of exercises.
Price: $24.95
Author: Charles Barnes, Robert Large
Available:
Instant Software
Peterborough, NH

## Resource Update

*(Continued from page 100)*

**Softalk**
$18.00 per year, 12 issues
Softalk Publishing, Inc.
11201 Magnolia Blvd.
North Hollywood, CA 91601

**Stems From Apple**
$9.00 per year, 11 issues
$2.00 application fee
Apple Portland Program
Library Exchange
c/o Dick Stein
P.O. Box 1608
Beaverton, OR 97075

**T.A.R.T.**
$15.00 per year, quarterly
The Apple Resource Team
c/o Sid Koerin, Editor
1706 Hanover Ave.
Richmond, VA 23220

**User Magazine**
DM 50.-
Apple User Group Europe e.V.
Hackstuekstr. 11
D-4320 Hattingen 15
West Germany
(Printed in German)

**Washington Apple Pi**
$18.00 per year, 12 issues
P.O. Box 34511
Bethesda, MD 20817

### AIM-RELATED

**Interactive**
$5.00 for 6 issues
Newsletter Editor
Rockwell International
P.O. Box 3669, RC55
Anaheim, CA 92803

**The Target**
$6.00 per year, 6 issues
Donald Clem, Editor
RR#2
Spencerville, OH 45887

### ATARI-RELATED

**A.N.A.L.O.G. Magazine**
$10.00 per year, 6 issues
P.O. Box 23
Worcester, MA 01603

**Atari Computer Enthusiasts**
$10.00 per year, 10 issues
c/o M.R. Dunn
3662 Vine Maple Dr.
Eugene, OR 97405

**The Atari Connection**
$10.00 per year, quarterly
Atari Incorporated
1265 Borregas Ave.
P.O. Box 427
Sunnyvale, CA 94086

**Iridis**
The Code Works
Box 550
5578 Hollister, Suite B
Goleta, CA 93017

### OSI-RELATED

**The Aardvark Journal**
$9.00 per year, 6 issues
Aardvark Technical Services, Ltd.
2352 S. Commerce
Walled Lake, MI 48088

**OSIO Newsletter**
$15.00 per year
c/o Rick Myers
12004 Partillo Rd.
Bowie, MD 20715

**OSI Users Group**
c/o Richard Ellen
12 Bennerley Rd.
London SW11
England

**OSI User's Independent Newsletter**
$10.00 per year, 6 issues
c/o Charles Curley
405 E. 3rd St. #123
Long Beach, CA 90802

**Peek(65)**
$15.00 per year, 12 issues
P.O. Box 347
Owings Mills, MD 21117

### PET-RELATED

**Commodore Magazine**
$15.00 per year, 6 issues
Commodore Business Machines, Inc.
681 Moore Road
King of Prussia, PA 19406

**Commodore PET Users Club Newsletter**
£ 10.00, 5-8 issues, £ 15.00 overseas
Commodore Information Centre
360 Euston Rd.
London NW1
England

**Nieuwegein PET Users Group**
Nijpelsplantsoen 252
3431 SR Nieuwegein
The Netherlands
Attn: Hans Tammer or Louis Konings

**The Paper**
$20.00 per year, 6 issues
c/o Centerbrook Software Designs
Pearl Street
Livingston Manor, NY 12758

**Microcomputer Printout**
$29.00 (air) per year, 12 issues
Stuart House, Perrymount Rd.
Haywards Heath, West Sussex, U.K.

**73 Magazine**
$25.00 per year, 12 issues
P.O. Box 931
Farmingdale, NY 11737

PET Benelux Exchange
$35.00 per year, quarterly (in Dutch)
Copytronics
Burg, Van Suchtelenstraat 46
7413 XP Deventer
The Netherlands

**The Transactor**
$15.00 (Canada) per year, (6-8 issues)
Commodore Systems
3370 Pharmacy Ave.
Agincourt, Ontario M1W 2K4
Canada

**VIC Computing**
$20.00 per year
Paradox Group
39-41 North Road
London N7 9DP,
England

### SYM-RELATED

**Sym-Physis**
$10.00 per year, quarterly
$13.50 per year, overseas
Sym-1 Users' Group
P.O. Box 315
Chico, CA 95927

### TANDY RELATED

**The Rainbow**
$12.00 per year
5803 Timber Ridge Dr.
Prospect, KY 40059

**TRS-80 Microcomputer News**
$12.00 per year, 12 issues
Tandy Corporation
P.O. Box 2910
Forth Worth, TX 76113

**80 Microcomputing**
$25.00 per year, 12 issues
1001001 inc.
80 Pine Street
Peterborough, NH 03458

### NON-COMPUTER MAGAZINES

**EDN (Electronic Design News)**
$25.00 per year, 22 issues
Cahners Publishing Co.
270 St. Paul Street
Denver, CO 80206

**Popular Electronics**
$15.00 per year, 12 issues
One Park Ave.
New York, NY 10016

**QST**
$25.00 per year, 12 issues
American Radio Relay League
225 Main Street
Newington, CT 06111

**Radio-Electronics**
$13.00 per year, 12 issues
200 Park Ave., South
New York, NY 10003

# MICRO™

## Hardware Catalog

Name: **16K/32K RAM board**
System: Atari 400/800
Description: Part #H216, the Mosaic 16/32K RAM, adds 16K to an Atari computer system. Upgrade to 32K is very easy using the $60 upgrade kit #H212. Atari 400 owners can use their existing 16K RAM to upgrade to 32K for $120. The Mosaic 16/32K RAM is of interest to owners of Atari 400 with 16K, Atari 800 with 16K or 32K.
Price: $119.95 — #H216
$69.96 — #H212
Available:
Mosaic Electronics
P.O. Box 748
Oregon City, OR 97045

---

Name: **8-Bit, 8-Channel A/D system**
System: Apple
Language: All
Hardware: Any Apple
Description: The Applied Engineering A/D board is an 8-bit, 8-channel, memory-buffered data acquisition system. Features: 8-bit resolution, on-board memory, fast conversion (.078 ms), A/D process *totally* transparent to Apple (looks like memory).
Price: $129.00
Includes PC board and manual
Available:
Applied Engineering

---

Name: **Inductive Coupled Originate/Answer Modem**
System: General Purpose
Description: Inductive-coupled modem eliminates room noise, vibration caused by acoustic coupling. 0.300 baud, Bell 103-compatible. Orignate/Answer, half/full duplex, RS-232, TTL, CMOS-compatible. Cassette tape ports save data. 110V AC or 9U batteries. Crystal controlled.
Price: $129.95
Available:
MFJ Enterprises, Inc.
921 A Louisville Rd,
Starkville, MS 39759

---

Name: **Communications Cables**
System: All
Description: Full line of RS-232 and RS-422/423 cables, adapters, interconnects for terminal, printer, and modem usage. RS-232 cables to custom lengths, switching boxes, parallel cables upon request. Disk data and power cables.
Available:
Interface Technology of Maryland
P.O. Box 745
College Park, MD 20740
(301) 490-3608

---

Name: **PKASO™ Interface**
System: Apple II, Apple III, and Printers (see below)
Memory: No restrictions
Language: All popular languages including Pascal and CP/M
Hardware: Epson MX-70, MX-80 (Graftrax), MX-100, IDS 560/Prism, Centronics 739, Okidata Microline 80, 82, 83, 82A, 83A
Description: Complete interface between Apple and popular matrix printers. Has built-in firmware for snapshot print of any screen image — graphics or text. When used with a word processor, *Pkaso* adds software-definable symbols for subscripts or math notation, and allows graphics within the text. A gray-scale-processor directly prints computer photography. *Pkaso* commands furnish a simple hardware-independent printer control language.
Price: $165 (U.S. list price)
Includes interface card, cable, demonstration diskette, manual
Available:
Apple Dealers or
Interactive Structures, Inc.
P.O. Box 404
Bala, PA 19004
(215) 667-1713

---

Name: **Microsette**
Description: Diskettes, cassettes for 6502 micros. Diskettes are certified, single-sided, soft-sectored, 5¼", with reinforced hubs.
Price: 10/$25 - diskettes
10/$7.50 - cassettes (C10)
Includes UPS shipping in continental USA.
Available:
Microsette Co.
475 Ellis St.
Mountain View, CA 94043

---

Name: **Single Board Computer — GMS 6506/26/27**
System: 6500/6800/Z80
Memory: 4K bytes RAM, 16K bytes ROM
Hardware: 6" × 9.75" board, EXORciser, System 65-compatible
Description: Single board computers offering interchangeable 6502, 6809, Z80 CPUs. ACIA, IEEE-488, printer ports. Eight I/O lines, two 16-bit timers, two 8-bit shift registers. VUA/VKA, bootstrapping, power on reset, reset, base address and enable/disable switches. GMS 6506 uses 6502 CPU; GMS 6526 uses 6809 CPU, GMS 6527 uses Z80 CPU.
Price: $489
Includes GMS 6506, 6502 CPU, 1 MHz, one-year warranty
Available:
General Micro Systems
1320 Chaffey Ct.
Ontario, Canada 91762

---

Name: **uCortex™/65**
System: 6502 single board computer/controller
Memory: 1K RAM, socket for 2K EPROM (2716) or 4K EPROM (TMS2532 or equivalent)
Language: 6502
Description: 6502 8-bit microprocessor, 16 I/O lines, TTL-compatible, each line independently programmable as input or output, 4 auxiliary control lines, 2 × 16-bit timer/-counters, 2K or 4K EPROM single jumper selectable, on-board 555 power-on reset circuit or external reset signal, non-maskable interrupt control on connector pin, crystal-controlled clock for precise timing applications, pinout-compatible with AIM 65 application connector, timing loops compatible with AIM 65, develop and test program on AIM 65, then transfer EPROM to uCortex™/65.
Price: $79.95
Includes board, assembled and tested (EPROM not included).
Available:
Cortex Research Corporation
1912 Raymond Drive
Northbrook, IL 60062
(312) 480-1088

---

Name: **Video-Printer Stand**
System: For use with TRS-80 Color Computer, Videotex, Atari 400, 800, Apple II or III, and others. Most small to medium sized printers, MX-80, MX-100, etc.
Description: Place a TV (or printer) on the Video-Printer Stand and the keyboard (or fan-fold paper, disk drives, etc.) slides right under. Desk top work space is increased, the screen is raised to eye level, all components are within easy reach, and the keyboard distance from the monitor can be adjusted as desired. The Video-Printer Stand's sturdy metal construction will easily support any portable color TV. Dim.: 19¼" × 12" × 6¾".
Price: $39.95
Includes choice of black or beige.
Available:
Advanced Effort-Saver Products, Inc.
P.O. Box 5001
Hialeah, FL 33014
(305) 821-9961

*(Continued on next page)*

Name: **Datalok**
System: Apple II
Memory: 48K RAM
Language: Applesoft
Hardware: One disk drive, DOS 3.2 or 3.3

Description: *Datalok* provides two basic utilities for the user: the ability to encrypt and decrypt any file created under Apple DOS — i.e., text, integer, binary, Applesoft, etc.; and the ability to lock and unlock an entire disk — i.e., render a disk inaccessible and unbootable to an unauthorized user. Provided also is a diagnostic program enabling the user to verify the system's operation while providing expertise on DES for the inexperienced user.

Price: $349.00

Available:
Atlantis Computers
31-14 Broadway
Astoria, NY 11106
(212) 728-6700

---

Name: **Disk Head Controller (DMC)**
System: OSI Mini-floppy, C1 or C2 or C4
Hardware: Single PCB

Description: This loads and unloads the head, and turns off the drive motor after five seconds (from the last disk access). No tracks to cut; simply unplug your disk connector and connect to the DMC, then plug the DMB connector to your disk drive.

Price: $19-$95 — bare PCB
$69-$95 — fully assembled and tested
Includes PCB, plus full instructions, one year warranty (assembled and tested version).

Available:
G. Cohen
72 Spofforth St.,
Holt, Act, 2615
Australia

---

Name: **Adventure: C1 Sound**
System: OSI C1P Series 1 (without sound)

Description: An inexpensive (and entertaining) data sheet giving simple instructions for adding the components for sound that OSI left out. Cost of the parts is about $1.00, available at any electronics store. For both Superboard II and C1P. Sound adds an unbelievable dimension to your computer. You'll never play another silent game once you've heard it.

Price: $2.75
(Free with purchase of two Pretzelland sound games!)

Available:
Pretzelland Software
2005 Whittaker Rd.
Ypsilanti, MI 48197
(313) 483-7358

---

Name: **Instant ROM, ROM/EPROM Emulators**
System: Any
Hardware: ROM/EPROM Sockets

Description: *Instant ROM* is 2K, 4K, or 8K of CMOS memory with internal battery back-up, in a standard 24-pin ROM/EPROM package. When plugged into a ROM socket, a single connecting lead enables it to be programmed like RAM, at normal system speeds. When the power is switched off the program remains. The internal lithium cell gives typically 10 years life. Uses: custom font character sets for printers and video terminals, speech synthesizer ROMs, long-term data storage in logging systems, utility programs in personal computers, etc. *Instant ROM* is available now in 2516/2716/2532/2732/2364 pinouts. Application notes are available.

Price: $78.00 - 2K
$112.00 - 4K
$178.00 - 8K
Includes application notes, read/write connector with lead.

Available:
LMS Electronics
3401 Monroe Road
Charlotte, NC 28205
(704) 376-7805

---

Name: **ADA 1600**
System: PET/CBM Computers
Language: BASIC
Hardware: Printer adapter

Description: The *ADA 1600* allows PET/CBM computers to use standard centronics-type printers. It has a two-foot cable that plugs into the PET IEEE port. Another IEEE card edge is provided for connecting disks and other peripherals. The address is switch-selectable. The switch selects upper/lower case. A four-foot cable with a standard 36-pin centronics connector is provided.

Price: $129.00

Available:
Connecticut microComputer
36 Del Mar Dr.
Brookfield, CT 06804
(203) 775-4595

---

Name: **PKASO Interface**
System: Apple II and Apple III
Language: Pascal, CP/M, BASIC, Assembler, etc.

Description: Printer interface family for the Apple II and III and Epson, Centronics, Okidata, IDS, NEC printers. Converts your system to a complete text and graphics output system. Everything is included to start printing letters using your favorite text processor program, or to start printing hi-resolution graphics exactly as you see it on the screen. No disk shuffling or program swapping is required to use.

Price: $165.00
Includes interface, cable, comprehensive user manual and demonstration disk.

Available:
Any Apple Dealer
or Factory Direct

---

Name: **AIM 65 Enclosure and Power Supply**
System: Rockwell AIM 65

Description: Brown, textured ABS plastic with an aluminum base. Case comes with power supply mounting brackets as well as circuit breaker and line cord. Optional power supply provides +5 V at 3 amps and +24 V at 0.5 amps.

Price: $165.00 w/p.s., Model A65-006
$95.00 w/o p.s., Model A65-002

Available:
Hamilton-Avnet
and all franchised AIM 65 dealers

---

Name: **Chieftain 5¼" Winchester Series of Computer Systems**
Memory: 64K RAM
Language: BASIC, COBOL, Pascal, FORTH, Assembler

Description: Smoke Signal, manufacturers of the 6809-based Chieftain™ computer systems, have announced a new addition to the existing line of 5¼" and 8" floppy-based, and 8" Winchester-based systems: the new 5¼" Winchesters. Announced are 95W4, 95XW4, 98W15, 9W15T20. The 95W4 and 95XW4 support 360K and 4 Mbyte drives, and 750K and 4 Mbyte drives, respectively. The 98W15 provides one 8" floppy (1 Mbyte) and a 15 Mbyte 5¼" Winchester. The 9W15T20 has a 20 Mbyte Tape Streamer and a 15 Mbyte 5¼" Winchester. All systems run DOS69D and the new UNIX-like OS-9 Levels I and II multi-user, multi-tasking operating system.

Price: $6895 - #95W4
$7195 - #95XW4
$9695 - #98W15
$11195 - #9W15T20

Available:
Smoke Signal
31336 Via Colinas
Westlake Village, CA 91362
(213) 889-9340

---

Name: **Color Port**
System: TRS-80 Color Computer
Memory: 2K of RAM or 2K of EPROM
Language: Any
Hardware: Plug-in Cartridge

Description: Adds powerful I/O capability to TRS-80 Color Computer. Results in a very cost-effective 6809-based control system. Adds two fully programmable 8-bit bidirectional parallel ports with full handshaking. Full interrupt capability is supported, and computer voltage and logic control lines are brought out to the standard edge connector. A socket in the cartridge allows insertion of either 2K of RAM or 2K of EPROM that allows software for the control of I/O to be stored separately from the main user memory space. Provision is made for selection of both autostart of the memory in the cartridge and synchronous reset of the cartridge and the computer.

Price: $129.95 with full instructions plus optional 2K RAM for $19.95, 2K EPROM for $12.95.
Includes hardware cartridge and full description and user manual.

Available:
Maple Leaf Systems
P.O. Box 2190
Station C
Downsview Ontario
Canada, M2N 2S9

*(Continued on next page)*

## Hardware Catalog *(continued)*

Name: **Power Pack**
System: TRS-80 Color Computer
Memory: 16K
Language: Assembly
Hardware: TRS-80

Description: The *Power Pack* allows you access to the 6809 microprocessor inside the Color Computer, and provides additional memory for the more sophisticated programs. It is a cartridge to plug into the expansion slot that contains a powerful monitor plus 6K of RAM. Special versions of such software as the Color Editor, Color Assembler, and Color Pascal are available for the Power Pack, thus eliminating the need for 32K and allowing the monitor to be resident.

Price: $159.95 includes cartridge, diagnostics cassette, instructions

Available:
Computerware
P.O. Box 668
Encinitas, CA 92024
(714) 436-3512

Name: **Time Machine II**
System: Apple II

Description: *Time Machine II* is a one-second to 99-year real time clock. A powerful 2048-byte firmware driver adds many user-friendly options, including READ, FORMAT, INTERRUPT, and SET commands. Included is software DOS DATE-STAMPER to date disk files, a 50-page manual, and battery backup. Optional software includes APPLE SPOOL, an interrupt-driven printer spooler. Listings are buffered in memory then spooled to the printer. Once a file is in the buffer, other programs can be executed.

Price: $139.00 retail
APPLE SPOOL $19.95 (optional)

Available:
Creative Software Dev.
4657 Thayn Drive
West Valley City, UT 84120

Name: **VDISK™**
System: 6809 with FLEX™
Memory: 56K Plus
Language: Machine Language
Hardware: 6809 with extended addressing

Description: *VDISK* allows FLEX users to treat extended memory as a super-fast disk drive. This "virtual" disk drive has its own directory and may contain program and data files. Files may be copied to it and from it. All FLEX utilities and user programs may read from and write to this drive, just as with any other drive. The virtual disk operates much faster than a physical disk, however. The speed advantage apparent to the user will depend on the amount and nature of disk operations being carried out. The time required to load a binary file is negligible.

Price: $99

Author: James Arbuckle

Available:
For dealer list contact:
James Arbuckle
P.O. Box 328
Ambler, PA 19002
(215) 643-0788

Name: **Micromodule™ 17 Monoboard Microcomputer**
System: Any EXORbus ™-based system (M68MM17)
Memory: Up to 40K bytes ROM, RAM, or EPROM
Software: Supported by SUPERbug™ debug monitor and RMS09 executive

Description: This module contains the MC6809 MPU, five sockets for program memory or static RAM, PIA with buffered interface, two ACIAs with RS-323C interface and a tripple programmable counter timer (PTM). Operates at 1 MHz.

Price: $495.00 (1 to 5 quantities) includes user's guide

Available:
Motorola Semiconductor Products, Inc.,
Atten: Microsystems Marketing
P.O. Box 20912
Phoenix, AZ 85036
(602) 244-5714
or Motorola Sales Offices

MICRO™

MICRO™

Dr. William R. Dial
438 Roslyn Avenue
Akron, OH 44320

# 6502 Bibliography

**1. MICRO No. 44 (January, 1982)**

Smith, Wayne D., "Some Help for KIM," pg. 69-72.

Hardware and software for an improved single-step function. Also included is a trace function. Switch-selectable K areas can be provided that will allow the use of a single-step program stored in any of the several K areas of the KIM micro.

**2. POKE-Apple 3, No. 12 (January, 1982)**

Garvey, Michael, "Storing and Reading an Array on Disk," pg. 23-25.

The time to write or read a "binary" file is significantly less than it takes for a "text" file. Here's how to save numeric arrays on the Apple.

**3. POKE-Apple 3, No. 12 (January, 1982)**

Haluza, Doug, "Adding Commands to BASIC with CHRGET," pg. 37-40.

An explanation of the PET command called CHRGET and how to use it. An example utility is given.

**4. Creative Computing 8, No. 1 (January, 1982)**

Ahl, David H. and Lubar, David, "Computer/Videodisk Coupling."

The combination of a videodisk player with a microcomputer affords a novel new area of instructional programs, games, etc. Equipment for implementing an Apple with a Pioneer VP-1000 videodisk player is described. A typical game listing and an explanation of its use is given.

**5. Creative Computing 8, No. 1 (January, 1982)**

Cook, Willis, "Disk Copying with A Single Eight-Inch Drive," pg. 4-6.

Listing of a program for OSI micros C2-4P or C4P to copy without the benefit of two drives. By filling available RAM memory with as much data from a disk as the memory will hold and then dumping to a swapped target disk, a full 73-track disk can be copied in twelve swaps.

**6. Mini'App'Les 5, No. 1 (January, 1982)**

Buchler, Dan, "Apple Plus 68000 equals DTACK Grounded," pg. 8-9.

Interfacing the Apple with the 68000 microprocessor gives your micro the capabilities of a next generation 16-bit system, Speed improvements of 6 to 13 times are claimed.

**7. Atari Computer Enthusiasts (January, 1982)**

Chastain, Ed, "Savmov: Disassembly of Cartridge Programs," pg. 4, 10.

A program for the Atari user interested in assembly-language programming. Two utilities named Saver and Mover to use in examining Atari cartridges. Saver creates a cassette-bootable tape, and Mover is used to relocate the cartridge program in RAM.

**8. Compute! 4, No. 1, Issue 20 (January, 1982)**

Butterfield, Jim, "TINYMON1: A Simple Monitor for the VIC," pg. 176-179.

A tape-loadable monitor for the VIC color computer, honoring all commands of the built-in monitors on other CBM systems.

**9. The Transactor 3, Issue No. 4 (January, 1982)**

Hook, David A. and Ontario, Barrie, "Word Count 9," pg. 28-36.

A utility for PET/CBM systems to count the number of words in a word processor file. For a WordPro file with 2200 words, a conventional BASIC program might require 21 minutes to count the file, while the machine-language routine here counted the same file in 13 seconds.

**10. Dr. Dobb's Journal 7, Issue 1, Number 63 (January, 1982)**

Fusina, Luca and Granuzzo, Claudio, "Interfacing the 68000 to an AIM 65," pg. 12-17, 36-38.

Hardware and software to interface the 16-bit 68000 to the AIM 65.

MICRO™

# Number Conversion

MICRO™

Data Sheet #5

**Least Significant Digit**

**Most Significant Digit**

| HEX | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 1 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 2 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| 3 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| 4 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| 5 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| 6 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| 7 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |
| 8 | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| 9 | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| A | 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 |
| B | 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 |
| C | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 |
| D | 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 |
| E | 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 |
| F | 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 |

| X = | X00 | X000 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 256 | 4096 |
| 2 | 512 | 8192 |
| 3 | 768 | 12288 |
| 4 | 1024 | 16384 |
| 5 | 1280 | 20480 |
| 6 | 1536 | 24576 |
| 7 | 1792 | 28672 |
| 8 | 2048 | 32768 |
| 9 | 2304 | 36864 |
| A | 2560 | 40960 |
| B | 2816 | 45056 |
| C | 3072 | 49152 |
| D | 3328 | 53248 |
| E | 3584 | 57344 |
| F | 3840 | 61440 |

| Binary | Hexadecimal | Decimal |
|---|---|---|
| 0000 | 0 | 0 |
| 0001 | 1 | 1 |
| 0010 | 2 | 2 |
| 0011 | 3 | 3 |
| 0100 | 4 | 4 |
| 0101 | 5 | 5 |
| 0110 | 6 | 6 |
| 0111 | 7 | 7 |
| 1000 | 8 | 8 |
| 1001 | 9 | 9 |
| 1010 | A | 10 |
| 1011 | B | 11 |
| 1100 | C | 12 |
| 1101 | D | 13 |
| 1110 | E | 14 |
| 1111 | F | 15 |

# Number Conversion

**BASIC Hex to Decimal**

```
 10 REM DELETE 70 & 80 FOR PET & OSI
 15 REM AND ENTER 90 AS: Z=Z*16+Y-48+7*(Y>57)
 20 REM X$<=FFFF
 30 INPUT X$
 40 Z=0
 50 FOR J=1 TO 4
 60 Y=ASC(MID$(X$,J,1))
 70 IF Y>57 THEN YY=-1
 80 IF Y<=57 THEN YY=0
 90 Z=Z*16+Y-48+7*YY
100 NEXT J
110 PRINT Z
```

**BASIC Decimal to Hex**

```
10 REM X < 65536
20 INPUT X
30 X=X/4096
40 FOR J=1 TO 4
50 IF X>9 THEN PRINT CHR$(X+55);
60 IF X<=9 THEN PRINT CHR$(X+48);
70 X=(X-INT(X))*16
80 NEXT J
```

**ASCII Character Codes**

Bits 5, 6, 7

Bits 0, 1, 2, 3

| HEX | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|---|
| | BITS | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 0 | 0 0 0 0 | NUL | DLE | SPACE | 0 | @ | P | \ | p |
| 1 | 0 0 0 1 | SOH | DC1 | ! | 1 | A | Q | a | q |
| 2 | 0 0 1 0 | STX | DC2 | '' | 2 | B | R | b | r |
| 3 | 0 0 1 1 | ETX | DC3 | # | 3 | C | S | c | s |
| 4 | 0 1 0 0 | EOT | DC4 | $ | 4 | D | T | d | t |
| 5 | 0 1 0 1 | ENQ | NAK | % | 5 | E | U | e | u |
| 6 | 0 1 1 0 | ACK | SYN | & | 6 | F | V | f | v |
| 7 | 0 1 1 1 | BEL | ETB | ' | 7 | G | W | g | w |
| 8 | 1 0 0 0 | BS | CAN | ( | 8 | H | X | h | x |
| 9 | 1 0 0 1 | HT | EM | ) | 9 | I | Y | i | y |
| A | 1 0 1 0 | LF | SUB | * | : | J | Z | j | z |
| B | 1 0 1 1 | VT | ESC | + | ; | K | [ | k | { |
| C | 1 1 0 0 | FF | FS | , | < | L | / | l | ; |
| D | 1 1 0 1 | CR | GS | - | = | M | ] | m | } |
| E | 1 1 1 0 | SO | RS | . | > | N | ∧ | n | ~ |
| F | 1 1 1 1 | SI | US | / | ? | 0 | __ | o | DEL |

Dr. William R. Dial
438 Roslyn Avenue
Akron, OH 44320

# 6809 Bibliography

**35. Popular Computing 1, No. 2 (December, 1981)**

Firedrake, George and Zamora, Ramon, "My Computer Likes Me," pg. 76-80.
A music program for the 6809-based TRS-80 Color Computer.

**36. BYTE 7, No. 2 (February, 1982)**

Barden, William, "Voice Synthesis for the Color Computer," pg. 258-286.
A hardware and software article for users of the 6809-based TRS-80 Color Computer.

Dubner, Joseph L., "6809 Machine-Code Disassembler," pg. 340-364.
A small and fast disassembler, both reentrant and relocatable, allowing it to be placed anywhere in RAM or ROM.

**37. CSRA Computer Club Newsletter (February, 1982)**

Gresham, Jim, "Color Computer Ramblings," pg. 1.
Start addresses in ROM of the Extended BASIC in the TRS-80 Color Computer based on the 6809 chip.

**38. Creative Computing 8, No. 3 (March, 1982)**

Linzmayer, Owen, "Chromasette Magazine," pg. 36.
Chromasette is a monthly magazine on a 30-minute cassette containing six to eight carefully debugged programs for the TRS-80 Color Computer.

**39. Softalk 2, No. 4 (December, 1981)**

Coats, Douglas E. and Waldman, Cye H., "FORTRAN," pg. 160-172.
In a discussion of FORTRAN for the Apple, some benchmark tests are reported which showed that FORTRAN operating with the 6809-based board, The Mill, outperformed FORTRAN with the native Apple in all respects, and also outperformed the FORTRAN80 (Softcard) in floating-point operations but not in other operations.

**40. TRS-80 Microcomputer News 4, No. 1 (January, 1982)**

Anon., "The More Serious Side of the CC," pg. 42.
Miscellaneous notes on the TRS-80 Color Computer.

Meyers, J.W., "Shoot-Em Again," pg. 43-44.
A game for the TRS Color Computer.

Jamieson, John, "Core Editor," pg. 44.
A simple core editor useful for writing machine language routines and wandering around in the BASIC object code of the 6809-based TRS-80 Color Computer.

**41. 80 Microcomputing Issue No. 27 (February, 1982)**

Wood, James W., "Colorful Titrations," pg. 202-203.
A chemical educational graphics program for the TRS-80 Color Computer.

**42. Microcomputing 6, No. 3, Issue 63 (March, 1982)**

Farnsworth, Dan, "More on the 6809," pg. 170.
Miscellaneous notes on the 6809 including a report that a prime numbers program requiring 58 seconds on an IBM 360 or six hours, 20 minutes on a TRS-80, required only one second on a 2MHZ 6809.

**43. MICRO No. 46 (March, 1982)**

Borgerson, Mark J., "A Disassembler for the 6809," pg. 89-94.
A disassembler written in Microsoft BASIC which will run on either the Apple II (using Applesoft) or on the Radio Shack TRS-80 Color Computer. Includes a description of the operation of the disassembler.

Barden, William, Jr., "Build A Half-year Clock for the Color Computer," pg. 100-122.
A self-contained clock for the TRS-80 Color Computer is described in this hardware article.

**44. 80 Microcomputing Issue No. 22 (October, 1981)**

Barden, William, Jr., "The Assembly Line," pg. 42-49.
A discussion of assembly language for the 6809 as implemented on the TRS-80 Color Computer.

Kitsz, Dennis,"80 Applications," pg. 52-64.
Discussion of the key addresses on the 6809-based TRS-80 Color Computer and Babybug for the Color Computer.

Anon., "Reload 80," pg. 366.
Some notes on programs for the TRS-80 Color Computer.

**45. 80 Microcomputing Issue No. 23 (November, 1981)**

Barden, William, Jr., "The Assembly Line," pg. 50-56.
Machine-language routines in the 6809-based TRS-80 Color Computer.

Kitsz, Dennis, "80 Applications," pg. 78-96.
A discussion of TRS-80 Color Computer expansions available, including Extended Color BASIC, higher speed I/O, memory expansion, etc.

Wood, James W., "Colorful Maneuvers," pg. 328-330.
A game for the TRS-80 Color Computer.

**46. 80 Microcomputing Issue No. 24 (December, 1981)**

Wrye, Charles T., "Color Concentration," pg. 298-300.
A game for the 6809-based TRS-80 Color Computer.

**47. 80 Microcomputing Issue No. 25 (January, 1982)**

Berenbon, Howard, "It's a Big, Big, Colorful World," pg. 72-73.
Discussion of the use of the 6809-based TRS-80 Color Computer in telecommunications.

Kitsz, Dennis, "80 Applications," pg. 342-360.
Hardware article on replacing the keyboard on the 6809-based TRS-80 Color Computer with a TRS Model 1 keyboard.

**48. 80 Microcomputing Issue No. 27 (March, 1982)**

McClenahan, Shawn A., "Where There's A Will...," pg. 84-86.
Hardware and procedure to allow the TRS-80 Color Computer to print while hooked into a telecommunications system.

Ledger, Anthony M., "Color Reversi," pg. 90-101.
A fast moving version of Othello for the 6809-based TRS-80 Color Computer.

Murphy, Brian, "Color Computer Upgrade," pg. 102-105.
How to install additional memory on the TRS-80 Color Computer.

Steiner, John, "Subchaser!", pg. 106-111.
A game with color graphics for the Radio Shack Color Computer.

Esposito, Richard, "Smarten Up, Color Computer," pg. 126-128.
Installing an extra 16K of RAM in the TRS Color Computer.

Green, Roy, "Is A Rose In Color Still A Rose?", pg. 142-150.
Translation of TRS-80 programs for the Color Computer.

McClenahan, Shawn A., "RAM Wars" pg. 156-161.
Hardware article to install 64K of RAM in the TRS-80 Color Computer.

# It's All 1's and 0's

## ACROSS

1. Clear it to add; set it to subtract (6502)
3. 6502 hex op code for "load accumulator, absolute"
5. Substantiate
9. ASCII zero
10. Base 8
11. Opposite of last
13. Manufacturer of Color Computer
15. Branch on result not zero (6502)
16. Increment accumulator A (6809)
17. Manufacturer of 68000 microprocessor
20. Metal used in galvanizing
21. Transfer accumulator to X register (6502)
22. Base 10
25. FOR I = 1 -- 10
26. Hexadecimal E
27. 12*15*13/(192 – 24*8)
28. Integrated circuit
29. It marks your place on the screen
32. The easy way out
34. Something deserved by a question
36. The adjacent side divided by the hypotenuse

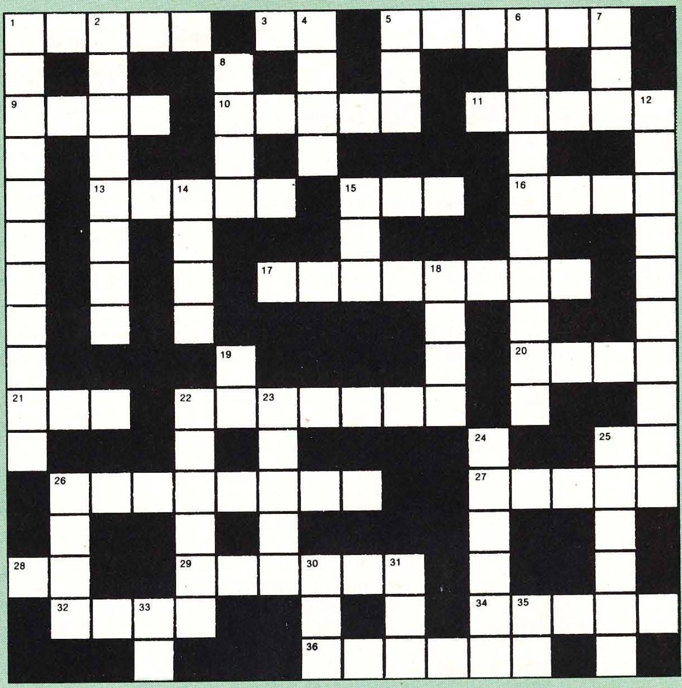
## DOWN

1. To put two strings together
2. Form of addressing for branch instructions
4. Goes with BASIC READ
5. BASIC function to get a number from a character
6. First step when accessing a disk
7. Affirmative
8. Turning the power on is a ----- start
12. PET stands for Personal Electronic -----
14. Goes with BASIC FOR
15. Less than a byte
18. Not imaginary
19. Twos complement of $42
22. Important page for the 6809
23. What CLC does to the carry
24. First key to wear out
25. How the SHIFT LOCK key works
26. Countenance or 64206 in hex
30. Mnemonic for 'subtract with borrow'
31. How to get back from a machine language subroutine
33. Conditional statement
35. 238 in hex

*In the 6502 puzzle in the April "It's All 1's and 0's," the second line of code should read:*

```
6D00 6C FF 6D START JMP (VECTOR)
```

*Thanks to John Krout of Arlington, VA for noticing the error.*



**Please send your**
**unusual observations,**
**puzzles, programming tricks,**
**system photos, etc.**
**to Dr. Bob,**
**MICRO,**
**P.O. Box 6502**
**Chelmsford, MA 01824**

*Answers to the Crossword Puzzle will appear in next month's MICRO.*

MICRO™

201 839-3478

APPLE 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 $139.00

**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 APPLE 48K

AGAIN! Ahead of all others.

**NIBBLES AWAY II**

THE ULTIMATE NEW AND IMPROVED DISK BACKUP SYSTEM

AUTO-LOAD PARAMETERS..
EXPANDED USER MANUAL
TRACK/SECTOR EDITOR
DISK DIAGNOSTICS
PROGRAM MODIFICATION.
FAST & MORE ACCURATE

**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. ERASES THE SOFTWARE BACK-UP BLUES

**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

MasterCard VISA

**MICRO-WARE DIST. INC.**
P.O. BOX 113
POMPTON PLAINS, N.J.
07444

## Advertiser's Index

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

## July Apple Feature

- **Build An Apple Cart —** Even if you live in an apartment and have only a few hand tools, it's possible for you to build a home for your Apple for less than $50.

- **An Overview of Apple DOS —** This article provides general information on the functional blocks of code in DOS. Thus Apple users can manipulate DOS, permit DOS modification, and allow machine language access to DOS commands.

- **Lo-Resolution Graphics for Apple Pascal —** Here's a method to access the Apple's lo-res capabilities from Apple Pascal.

## New Apple Column!

July brings the first installment of our Apple column, written by Apple expert, Tim Osborn. Tim's discussion will cover passing parameters from Applesoft to machine language and back again.

## More July Articles

*Auto Entry for the C1P*
*Removing Frustrating Interference*
*Timing and Counting with the 6522*
*Computer-Assisted Translation of Programs from 6502 to 6809*
*AIM Tape Copy Utility*

### Plus Our Regular Columns and Departments...

*PET Vet*
*The Single Life*
*Reviews in Brief*
*Tech Data Sheet (Apple)*

And more!

## The Rest of '82

Look for our features in upcoming months:

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.<br>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 for a complete 6502-based computer system.

## DRAM PLUS

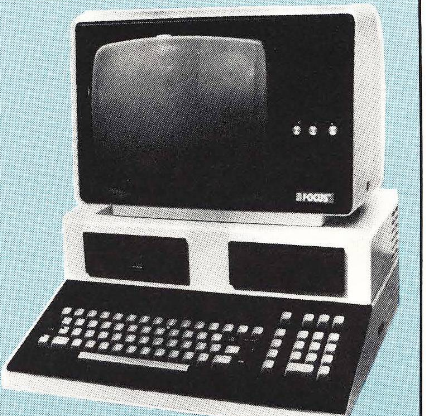
**A multi-purpose expansion board which features:**
- **Up to 40K RAM memory with a memory manager**
- **Up to 16K ROM or EPROM memory**
- **EPROM Programmer for 2516, 2716, 2532 and 2732**
- **Multiple parallel/serial I/O ports and timers/counters**
- **Prototyping area for custom circuits**

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

## Apple®

### ADVANCED OPERATING SYSTEMS

| | | |
|---|---|---|
| #9483 | Apple-Aids | $44.89 |
| #9482 | Mostly Basic/Household | $26.89 |
| #9481 | Mostly Basic/Interface-Scientific | $26.89 |
| #9480 | Voyage of the Valkyrie | $35.89 |

### ARTSCI

| | | |
|---|---|---|
| #6061 | Basic Mailer | $59.49 |
| #6060 | Magic Window | $84.99 |
| #6062 | Magic Words | $62.89 |

### AURORA

| | | |
|---|---|---|
| #4801 | Executive Secretary | $212.49 |
| #4803 | Gradebook | $51.00 |
| #4804 | Hebrew II | $50.99 |
| #4811 | Hebrew II | $79.19 |
| #4805 | Quicktrace | $43.99 |

### AVALON HILL

| | | |
|---|---|---|
| #2058 | Major League Baseball | $25.49 |
| #2052 | North Atlantic Convoy (Cassette) | $12.69 |
| #2060 | Empire of the Overmind (Cassette) | $25.49 |
| #2059 | Empire of the Overmind | $29.69 |
| #2061 | Tanktics | $24.59 |

### BEAGLE BROS.

| | | |
|---|---|---|
| #3257 | Alpha Plot | $34.69 |
| #3250 | Dos Boss 3.3 | $20.39 |
| #3255 | Dos Boss 3.2 | $20.39 |
| #3251 | Game Pack #1 | $20.39 |
| #3252 | Game Pack #2 | $20.39 |
| #3253 | Game Pack #3 | $20.39 |
| #3254 | Game Pack #4 | $20.39 |
| #3258 | Tip Disk | $17.99 |
| #3256 | Utility City | $25.89 |

### BLACKBURG

| | | |
|---|---|---|
| #9160 | Active Filter Design | $35.89 |
| #9162 | Electronics I | $35.89 |
| #9163 | Electronics II | $35.89 |
| #9164 | Electronics III | $35.89 |
| #9165 | Video Display Graphics | $35.89 |

### BRODERBUND

| | | |
|---|---|---|
| #1958 | Alien Rain | $21.19 |
| #1960 | Alien Typhoon | $21.19 |
| #1967 | Arcade Machine | $39.49 |
| #1966 | David's Midnight Magic | $30.69 |
| #1952 | Galactic Empire | $22.99 |
| #1951 | Galaxy Wars | $22.99 |
| #2000 | Genetic Drift | $25.39 |
| #1957 | Golden Mountain | $16.89 |
| #1961 | Payroll | $335.69 |
| #1949 | Red Alert | $22.99 |
| #1965 | Space Quarks | $25.39 |
| #1964 | Space Warrior | $21.19 |
| #1956 | Tank Command | $12.99 |
| #1955 | Tawala's Last Redoubt | $25.39 |
| #1959 | Track Attack | $26.89 |

### HARDWARE

| | | |
|---|---|---|
| #8035 | Apple Clock (Mountain) | $246.39 |
| #8200 | Applesoft Renumber and Merge in ROM (Soft Control) | $33.89 |
| #8206 | Applesoft Utility ROM (Soft Control) | $33.89 |
| #747 | B S R Control (Novation) | $16.69 |
| #2107 | Basic Compiler (Microsoft) | $335.69 |
| #8205 | Basic in ROM (Soft Control) | $33.89 |
| #8039 | Blitz Bug (Omni) | $21.89 |
| #8025 | C P S Pascal (Mountain) | $16.99 |
| #86 | Cable Adaptor (Centronics) | $35.19 |
| #24 | Character Set Plus (Lazer) | $22.39 |
| #6534 | Cassette Control Device (Hartley) | $67.95 |
| #9043 | Data Link (Link) | $87.99 |
| #217 | DISASM/65 (Lazer) | $26.29 |
| #2362 | Dithertizer II with Camera (Computer Station) | $639.99 |
| #8203 | Dual DOS ROMS (Soft Control) | $46.69 |
| #7951 | E Z Port (Versa) | $21.19 |
| #748 | Expansion Mod (Novation) | $33.15 |
| #1476 | Game Paddles (TG) | $33.89 |
| #1070 | Joyport (Sirius) | $67.39 |
| #12 | Joystick (Mimco) | $59.95 |
| #322 | Large Character Eprom (Videx) | $34.29 |
| #27 | Lower Case Adaptor Rev. 7 (Paymar) | $29.99 |
| #9021 | Memory Expansion Module (Prometheus) | $148.69 |
| #8060 | Micro-Verter (ATV) | $31.49 |
| #749 | Modem Handset (Novation) | $24.65 |
| #6240 | Paddle-Adapple (Southern California) | $26.89 |
| #39 | 560G Paper Tiger (IDS) | $1,225.00 |
| #8020 | MX100 Ft. Printer (Epson) | $749.00 |
| #725 | Pro Paddle (Rainbow) | $44.89 |
| #35 | Ram Card (Microsoft) | $169.00 |
| #16 | 16K Ram Expansion Board (Andromeda) | $169.00 |
| #325 | Soft Switch (Videx) | $25.49 |
| #36 | Z-80 Softcard (Microsoft) | $299.00 |
| #860 | Super Fan II (R. H. Electronics) | $64.99 |
| #31 | Switchplate (Videx) | $17.99 |
| #59 | Vision 80 Card (Vista) | $297.49 |

### MICRO LAB

| | | |
|---|---|---|
| #2253 | Crown of Arthain | $29.69 |
| #2250 | Dogfight | $25.99 |
| #2254 | Mad Venture | $21.19 |
| #2259 | Palace in Thunderland | $21.19 |
| #2263 | Roach Hotel | $30.69 |

## Atari®

### ADVENTURE INTERNATIONAL

| | | |
|---|---|---|
| #239 | Angle Worms | $12.64 |
| #235 | Galactic Empire | $16.74 |
| #234 | Lunar Lander | $12.64 |
| #233 | Sunday Golf | $12.64 |

### ARTSCI

| | | |
|---|---|---|
| #204 | Cypher Bowl | $43.94 |
| #203 | Gomoku (Cassette) | $16.94 |
| #201 | Poker Solitaire | $12.74 |
| #202 | Reversi | $16.94 |

### ARTWORX

| | | |
|---|---|---|
| #834 | Giant Slalom | $17.54 |
| #823 | Intruder Alert (Cassette) | $14.84 |
| #822 | Intruder Alert | $18.34 |
| #825 | Rings of the Empire (Cassette) | $14.84 |
| #824 | Rings of the Empire | $18.34 |
| #832 | Stud Poker | $16.64 |

### ATARI

| | | |
|---|---|---|
| #285 | Black Jack | $12.64 |
| #276 | Mailing List | $16.94 |
| #280 | Missile Command | $35.14 |
| #273 | Space Invaders | $16.94 |
| #277 | Star Raiders | $33.94 |

### AVALON HILL

| | | |
|---|---|---|
| #302 | Empire of the Overmind (Cassette) | $25.44 |
| #301 | Empire of the Overmind | $29.64 |
| #308 | North Atlantic Convoy (Cassette) | $12.74 |
| #307 | Nuke War (Cassette) | $12.74 |
| #306 | Planet Miners (Cassette) | $12.74 |
| #304 | Tanktics (Cassette) | $20.34 |
| #311 | Tanktics | $24.64 |

### EDU-WARE

| | | |
|---|---|---|
| #385 | Compu Math/Decimals (Cassette) | $26.34 |
| #384 | Compu Math/Decimals | $35.14 |
| #383 | Compu Math/Fractions (Cassette) | $26.34 |
| #382 | Compu Math/Fractions | $33.94 |
| #381 | Compu-Read | $25.44 |

### EPYX

| | | |
|---|---|---|
| #409 | Crush, Crumble & Chomp (Cassette) | $26.34 |
| #408 | Crush, Crumble & Chomp | $26.34 |
| #407 | Datestones of the Ryn | $17.54 |
| #403 | Invasion Orion (Cassette) | $21.14 |
| #406 | Invasion Orion | $21.94 |
| #401 | Rescue at Rigel (Cassette) | $25.44 |
| #404 | Star Warrior (Cassette) | $33.94 |
| #405 | Star Warrior | $33.94 |
| #412 | Temple of Apshai (Cassette) | $35.14 |
| #400 | Temple of Apshai | $35.14 |

### MISCELLANEOUS

| | | |
|---|---|---|
| #560 | Ali Baba & the 40 Thieves (Quality) | $28.94 |
| #861 | Andromeda (Gebelli) | $35.94 |
| #761 | Casino Blackjack (Manhattan) | $21.94 |
| #763 | Concentration (Manhattan) | $13.14 |
| #466 | Crossfire (On-Line) | $26.94 |
| #746 | Dominoes (Cassette) (Creative Computing) | $10.44 |
| #721 | Galactic Chase (Cassette) (Spectrum) | $21.94 |
| #720 | Galactic Chase (Spectrum) | $26.34 |
| #760 | Gin Rummy 3.0 (Manhattan) | $21.94 |
| #740 | Hail to the Chief (Creative Computing) | $21.94 |
| #921 | Kayos (Cassette) (Computer Magic) | $30.74 |
| #920 | Kayos (Computer Magic) | $30.74 |
| #762 | Labyrinth Run (Manhattan) | $13.14 |
| #860 | Match Racers (Gebelli) | $26.34 |
| #862 | Pathfinder (Gebelli) | $30.74 |
| #748 | Pool (Cassette) (Creative Computing) | $10.44 |
| #322 | Safari (Cassette) (CDS) | $25.44 |
| #321 | Safari (CDS) | $33.94 |
| #1000 | Shadow Hawk One (Horizon) | $44.94 |
| #900 | Shattered Alliance (Strategic Simulations) | $35.14 |
| #567 | Tank Trap (Cassette) (Quality) | $10.14 |
| #568 | Tank Trap (Quality) | $12.64 |
| #561 | Tari Trek (Cassette) (Quality) | $10.14 |
| #562 | Tari Trek (Quality) | $12.64 |
| #461 | The Wizard & The Princess (On-Line) | $27.94 |
| #800 | Yahtman (Compu-Core) | $17.54 |

**COMING SOON!**

**Hodge Podge**
**for the Atari®**

## Apple®

### MISCELLANEOUS

| | | |
|---|---|---|
| #2182 | Adventure in Time (Phoenix) | $25.39 |
| #2301 | Apple World (USA) | $52.99 |
| #7890 | Applecillin (XPS) | $33.89 |
| #3700 | Asteroid Field (Cavalier) | $22.49 |
| #2700 | Astroscope (AGS) | $26.49 |
| #2180 | Birth of the Phoenix (Phoenix) | $35.89 |
| #2407 | Bishop's Square (Datasoft) | $25.39 |
| #1601 | Bloody Murder (Stoneware) | $17.99 |
| #3000 | Bookkeeper (Delta) | $76.39 |
| #3702 | Bug Attack (Cavalier) | $26.29 |
| #9700 | Castles of Darkness (Logical) | $30.69 |
| #3001 | Checkwriter (Delta) | $33.99 |
| #9860 | Chinese Lessons (CTI) | $26.09 |
| #7851 | Cosmo-Mission (Cosmos) | $21.19 |
| #9241 | The Count (Pear) | $22.39 |
| #2752 | Creative Financing (Howardsoft) | $127.49 |
| #2205 | Creature Venture (Highland) | $21.19 |
| #4401 | Cyborg (Sentient) | $28.99 |
| #1607 | D B Master Version 3 (Stoneware) | $194.59 |
| #4250 | Falcons (Piccadilly) | $25.39 |
| #7878 | Financial Partner (Denver) | $220.49 |
| #6080 | Firebird (Gebelli) | $25.39 |
| #3954 | Form Letter (S. Sorcery) | $26.99 |
| #7852 | Hungry Boy (Cosmos) | $21.19 |
| #3451 | Interlude (Syntonic) | $16.99 |
| #9840 | Jabbertalky (Mind Toys) | $26.29 |
| #3150 | The Landlord (Min Micro) | $649.99 |
| #9242 | The Liberator (Pear) | $26.89 |
| #2406 | Lisp Interpreter (Datasoft) | $106.19 |
| #3953 | Mailing List (S. Sorcery) | $44.89 |
| #9240 | The Manipulator (Pear) | $31.39 |
| #4552 | Market Analyzer (RTR Software) | $224.99 |
| #4550 | Market Charter (RTR Software) | $110.39 |
| #9100 | Memory Page Editor (Compu-tron) | $44.89 |
| #9580 | The Menu (C & H Video) | $26.29 |
| #9380 | Menu Generator (Crane) | $35.09 |
| #1804 | Monty Plays Monopoly (Visicorp) | $31.99 |
| #2211 | Mummy's Curse (Highland) | $26.39 |
| #2200 | Oldorf's Revenge (Highland) | $16.89 |
| #4400 | OO-Topos (Sentient) | $27.99 |
| #7870 | Pascal Programmer (Denver) | $105.99 |
| #7871 | Pascal Tutor (Denver) | $105.99 |
| #7650 | Pornopoly (CCI) | $25.39 |
| #3484 | Pot of Gold I (Rainbow) | $33.89 |
| #3485 | Pot of Gold II (Rainbow) | $33.89 |
| #2751 | Real Estate Analyzer (Howardsoft) | $127.49 |
| #6600 | Rubik's Cube (Software Alternative) | $16.89 |
| #6081 | Russki Duck (Gebelli) | $31.39 |
| #1150 | Sargon II (Hayden) | $29.79 |
| #2701 | Sexoscope (AGS) | $26.49 |
| #8901 | Snack Attack (Datamost) | $25.39 |
| #8520 | Solitaire (Computek) | $25.39 |
| #2302 | Space Raiders (USA) | $25.39 |
| #2306 | Star Dance (USA) | $21.89 |
| #3701 | Star Thief (Cavalier) | $25.39 |
| #7163 | Star Venture (Compu-Thing) | $26.89 |
| #9180 | Starship Commander (Voyager) | $35.89 |
| #4251 | Suicide (Piccadilly) | $26.29 |
| #3955 | Text Editor (S. Sorcery) | $58.39 |
| #8901 | Thief (Datamost) | $25.39 |
| #7853 | Thunderbirds - GX (Cosmos) | $21.19 |
| #7102 | Trickshot (IDSI) | $35.09 |
| #1606 | Utility Pak #1 (Stoneware) | $21.19 |
| #4100 | Visicaid (Data S. C.) | $29.69 |
| #1820 | VisiCalc 3.3 (Visicorp) | $199.00 |
| #1805 | VisiFile (Visicorp) | $212.49 |
| #9740 | Vocanoes (Earthware) | $43.49 |
| #2181 | Zoom Grafix (Phoenix) | $35.89 |
| #7880 | Zork I (Infocom) | $33.89 |
| #7881 | Zork II (Infocom) | $33.99 |

### SIRIUS

| | | |
|---|---|---|
| #1058 | Autobahn | $25.39 |
| #1068 | Beer Run | $19.99 |
| #1073 | Borg | $26.89 |
| #1074 | Computer Foosball | $25.39 |
| #1065 | Copts & Robbers | $29.69 |
| #1051 | Cyber Strike | $33.89 |
| #1067 | Dark Forest | $19.99 |
| #1060 | Gamma Goblins | $25.39 |
| #1061 | Gorgon | $33.89 |
| #1069 | Hadron | $29.69 |
| #1057 | Pulsar II | $25.39 |
| #1071 | Snake Byte | $25.39 |
| #1064 | Sneakers | $25.39 |
| #1056 | Space Eggs | $19.99 |
| #1050 | Star Cruiser | $20.99 |
| #1072 | Twerps | $26.89 |

### SYNERGISTIC

| | | |
|---|---|---|
| #1220 | Adventure to Atlantis | $35.99 |
| #1216 | Data Reporter | $186.99 |
| #1202 | Dungeon Campaign | $14.79 |
| #1210 | Escape from Arcturus | $25.39 |
| #1214 | The Linguist | $33.99 |
| #1200 | Odyssey | $25.39 |
| #1207 | Program Line Editor | $33.99 |
| #1201 | Wilderness Campaign | $18.49 |

**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 (excluding Canada): remit U.S. currency, checks on U.S. banks, 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.