Computer-Aided Instruction: Boon Or Bust

The 6502 Resource Magazine
PET • Apple • Atari • OSI • KIM • SYM • AIM

Running 40 Column Programs On A CBM 8032

COMPUTE!

$2.50
May, 1981
Issue 12
Vol. 3, No. 5
63379

The Journal For Progressive Computing™

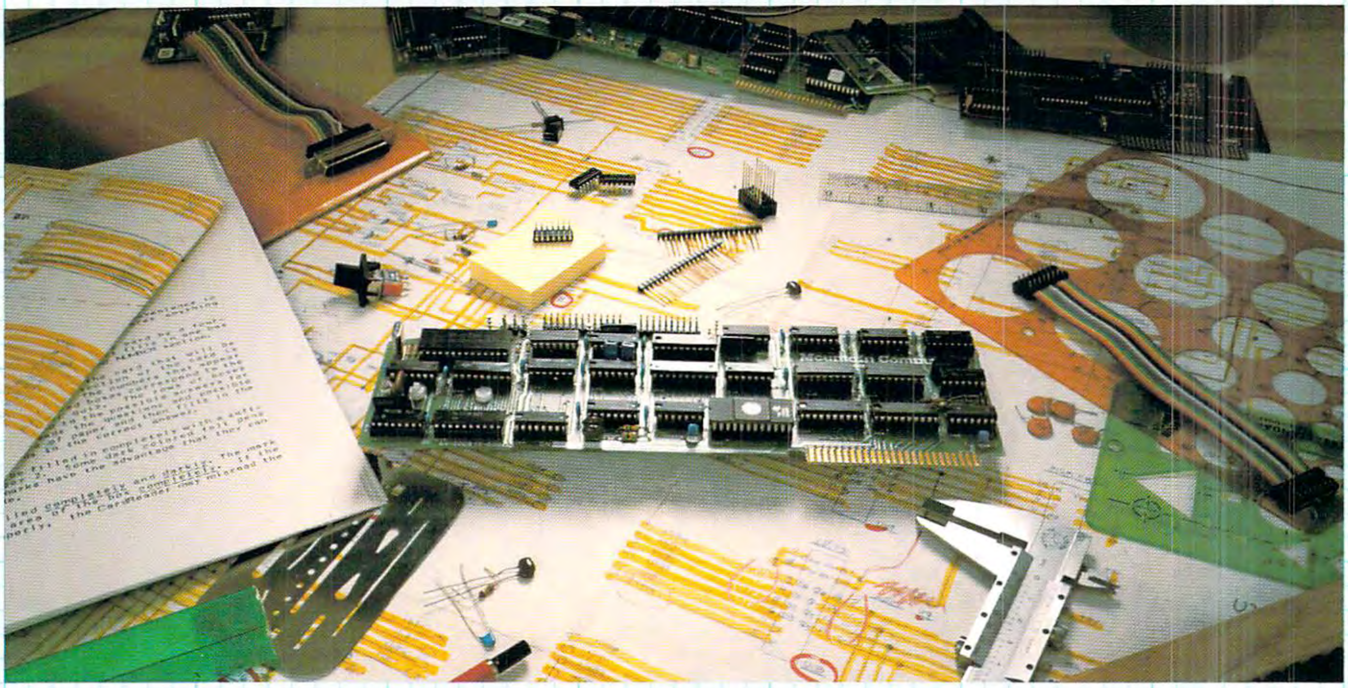
PET As An IEEE-488 Logic Analyzer

Using Strings For Graphics Storage On The Atari

Using Named GOSUB And GOTO Statements In Applesoft BASIC

Using The 6522 To Drive A Printer

Mountain Computer put it all together for you.



The CPS MultiFunction Card

Three cards in one! The Mountain Computer CPS MultiFunction Card provides all the capabilities of a serial interface, parallel output interface and real-time clock/calendar—all on one card—occupying only one slot in your Apple II®. Serial and Parallel output may be used simultaneously from CPS. CPS is configured from a set-up program on diskette which sets the parameters (such as baud rate, etc.) for all functions contained on the card and is stored in CMOS RAM on the card. Once you have configured your card, you need never set it up again. You may also change parameters from the keyboard with control commands. All function set-ups stored on-board are battery powered for up to two years. "Phantom slot" capability permits assigning each of the functions of CPS to different slots in your Apple without the card actually being in those slots! For example, insert CPS in slot #4 and set it up so that is simulates a parallel interface in slot #1 and a clock in slot #7 and leave the serial port assigned to slot #4. CPS's on-board intelligence lets it function in a wide variety of configurations, thereby providing software compatibility with most existing programs. "We've put it all together for you"—for these reasons and many more! *Drop by your Apple dealer and see for yourself how our CPS MultiFunction Card can expand the capabilities of your Apple and save you a great deal of money as well!*

Calendar/Clock
- One second to 99 years
- Battery backed-up (2 years)
- Two AA standard alkaline batteries for back-up (provided)
- Compatible with MCI Apple Clock™ time access programs

Parallel Output
- Features auto-line feed, Apple tabbing, line length, delay after carriage return, lower to upper case conversion
- Centronics standard—reconfigurable to other standards
- Status bit handshaking

Serial Interface
- Features auto-line feed, transparent terminal mode, Apple tabbing, line length, delay after carriage return, local echo of output characters, simultaneous serial/parallel output, lower to upper case conversion, discarding of extraneous LFs from serial input
- Uses the powerful 2651 serial PCI chip
- 16 selectable internal baud rates—50 to 19.2Kbaud
- Half/Full duplex terminal operation
- I/O interface conforms to RS-232C
- Asynchronous/Synchronous operation

Mountain Computer
INCORPORATED

300 El Pueblo Scotts Valley, CA 95066
(408) 438-6650 TWX: 910 598-4504

™Apple Clock was the trademark of Mountain Computer Inc.

®Apple and Apple II are registered trademarks of Apple Computer Inc.

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

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

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

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

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

More languages. With SoftCard and CP/M, you can add Microsoft's ANSI Standard COBOL, and FORTRAN, or Basic Compiler and Assembly Language Development System. All, more powerful tools for your Apple.

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

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

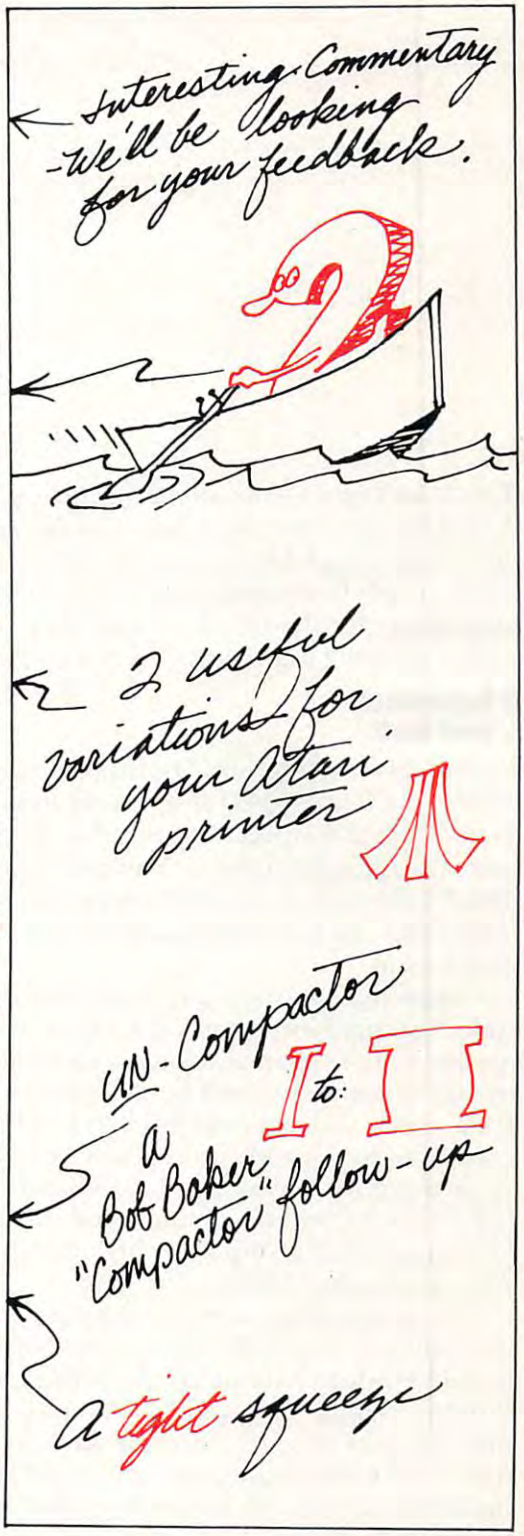
SoftCard is a trademark of Microsoft. Apple II and Apple II Plus are registered trademarks of Apple Computer. Z-80 is a registered trademark of Zilog, Inc. CP/M is a registered trademark of Digital Research, Inc.

MICROSOFT
CONSUMER PRODUCTS

Microsoft Consumer Products, 400 108th Ave. N.E.,
Bellevue, WA 98004. (206) 454-1315

Table of Contents

May, 1981, Vol. 3, No. 5



COMPUTE! The Journal for Progressive Computing (USPS: 537250) is published 12 times each year by Small System Services, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 200 East Bessemer Ave., Greensboro, NC 27401.

Domestic Subscriptions: 12 issues, $20.00. Send subscription orders or change of address (P.O. Form 3579) to Circulation Dept., **COMPUTE!** Magazine, P.O. Box 5406, Greensboro, NC 27403. Controlled circulation postage paid at Greensboro, NC 27403. Application to mail at controlled circulation rates pending at Hickory, NC 28601. ISSN 0194-357X.

A Small System Services, Inc. Publication

COMPUTE! The Journal for Progressive Computing

The Editor's notes...

Robert Lock,
Editor/Publisher

The West Coast Computer Faire was exceptional. A real joy. Do you realize how fast this industry of ours is growing? And I mean growing in terms of more people becoming interested in what we've been doing for the past few months or years, as well as growing in breadth. Here's a sample:

It Talks Back ... And Well

Votrax (500 Stephenson Highway, Troy, MI 48084, (313) 588-0341) showed off their "Type-'N-Talk™" a text to speech synthesizer that produces quite recognizable speech. You interface "Type-'N-Talk™" through an RS-232C interface, type English text with a talk command, and your computer talks back to you.

Now you should understand that this isn't a speech recognition device. It's a speech output device. It more than adequately constructs verbal strings of text from your keyed input in programs. It's just that you can't talk back to it. The company expects to have production quantities available in June. Suggested retail price is $345.00. Watch for a full review by Susan Semancik and our Delmarva Computer Club group in an upcoming *Micros With The Handicapped* column.

A second interesting product at the show was the Osborne 1, a (Z-80) based portable computer utilizing industry standard technology in a clever fashion. Designed as a portable, hand carriable unit, it meets its specs. Primary attractions, beyond that, are its price and some innovative software bundling. At a $1795 retail price, the Osborne 1 has these features:

- 64K, Z80A
- Standard Business Keyboard
- A 5" CRT with CLEAR resolution
- Serial and IEEE 488 interfaces
- Dual "100K" minifloppies
- Weatherproof carrying case

The interesting break is the software bundling — the $1795 price includes:

- Wordstar word processing with Mailmeye option
- The CP/M disk operating system
- CBASIC and MBASIC languages
- The Supercalc electronic calculator

Additional hardware options will be offered. I think if you're on the market for such a machine, this'll be a good place to start looking. As always, not the only place, but the concept of bundling of software is certainly attractive.

Introducing "Super-PET"

Commodore has made what appears to be a breakthrough of major significance for the industry. The machine's true name is unknown at press time. It has been variously called; the "Mini-Frame", the "Micro Mini-Frame", the "Mini Main-Frame", and the "Micro Main-Frame". *(We would have been happy to sponsor a "Name the Super-PET" contest.)*

We received much of this information in a March 3 interview, but held off because of on-going "delicate negotiations". These apparently over, "Super-PET" was introduced at the Hanover Faire in Germany during the first week of April.

How super is it? Here are the specifications:

- 134K Mixed RAM and ROM allocated as follows:
  - 18K ROM Operating System for the 6502 processor
  - 18K ROM Operating System for the 6809 processor
  - 2K Screen RAM
  - 32K "normal" CBM 8032 RAM
  - 64K Bank Switched RAM operating as virtual memory.

1 RS-232C fully programmable serial port
1 High-speed serial communications port for networking at 200KB

- Languages:
  - Waterloo Extended BASIC.

Some of the highlights of this BASIC include unlimited length strings, name called subroutines with parameter passing, local and global variables, program chaining, and total variable preservation. (Meaning you can correct some types of errors in a

Subscription Information

(12 Issue Year):
COMPUTE! Circulation Dept.
P.O. Box 5406
Greensboro, NC 27403 USA
U.S. $20.00
Canada $25.00 (U.S. funds)
Europe: Surface Subscription, $25.00 (U.S. funds) if ordered direct, or available in local currency from the following distributors:

United Kingdom
Contact L. P. Enterprises,
8-11 Cambridge House
Cambridge Road
Barking, Essex
England IG1 18NT

Germany,
Switzerland,
Austria,
Contact Ing. W. Hofacker GMBH
8 Munchen 75
Postfach 437
West Germany

Canadian Retail Dealers should contact:
Micron Distributing
409 Queen Street West
Toronto, Ontario M5V 2A5
(416) 361-0609

running program, type continue and resume the program.)

– PASCAL The Jensen-Worth Standard implementation
–FORTRAN Waterloo Standard version
– APL
– COBOL (later in the spring)
– An Assembler that's supposed to be quite powerful

This entire package plugs into the standard CBM 8032. You plug it in and go with a switch on the side to select your processor mode.

The "delicate negotiations" were necessitated by the fact that all of this expansion power was developed outside of Commodore. Bill McLean and crew at BMB Compuscience in Canada were responsible for developing the hardware, and Waterloo University in Canada, developed the software. Commodore will be marketing the product worldwide. My thanks to Dr. Frank Winter at Sheraton College for his help in putting this all together.

The unit will be introduced in the US at the NCC beginning May 4. Given the configuration of hardware and software, it certainly looks as if we're looking at a potentially viable entry into the small business market of the Apple III and others. We have no confirmation of the upgrade price, but the reliable rumors suggest the expansion will cost much less than the current retail 8032 price of $1795.00.

Well Dr. Chip, it looks like **COMPUTE!** will be covering the 6809 before too long.

News From The Atari Front

Atari has announced a major software development and support project. See the new products section for more information. Axlon has announced a 256K memory system for the Atari 800. The unit provides eight expansion memory slots, allows bank selection of memory, and comes with memory management software. For more information, they're at 170 Wolfe Road, Sunnyvale, CA 94086. (408) 730-0216.

At the West Coast Faire, Atari interest was quite strong. Macrotronics, showing off their screen printer package. (Atari to Trendcom 200 or Paper Tiger) was quite busy. Atari corporate, though not exhibiting, had a private preview for user group officers. Among other things they showed off the new word processor and I heard excellent reports on it.

That's A Switch, PET

Data Equipment Supply was demonstrating a new ROM switching device at the show, and at least two companies (one, Canadian and one, English) have now announced versions of "soft" ROM — PET or CBM RAM expansion boards or chips that can retain information. In a future issue, we'll have some enlightenment on the situation, furnished by Jim Butterfield. ©

The Readers' Feedback

Robert Lock and Readers

It's nice to be back. First of all, we're hoping to have *Ask The Readers* up and running by next month. That's our new three-way column that serves as an interface between programmers with problems and readers with solutions. *The Beginner's Page* returns next month.

On this positive note, let's get started:

"Thanks for:
1. Putting the magazine into envelopes again.
2. Ask the Readers. I will answer.
3. A magazine that gets better each month."

Thanks for the boost. For you cynics saying it may be better but it also gets later, we know. We've expanded our production staff, and brought all typesetting and camera work in-house. Frankly, we've been growing so fast we had to do a little catch-up. This is our 12th issue, and we've almost tripled in circulation in the last 6 or 7 months of our 20 month history!

"Keep up the good work ... need some good small business programs for the Atari (Payroll, taxes, investments, etc.)."

"Article on stock market, financial news software. Is there software available that allows user to create own daily bar charts from Dow Jones News Service quotations?"

Okay, you reader/authors. Anyone willing to share their business investment programs in articles. We're all interested.

"The main reason I buy **COMPUTE!** *is the strength of its articles on ATARI. Better distributorship in the Orange County, CA area is needed."*

We're certainly interested in developing better distribution. If your local dealer sells out of **COMPUTE!** in two days (which many do), suggest they order more. If your local dealer doesn't yet carry **COMPUTE!**, ask them to give us a call.

Until next time ... keep those cards and letters coming.

Thanks,
Robert Lock

©

Shortcuts to more Commodore capability

4 amazing new devices to make your Commodore friendlier, faster and more productive than ever

If you have a Commodore computer, life at the keyboard just got easier, thanks to four new products from Kansas City Computers, Inc.

THE SILENT SWITCH

If you use a switching device on your ROM based software, you may have experienced —select line transient problems.

Introducing our **Triple Flip** firmware. It's a remarkably simple switch that can be electronically *mechanically* switched. With the external switch running to internal switching, there are no address lines outside the computer. That nips select line transient problems right at the source.

Triple Flip lets you place three individual ROMS in a single ROM socket. A single switch can control up to seven **Triple Flips.** That gives you 21 ROM capability.

Versatile? **Triple Flip** is one-socket ROM selectable for Commodore computers 2716, 2532 and 2732. It also fits Pet 2001B, N and later models.

And like all KCCI products, it's backed by our full guarantee and 90-day warranty.

Don't change chips everytime you switch to another program. And don't risk select line transient problems. Put **Triple Flip** to work for you for just $39.95.

30 NEW COMMANDS

Until now, you've had to punch the keys 24 times or more to put some commands into action. Commands like transferring programs to disk or paper, or appending program modules.

Order our **UtiliRom** and start using your computer's memory instead of your own. **UtiliRom** includes an enhanced DOS universal wedge that lets you make most of your present commands in two keystrokes. Imagine having all this at your command:

- Wild card directory (DOS)
- Rename a file (DOS)
- Wild card file scratch (DOS)
- Copy files on disk (DOS)
- Reset disk system (DOS)
- Initialize 2000 series drives (DOS)
- Check error channel (DOS)
- New a disk (DOS)
- Validate a disk (DOS)
- Scroll down
- System cold start
- One key command to load a program (DOS)
- Send program listing to printer (with* or without* form feed at end)
- Send screen contents to printer (normal mode* or squeezed*)
- Send screen contents to disk file by any name*
- Disk program append*
- Repeat key function*
- Kill to turn off repeat*
- Escape to turn off ROM*
- Convert hex to decimal or
- Convert decimal to hex (with error detection)
- Fast jump to monitor
- Fast shift to upper or lower case
- Fast jump to cold start
- One key command to save a program
- Beep (programmable)*

**Asterisk indicates routines which can be called in basic as subroutines for increased computer power.*

Warning! Extensive testing has proven that even casual users of **UtiliRom** become hooked on its efficiency and speed. Serious programmers and custom software designers appreciate the fact that routines are included for fast disk I/O — up to 3 times faster than the system routines. In addition, **UtiliRom** can help protect software designs from theft.

UtiliRom firmware is available for any socket on any Commodore system (including the Address 9000). When used with **Chipmate** firmware (below), you can use UtiliRom in the same socket simultaneously with one other 2K ROM such as Toolkit™.

Order **UtiliRom** now at just $49.95.

Order **Chipmate firmware with UtiliRom** for same socket simultaneous use with most other 2K ROMs. **Chipmate with UtiliRom** is only $61.90. **Chipmate alone** is $14.95.

COMPUTERIZED SCRATCH PAD

Dream up anything you want and put it on your screen. Punch a couple of keys and put it on disk in less than two seconds. Retrieve it anytime, almost as fast as you can visualize it. Hit two more keys and it's generated on your printer in page-perfect format.

You're not dreaming. You're using **Screen Pro** software, the new disk based system that adds exciting new powers to your computer — and a faster response than ever before.

From compiling grocery lists to editing catalogs, from diagramming football plays to animating them, **Screen Pro** makes text and diagram work incredibly simple.

Screen Pro's entire 2000 character screen is displayed in a fraction of a second. You can easily place this subroutine into any software you write, for quick access later. In fact, an eight line program gives you access to any one of 10,000 pages of information — all within four screens!

Clear, understandable documentation shows you how to tailor **Screen Pro** to your exact needs. And it opens your eyes to uses limited only by your imagination — for only $39.95.

MONEY BACK GUARANTEE

All four products come to you with complete, easy-to-follow documentation. Installation instructions that are a snap. Plus step-by-step examples that walk you through the process.

You get more than documentation with your KCCI order. You get dependability. Dependability backed by our Money Back Guarantee. Plus our full 90-day Service Warranty (that's three times as long as most of our competitors' warranties).

Start investing less time in your computer — and start getting more out of it. Fill out and mail this coupon with your check today.

Dealer inquiries welcome — Phone us right now for full information, (816) 356-6502.

Toolkit™ is a trademark of Palo Alto ICS Inc.
™Commodore Business Machines

KCCI

Kansas City Computers, Inc.
5214 Blue Ridge Boulevard
Kansas City, Missouri 64133 • Phone 816/356-6502

Name ______
Address ______
City ______ State ______ Zip ______
Computer make ______ Model ______
Disk system make ______ Model ______

YES! Please send me these shortcuts for more computer capability. My check is enclosed.

☐ TRIPLE FLIP	$39.95	☐ CHIPMATE WITHOUT UTILIROM	$14.95
☐ UTILIROM	$49.95	☐ SCREEN PRO	$39.95
☐ CHIPMATE WITH UTILIROM	$61.90	☐ TOTAL ORDER	

(Add $1.50 shipping & insurance for U.S. orders. Add $3.10 shipping for foreign orders — payment must be in U.S. funds. In Missouri, add 4% sales tax.)

BYTEWRITER-1

Price Breakthrough from

MICROTEK inc.

The World's First Under $300 80 Column Dot Matrix Printer

WHY DID IT HAPPEN?

We were approached recently by a man with an idea. He is one of the leading American manufacturers of dot matrix print mechanisms. He had observed, as we had, that in recent months certain foreign printer manufacturers were increasing their share of the low-cost printer market at an alarming rate.

He thought there MUST be a way to fight back. And he was right. We've since formed a working alliance with this manufacturer, and have brought our first joint offering to the market.

HOW DID WE MAKE IT POSSIBLE?

We set out to combine his proven low cost print mechanism with the simplest possible control electronics. Advances in single-chip microprocessor technology and price erosion of components during the last year helped to make this long awaited dream come true - a printer that can be sold for less than half the cost of the computer that drives it. A $299 printer.

But cost-effective designs and efficient manufacturing operations weren't enough. Computer retailers can make up to a $250 markup on the foreign models. Could we hold to a $299 list price and give the dealer enough incentive to sell the Bytewriter-1? No way. We had to try a more direct approach.

YOUR BUY DECISION - DEALER OR MAIL ORDER

There are some very good reasons to buy your first computer through a dealer. There is a certain amount of hand-holding required when you decide to buy a personal computer. This is one of the main functions of the retail computer store. And most of them perform this function very well.

But why would anyone want to buy add-on equipment through a dealer? If you find a product that has been designed for and tested with your particular computer, you can safely bypass the computer dealer. You can have the best of both worlds. You can save money by buying direct from the manufacturer, and you can be certain that your new device will work when you get it.

We've done extensive testing with the most popular computers - the TRS-80, the Apple II, and the Atari 400 and 800. If you own one of these computers, we guarantee you won't have any interface problems with the Bytewriter-1.

TRS-80 is a trademark of Radio Shack, Div. of Tandy Corp.
Apple II is a trademark of Apple Computer, Inc.
Atari 400 & 800 are trademarks of Atari, Inc.
Bytewriter-1 is a trademark of Microtek, Inc.

FOUR THINGS YOU SHOULD KNOW BEFORE YOU BUY THIS PRINTER

We don't want any unhappy customers. We'd like you to know the limitations of our printer, as well as its advantages. There are some differences between the Bytewriter-1 and the higher priced printers you may be looking at:

1) The Bytewriter-1 takes single sheet and roll paper only. No pin feed paper.
2) We've used a 7-wire print head. No fancy lower case descenders.
3) There aren't any software frills in the Bytewriter-1, like VFU controls. However, if your main interest is getting software listings or printing letters, you won't care. And, with a bit of ingenuity, you can provide VFU functions in your own programs.
4) You can't go into a computer store and pick up a Bytewriter-1. They're sold direct only by MICROTEK.

We realize it's unusual to point out the limitations of a product in an ad that promotes it, but we think it's important for mail order buyers to fully understand what they're buying.

The Bytewriter-1 will fill the needs of most people. People who don't see the sense in spending extra money for features they'll never use.

ORDER THE CONFIGURATION THAT'S JUST RIGHT FOR YOU

The Bytewriter-1 is available with an interface cable and complete instructions for use with three of the most popular small computers on the market today, the Apple II, the Atari 400/800, and TRS-80 Models I, II, and III. One of our divisions, MICROTEK PERIPHERALS CORP., can even provide you with the expansion card or module that your computer may require to drive a printer.

CALL OR WRITE TODAY FOR MORE INFORMATION.

We have people standing by to answer your questions.

MICROTEK inc.

Microtek, Inc.
9514 Chesapeake Drive
San Diego, CA 92123
Tel. 714-278-0633
Outside Calif. call toll free: 800-854-1081
TWX. 910-335-1269

Computers And Society

David D. Thornburg
Los Altos, CA

Several years ago, when Betty Burr and I were conducting workshops for "computer-phobic" adults, we thought that someone should write a "computer demystification" book which would sell to general audiences. Since that time we have seen several such books come to market (some of which have been reviewed in this column).

I recently received another book on this topic which is certain to sell quite widely, both because it is handled by a well known publisher (Simon and Schuster) and because its principal author is the famous science fiction writer Frank Herbert. The book was written with the help of Max Barnard, the person who worked with Herbert in setting up his computer system.

The book's title, "Without Me, You're Nothing, is taken from the author's advice that when you first set up your computer you should stand in front of it and say:

> *"You stupid, inanimate chunk of hardware! Without me, you're nothing!"*

As you can see, this book is a bit theatrical. This sense of theatrics, more than anything else, becomes the basis of one of this book's greatest shortfalls. I share some of Frank Herbert's goals, *e.g.*, the demystification of computer technology for the general public; but my fear is that he has replaced one myth with another one.

Betty and I found that many adults feel that you have to be a technical wizard to use computers effectively. We feel that this is a most damaging myth since it serves to disenfranchise a large number of people who might otherwise find utility in this technology. Our position (as regular readers of this column might remember) is that computers are like automobiles in the following way. You do not *have* to know how to drive a car to survive in our society, but you do need to know enough about them to not walk out in the street in front of one. I think that "computer literacy" is important for much the same reason. Computers are becoming so commonplace that each of us should have enough awareness of their capabilities to decide for ourselves whether or not to gain access to this technology.

Frank Herbert has a different goal in mind. He places the potential computer user in an "us" vs. "them" context. For example:

> *Things are happening in our world that make a necessity of the skills we are about to share with you. Before long it will at least be a matter of self-defense for you to have your own computer and be able to use it. You are already being taken advantage of by people with computers. You will not be able to meet that challenge or keep up with other changes unless you acquire a computer yourself.*
>
> *... Please take our warning to heart. Very soon, if you don't have access to a computer, you're going to be racing in something equivalent to the Indianapolis 500—only you'll be on foot.*

...demystification of computer technology for the general public...

Hmmm. My fear is that Mr. Herbert's zeal will result in the replacement of one type of misconception with another one.

Fortunately there are delightful streams of insight in this book which tend to counter the mild spasms of hysteria sampled above. One of the most important points that Herbert makes is that the computer is a *tool*, not a "thinking machine". The computer can amplify creative imagination, but not be creative itself. As he says:

> *A pen is a tool. A typewriter is a more sophisticated pen. A library is a tool. A painter's easel is a tool. It is the creative mind behind the tool that is important.*

Later on he says:

> *Computers may be superb for logic and accuracy within described and describable limits, but don't ever depend on one for creative work. The machine will not go outside its limits. It has no imagination. In fact, people of limited imagination, people who don't understand what you mean by "creative brainstorming", tend to lead the argument for the "electronic brain" myth. They impose limits on themselves and they want to apply similar limits to the universe because that makes them feel safer.*

So much for philosophy. The book also promises to be "a practical, easy to understand guide to *using* your own personal computer system". The technical side of this book needs tremendous reworking. I am astounded that a publisher as large as Simon and Schuster would publish a book with so many basic errors in it. For example, I have *never* heard of a disk drive being refered to as a "disk driver", but that is what Herbert calls it throughout the entire book. In his quest to show

CRYSTALWARE

12215 MURPHY AVENUE,
SAN MARTIN, CALIFORNIA 95046

THE FINEST IN FANTASY GAME SOFTWARE

APPLE	ATARI	PET	TRS-80
48K — w/disk	48K — w/disk	32K — w/disk	48 — w/disk

SPACE GAMES — Climb aboard the Starship Herman

GALACTIC QUEST — Warp through the 64 solar systems in Hires graphics with Crystalsonics sound. Fire lasers and rifter pods as you battle the Vegans in real time simulation. In Galatic Quest we have attempted to combine the best elements of a Star Trek type game with one similar to Space Trader. What is Skishi and where is the Pleasure Planet? In some systems, lurking behind swarms of asteroids, are Space Pirates who will gut your ship and leave you a derelict in space. To win Galactic Quest you must be shrewd a business man as well as a top notch warrior. $29.95

LASAR WARS — Attacking aliens in 3D. Thousands of approach simulations and 10 levels of difficulty. Three kinds of invader craft scored by size and attack mode. An exciting Invaders type game with perspective and sound. For fantasy enthusiasts from 8 to 80. $29.95

SANDS OF MARS — You embark on the Maiden Voyage of the Starship Herman. This game takes up almost 200K and uses 2 disks. It includes more than 300 Hires screens animated scenarios, a 3 dimensional maze, and a Martian labyrinth drawn entirely with hexagons. The takeoff and landing sequences are paddle or joystick controlled for the Apple and Atari and the game has 5 full scenarios. Once you land on Mars, you wander through fields of Sasquati and Degwat. Beware of the Vishu and the Lizardmen of Meshim. Seek the wise Mudra and unlock secrets hidden for centuries. Glyphs written on the Martian Sand hold the key to the mystery for which we offer $100 to the first to solve it. To answer a question we have been asked many times — yes, Sands of Mars attempts to fully utilize the sound and graphics capabilities of the Apple and Atari to their max! $39.95

ADVENTURES — Cross the threshold to new worlds.

HOUSE OF USHER — Walk the dreaded corridors of the deadly House of Usher. Complete with 40 rooms and hundreds of aggressive monsters and unique treasures. This is not your typical text adventure game but goes far beyond that, with animated monsters and a visual display of each room. We believe this to be the first indoor-outdoor game ever written for a microcomputer game which includes graphics. You may choose to wander through Usher's scenic garden paths or brave the perils of the graveyard and descend into the crypt. Beneath the house there are labyrinths, shrinking rooms, and torture chambers with no doors or windows. Your character may pick up, drop, or use objects, fire arrows, or run frantically for the door when pursued by some loathsome creature. As the old grandfather clock ticks away, you will have until dawn to solve the Usher Mystery and win a real live $100 prize! $24.95

IMPERIAL WALKER (ATARI 16K) — For you Atari people out there, here's a fantastic game pack of 4 games with revolutionary new graphics routine and Crystalsonics. As Luke Skywalker you have taken control of an All Terrain Armored Transport and with your lasers are fending off the Imperial attack forces. Written especially for Atari by Mike Potter. The game pack also includes Laser Nim, Auto Race, and Gunfight. Will run on Atari 400 or 800 with disc and Joysticks. $29.95

ADVENTURELAND 2041 (Coming June 1) — Step with us through the Crystal door into the world of tomorrow's fantasy. In the great Outback of Australia in the year of 2041, will be built the greatest Adventureland of all time. Through your computer, you can experience six separate worlds of fantasy. These programs will take up more than 500,000 bytes of memory and fill 6 disks. Many of the options may be played by up to six players and in many cases against the computer itself. $59.95

WAR GAMES

WORLD WAR III — This is a three scenario war game in Hires graphics with sound. It is not merely the conversion of a board game to computer, nor are your pieces represented by lifeless text charcters. It may be played by two persons and takes about 8 hours to complete. The rules are simple enough that you won't have to spend several days reading your manual before you can play. It contains 2 world maps and a fairly detailed map of the Iran-Iraq battle field. All scoring, animation, and positions are handled by the computer — no separate tablets to fool with. Moves are input by both players in series of 3 and when the space bar is pressed the battle becomes animated. A must see to believe $29.95

WATERLOO (Coming July 1) — A war game with graphics very similar to World War III. We have attempted to make this as detailed as possible, down to what each individual is wearing, his line of sight, and the number of bullets he has fired. It will occupy two disks and may be saved over a period of weeks. We will be publishing more information on this in BYTE MAGAZINE in July. $49.95

Order line open 24 hrs / day (408) 683-0696

VISA We accept VISA & Mastercharge master charge how "simple" computers are, he says that a light switch is the simplest computer. This kind of misconception serves no one well. For one thing, computers *are* complex (just as automobiles are complex). The beauty of computers is that you don't have to understand how they work to use them. So why, for this audience, should an author fill the book with inaccurate simplifications which might make the reader feel like a fool when sharing this new found knowledge with more technical comrades?

The authors are strong proponents of top-down programming, and have developed a new flow chart system (called PROGRAMAP) for laying out programs. I found this concept to be poorly presented, but, like much else in this book, created with good intentions. As for languages, BASIC is king for Herbert. It isn't clear how well he grasps the language himself, though, as you can see from his definition for the BASIC keyword RETURN:

> *RETURN transfers the program back to the statement after GOSUB. It is the last statement of a subroutine. (Not to be confused with directions referring to the RETURN key on your keyboard. The RETURN we refer to here is a word in BASIC that performs in the computer in a way similar to that key. With this word, you build the key's function into the program.)*

COME *ON* FRANK! The RETURN key is built into a program by PRINTing CHR$(13). It is a line terminator, period. The keyword RETURN is completely unrelated to this function.

The author of a book with the circulation this one will have should be getting much better technical advice, and his agent and publisher must share the blame for mistakes of this sort. Now, if only Erma Bombeck would write the sequel. . .

A reader writes . . .

I received a letter a few weeks ago from **COMPUTE!** reader Bob Forman who is concerned that I might be paying too much attention to the futurists. Commenting on the January '81 Computers and Society column on communications, he says:

> *I'm a believer in the computer and its place in the family, in business and in many more places that it keeps falling into. BUT IT WILL NEVER REPLACE THE NEWSPAPER and the 10 o'clock news!*

As someone who works closely with the newspaper industry, Bob shared his experiences with the use of microfilm as an alternative to bound volumes of newspapers. He found that – whatever its efficiencies might be – the poor human factors aspect of microfilm prevented it from replacing bound files (as many thought it would). He says that the reasons for this are simple:

> *Why? Bound files are simple, easier to use. Try getting someone 70 years old to sit in front of a microfilm reader or a computer long enugh to read a whole newspaper. You can't sit back in your old lounge chair and read a film reader without some pretty expensive stands or cranes to manipulate the thing, so it's not a practical thing for every evening. The young bucks can stand to read a screen for a while but it's a more tiring process than reading a paper ... And, I haven't seen a high speed printer yet which will show a picture of a cabbage head accurately, or anything that approaches a good photograph.*

I think that reader Bob makes a good point – but only if one talks about one media format replacing another one. The telephone has not yet eliminated the mail and telegraph. The television has not yet eliminated the radio. I do not believe that any rational person thinks that the printed word will disappear when terminals appear in everyone's home. What I *do* believe is that a very large segment of the general population will start fitting the computer information utility into their mix of information sources, and that it will result in the kind of re-equilibration period we had when television started to compete with radio.

The most important advantage of computer based information utilities is their ability to access many diverse data bases, rather than forcing the user to listen to one person's view of the news.

As always, it is great to hear from readers. I look forward to your letters and messages (I can still be reached on the Source at TCE132). Till next month. . . . ©

You may write to David at Innovision,
P.O. Box 1317,
Los Altos, CA 94022.

NEC spinwriter™

quality, reliability, and affordability is NOW available backed by service and support you can depend on.

Dealers, O.E.M.'s, and end-users all realize that service is the key ingredient of a wise purchase—SO DOES T-COM!

T-COM, a fully authorized NEC distributor, believes a successful sale doesn't end with the delivery of the product. We offer the service and support to keep your Spinwriter operating with the least amount of down time for you and your customers. With our fully factory trained service staff and unique service policies, we make every effort to keep your purchase from T-COM an investment you can count on to deliver satisfaction everyday.

T-COM represents a select group of dependable state-of-the-art word and data processing products and supplies. From Data-MATE and Smith Systems terminal furniture to NEC printers and supplies, you can count on the finest products available all backed by T-COM's firm commitment to service and support.

Dealer and end user inquiries are welcome for these fine products:

■ Data-MATE Terminal Furniture ■ Printer Interface Cables ■ Gates Sound Covers
■ NEC Spinwriter ■ Spinwriter supplies and accessories ■ Smith System Furniture

FOR INFORMATION ON DEALERSHIPS, PRODUCTS, AND SERVICE

Contact T-COM, INC.
Two North Plaza P.O. Box 3005
Jackson, Michigan 49204
Phone (517) 787-9280

T-COM, INC.

a C, C&S Systems Company

EXCLUSIVE from COMPUMART! Special Offer. Zenith Color Video Monitor for $379!

NEW FROM SANYO—Four Great Monitors at Low COMPUMART Prices. Sanyo's new line of CRT data display monitors are designed for the display of alphanumeric or graphic data.

9" Sanyo Monitor B/W	**$179.**
12" Sanyo Monitor B/W	**$289.**
12" Sanyo Monitor with green screen	**$299.**

THE NEC MONITOR—Composite video using BNC connectors. 8-pin connector for VCR/VTR video loop, In/Out and television reception. **$599.**

CENTRONICS PRINTERS
The incredible Model 737. The closest thing to letter quality print for under $1,000. 737-1 (Parallel Interface)
List Price $899. **SPECIAL $795.**
730-1 **Sale Price $695.**

NEW FROM INTEGRAL DATA
THE IDS 560 PRINTER. All the exciting features of the 400 series plus 14½" paper capacity, 132 col. graphics printer.
Price $1695.

IDS 445. Priced lower than the 440 and equipped with a better print head. IDS 445 w/graphics **$894.** IDS 445 w/o graphics **$795.**

IDS 460 $1,295
The 460's features include: Correspondence quality printing, high resolution graphics capability, programmable print justification.

Omni Printers from Texas Instruments
The 810—List $1895. **SALE! $1795.**
The 820 (Ro) Package—Includes machine-mounted paper tray and cable. A compressed print option and device forms control are standard features. **$2155.**
The 820 (KSR) Package—Includes full ASCII Keyboard plus all of the features of the RO **$2395.**

COMPUMART FEATURES TYPEWRITER QUALITY PRINTERS:

C-ITOH STARWRITER: LETTER QUALITY PRINTING FOR UNDER $2,000!
This daisywheel printer gives high quality at a low price. 25 cps. Parallel and serial interfaces available. **Price $1895.**
For this month only COMPUMART will give **10% off** on any word processing package when you buy one of the above letter quality printers from us.

NEC THE FIRST NAME IN LETTER QUALITY PRINTERS
COMPUMART offers beautiful print-quality with NEC Spinwriter terminals. Both KSR and RO versions give unsurpassed hard copy output. CALL

NEC Spinwriters include these special features:
—One of the highest reliability factors in the industry. Less down time.
—Spinwriters utilize the unique NEC design thimble as opposed to the daisywheel. Capable of 30 million impressions.
—National Service.
—Both thimbles and ribbons are quickly and easily changed. Ribbons come in a cartridge for stain free removal.

Options Available for NEC Spinwriters:
—Field installable tractor feed
—Can be ordered with cut sheet feeder installed at factory.

	COMPUMART PRICE	List Price
5510	**$2749.**	$3055.
5515	**$2849.**	$3135.
5520	**$3149.**	$3415.
5525	**$3249.**	$3455.
5530	**$2749.**	$3055.

SPECS

5510 Receive—Only (RO), Friction feed platen, copy control, impression control, bidirectional, 10-12 pitch, 6-8 lines per inch, autofeed, ribbon-out, cover open switch, and self-test.
5515 RO unit, specs. same as 5510s. Diablo compatible.
5520 KSR with numeric keypad. Specs same as 5510.
5525 KSR with numeric keypad. Diablo compatible, specs same as 5510.
5530 RO unit, same specs as 5510 Centronics parallel interface.

NEC AND C-ITOH!

COMPUMART has been serving the computer needs of industry since 1971.

We stock, for immediate shipment, only those products from the finest micro-computer manufacturers.

And any product, except software, can be returned within 10 days for a full refund—even if you just change your mind.

We also honor all manufacturers' warranties.

VISA MasterCard

Our expert technicians will service any product we sell that cannot be better, or faster, serviced by the manufacturer's local service center.

Call us for more information on products, product configuration and service. Our phones are open Monday thru Friday, 8:30 a.m. to 7:00 p.m. and Saturday 11:00 a.m. to 4:00 p.m.

We have a staff of highly knowledgeable sales people waiting to hear from you, and to help.

Because service is what we're all about.

800-343-5504

In Mass. Call 617-491-2700

SYSTEMS Customized

COMPUMART

65 BENT STREET, DEPT. 1406, P.O. BOX 568
CAMBRIDGE, MA 02139

Editor's Note: From time to time we present what we choose to call "Guest Commentaries". These articles don't necessarily express the opinion of **COMPUTE!**, *but generally do raise questions we think should be discussed. . . RCL*

Computer Aided Instruction, Boon or Bust?

Alfred D'Attore
Phoenix, AZ 85021

Computer Aided Instruction, (CAI,) has been around for quite a while. Originally introduced into out public schools when "Time Share" became commonplace — about ten years ago — it has met with rather indifferent success. At its best, it appeared to offer no particular advantage over traditional teaching methods. At its worst — and that could be very bad indeed, with the frequent equipment "crashes" and student blunders — it was frustrating and ineffective. It was *always* crushingly expensive. School boards had horrible visions of endless banks of computer terminals with attending telephone connections, computer time costs, repair contracts — an endless cash flow.

The personal computer boom of recent years has eased expenses somewhat, but CAI is still not employed to any great extent in our public schools. Even when computer systems are purchased, they are rarely used for CAI. Rather, they are used to support a relatively minimal study of computer programming and the endless, ever-present games. Sometimes, they are not used at all. I know of one school in North Phoenix which recently purchased a disk-operated computer system complete with printer. Although access is provided, it lies virtually unused in an office, gathering dust.

The reason, of course, is the lack of suitable, appropriate software. Too few people are programming for our public schools. And when, occasionally, we do obtain CAI programs, they are most often tutorial in nature and therefore inappropriate for use in primary and secondary schools. Let me elaborate upon this point.

Any public school teacher can tell you that the normal learning process involves a very small amount of "teaching" and an immense amount of "doing." This is especially true when the subject areas are basic; for example: reading and arithmetic. In this circumstance, even the most skillful CAI, if it is basically tutorial, is a waste of time and good programming talent. It is simply too much work for too little return.

And this assumes the programming is successful. Often it is not. Often, the programming places too much burden upon the student with respect to display interpretation and console operation. Many programs have "bugs." Since with this type of programming, the student interfaces directly with the computer, the frustration level often runs very high.

But the most important reason for the general ineffectiveness of this type of programming in our public schools, lies in the very nature of our young students. The classroom teacher quickly learns that young people must establish an acceptable *personal* relationship with their instructor before meaningful learning can take place. An indifferent machine is at a big disadvantage there.

...let's allow the teacher to teach...

Certainly, if tutorial programs are prepared cleverly, students will be enthralled, initially. But that never lasts very long. In my classes, three weeks is about par, after which the system becomes just another classroom static fixture, like the countless desk calculators, visual aids and programmed instruction packages that remain largely unused in every classroom. Yet, if software is available at all for the first twelve grades, it is most often of this type.

Of even less use are the ancillary programs: the "curriculum guides to CAI," the "systems approaches-cum-administrative programming" packages and the various conceptual outlines. Teachers get "overviews" by the bucketfull. We treat them with the respect due most things that come in buckets. We need specifics, not generalities. I will be specific.

Let's allow the teacher to teach. Then we may use the computer to help him with his job.

The computer should be programmed to do that which it is uniquely qualified to do: create exercises. As I pointed out previously, individualized student work — exercises — represents the greater portion of the learning process. A computer, working in this fashion, will be helping the teacher do the greater part of his job. In skill-oriented subjects like arithmetic, for example, students are required to do exercises repeatedly, with graduated levels of difficulty. Students are drilled.

There, I've said it. That dirty word: drill. It has become anathema in recent years. It is supposed to turn students off. But realistically, there isn't anyother way to learn basic skills, especially basic

COLLEGE BOARDS

for TRS-80 PET, APPLE

The best way to sharpen your skills for the College Boards is to work on actual examinations. Each of these program sets confronts the user with a virtually limitless series of questions and answers Each program is based on past exams and presents material of the same level of difficulty and in the same form as used in the College Board examination. Scoring is provided in accordance with the formula used by College Boards.

SAT, PSAT, N.M.S.Q.T., set includes 7 programs covering Vocabulary, Word Relationships, and Mathematics. Price $79.95

EDUCATOR EDITION - SAT, PSAT Includes all of the above programs plus detailed solutions and explainations for each problem plus drill exercises. SAT set includes 14 programs. $149.95

GRADUATE RECORD EXAMINATION set includes 10 programs covering Vocabulary, Word Relationships, Mathematics, Logical Diagrams, Analytical Reasoning. $139.95

EDUCATOR EDITION - Graduate Record Exam Set includes 20 programs. $199.95

SWORD OF ZEDEK

Fight to overthrow Ra, The Master of Evil. In this incredible adventure game, you must confront a host of creatures, natural and supernatural. To liberate the Kingdom, alliances must be forged and treasures sought. Treachery, deceit and witchcraft must be faced in your struggles as you encounter wolves, dwarves, elves, dragons, bears, owls, orcs, giant bats, trolls, etc. Each of the 12 treasures will enhance your power, by making invisible, invulnable, more eloquent, more skillful in combat etc., etc., as you explore the realms of geography both on the surface and underground. Dungeons, temples, castles, mountains etc., are all a part of the fantastic world of Ra. Each game is unique in this spectacular and complex world of fantasy. $24.95

KRELL GAME PAK

Incredible bargain, 15 unique and challenging games!!!

Hostage — **Primary Fight**
Prime Time — **Black Gold**
The Black Death — **Wordsworth**
Star Clipper — **Hard Scrabble**
Bulls & Bears — **Bible Quode**
Banzai — **Shakespeare Quode**
Banzai II — **Bill of Rights Quode**
Super Banzai *TRS-80 only, $39.95*

SUPER STAR BASEBALL

ALL TIME SUPER STAR BASEBALL Sample Lineup		SUPER STAR BASEBALL Sample Lineup	
B. Ruth	T. Williams	D. Parker	J. Rice
L. Gehrig	J. Foxx	W. Stargell	H. Aaron
J. DiMaggio	H. Greenberg	W. Mays	L. Brock
J. Jackson	R. Hornsby	P. Rose	R. Carew
G. Sisler	H. Wilson	O. Cepeda	H. Killebrew
S. Musial	B. Terry	C. Yazstremski	R. Allen
T. Cobb	M. Mantle	W. McCovey	R. Leflore
W. Mays	H. Aaron	R. Jackson	R. Zisk
C. Young-P	W. Johnson-p	G. Brett	B. Madlock
		R. Guidry-P	T. Seaver-p

Performance is based on the interaction of actual batting and pitching data. Game can be played by one or two players with the computer acting as a second player when desired. Players select rosters and lineups and exercise strategic choices including hit and run, base stealing, pinch hitting, intentional walk, etc. Highly realistic, there are two versions, ALL TIME SUPER STAR BASEBALL, and SUPER STAR BASE—BALL featuring players of the current decade. Each includes about 50 players allowing nearly an infinite number of roster and lineup possibilities.

$14.95

TIME TRAVELER

The best of the adventure games. Confronts player with complex decision situations and the demand for real time action. Using the Time Machine, players must face a challenging series of enviornments that include; the Athens of Pericles, Imperial Rome, Nebuchadnezzar's Babylon, Ikhnaton's Egypt, Jerusalem at the time of the crucifixion, The Crusades, Machiavelli's Italy, the French Revolution, the American Revolution, and the English Civil War. Deal with Hitler's Third Reich, Vikings, etc. At the start of each game players may choose a level of difficulty... the more difficult, the greater the time pressure. To succeed you must build alliances and struggle with the ruling powers. Each game is unique. $24.95

*ALL PROGRAMS AVAILABLE FOR TRS-80, APPLE II & PET

•Programs for APPLE or TRS-80 are on disk or cassett, please specify.

All programs require 16K • TRS-80 programs require LEVEL II BASIC • APPLE programs require Applesoft BASIC

KRELL SOFTWARE

Send check or money order to
21 Milbrook Drive, Stony Brook, NY 11790
(516) 751-5139

arithmetic skills. One must perform in a skill to make it one's own. And I don't mean just once. Indeed, I sometimes think, to the extent the learning process is difficult, to that same extent is the learning worthwhile.

But far from turning students off, my experience has taught me that young people become eager — even enthralled — when they begin to acquire measurable skills. And drill does it. Disciplined, repeated, old-fashioned drill. For drill, the computer is without parallel.

In my approach, a printer is required. Exercises must be printed out at all times, if the computer is to be used effectively. Exercises must be produced immediately, in unlimited numbers, tailored specifically to meet the particular need, and optimized for clarity, organization and student use.

Answers must be provided for all exercises. Where appropriate, they should be reduced to lowest terms. There should be no ambiguities. When dividing with decimals, for example, accuracy requirements should be ordered and neat. Since students work directly upon these exercise sheets, this will coerce them, gently, to be equally neat. This is most important, especially for students in remediation. Very often, their work is much too sloppy, and like other students, they tend to relate their teacher's requirement for neatness to "nit-picking," rather than to recognition of the fact that ours is a place-value number system. A digit's position in a numeral is quite as important as its value. Sloppiness confuses "place."

Spaces should be provided between digits in all those exercises where "carries" and like manipulations are required. Students should not be forced to crowd their work. Alternately, they cannot be permitted so much room as to encourage carelessness. "Neatness begets neatness. Order begets order." I don't know who said that first. Perhaps it's a paraphrase. But it is a dictum that should be kept foremost in mind when preparing computer aided instruction of this type.

To illustrate, a portion of an exercise sheet for integer addition is shown in figure 1. In this particular program, an ordered pair of numbers specifies the number of addends and the number of digits per addend. Note the "spacing" of digits. The number of problems and their spacing are set under program control. They vary automatically with the difficulty level of the problems.

```
LESSON NO. 1  Name ______________  Class ________
(01)   5 4 6 1    (02)   6 7 4 7    (03)   7 5 8 2
       5 4 6 5           2 2 7 2           7 7 6 7
       9 5 0 6           9 8 6 0           3 5 7 1
      ________          ________          ________
```

Figure 1

Of course, for basic skills instruction, programs running the gamut of arithmetic skills are required. I have used just such programming for the past five years. Permit me now to enumerate the advantages that have come to light in this period:

Programs are immediately adaptable to student competency levels. Through simple question and answer, an instructor may choose from a number of levels of difficulty.

Parents and family may enter into the training process. Since exercises are produced in moments and answers are provided in the appropriate formats, students may take any number of them home and be drilled by other family members.

...the computer becomes a valuable teacher's aid...It is not a surrogate teacher...

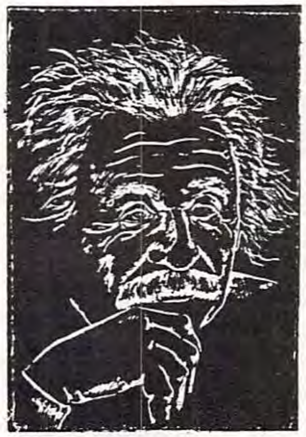
Individualized instruction — always desirable in the classroom situation — becomes less open-ended. The student runs little risk of drilling himself in incorrect procedures. With individualized instruction — for reasons of practicability — a student is often required to work for extended periods without direct supervision. With the answers before him, however, he cannot fail to be alerted to incorrect procedures.

The computer becomes a valuable teacher's aid. It is swift, versatile, flexible, indefatigable and inexhaustible. But it is an aid: no more. It is not a surrogate teacher. This approach is, therefore, non-threatening. Since computer aided instruction and its associated equipment must be sold — essentially — to teachers, this is a not-inconsiderable advantage.

Last, this approach is cost-effective. A computer system, used in this manner, is easily affordable. A 2,000 dollar system can serve a school. Such a system currently serves the school wherein I'm employed. Admittedly, this is a bare-bones approach, and I don't suggest for a moment that other schools should spend so little. In today's market, 5,000 dollars would purchase a disk-operated system with sufficient equipment backup to insure reliable operation for an indefinite period. *That* is the proper way to go.

In this article, I have dealt primarily with the mathematics in describing this "alternate approach" to CAI. But I have gone far enough afield in my programming efforts to have determined these methods are applicable in other teaching disciplines. With right programming, computers can be a boon indeed for our public schools. Without it, they are just expensive toys. So what shall it be? Boon or bust?

An Intelligent Alternative



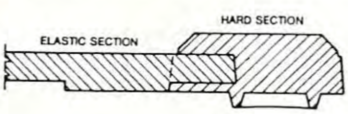
TYPRINTER 221

In the research you are doing before purchasing your computer printer, you are probably confused by the various claims, speeds, choices, shapes and prices. Well, we'd like to clear the air a bit and tell you about the most unusual computer-printer around — the TYPRINTER 221.

You see, it's unusual because it is **totally compatible** with **every** computer and word processing program . . . from the largest to the smallest. It's versatile to the point of incredibility . . . We'll discuss the broad advantages and explain the details.

THE DAISY WHEEL

The special daisy wheel supplied is of a unique design consisting of a 100 character carrying radii. Each radii is formed of two distinct types of plastic — an "elastic plastic" for the stalk of the radii, and a comparatively "hard plastic" used to form the character area. This, combined with a very narrow character profile and a special positioner on each of the 100 radii, guarantees a uniform character density. There is near perfect geometric positioning of the character with no character higher or lower than the others. And because of its unique dual material design, microvibrations have virtually been eliminated, leaving your final copy clean, clear and smudge free. The copy produced is comparable to that produced by metal daisy wheels and at a fraction of the cost.



THE KEYBOARD

The keyboard has been referred to as a triumph of human engineering - from the way the keys seem to have been custom designed to fit your fingers, to the way the special feature switches have been grouped. A flip of a switch (or under computer control of course) and the printer becomes a foreign language machine. Push a button, and like magic the printer automatically locates and lines up columns of figures, perfectly balanced between the margins. This incredibly fast, extraordinarily quiet electronic keyboard puts more programming power at you fingertips then printers costing five to ten times as much.

THE DISPLAY

The TYPRINTER 221 presents a new dimension in operator/machine communications. In the manual (typewriter) mode, the printer controls and verifies all entries before printing. The display exhibits the last 15 characters of the text, word-by-word, until the end of the line. The operator may control what will be printed before the actual printing takes place. This new found flexibility enables you to make modifications along the entire line and in both directions. This 20 character plasma display has the ability to scroll backwards as well as forwards; will give the operator a visual indication as to which print mode is currently being selected as well as the number of characters remaining before the right margin is reached. The display will also indicate to the operator:

The number of characters available in the memory	What characters will be inserted into an existing text.
When the printer is in an error condition	When the memory for the previous line has been selected.
When a pre-programmed form layout has been selected	A warning message that the end of the page is being approached.
When the printer is operating from the internal memory.	That a hyphenation decision must be made

PRINT MODE

The TYPRINTER 221 will allow you to automatically highlight individual characters, words or complete sentences. Whatever is entered from the keyboard or from the computer, even an existing text file, can be printed in one or more of the five different modes:

- traditional printing;
- underlined characters;
- true bold characters where the horizontal component of the character is increased without disturbing the vertical component;
- characters which are both bold and underlined, and;
- a feature unique among computer printers-printing in reverse — white on black, sort of reverse video on paper.

MULTILINGUAL CAPABILITY

A unique and useful feature of the TYPRINTER 221 is its capability of being able to print in several languages without changing the daisy wheel. In addition to English, every standard daisy wheel has the ability and the necessary characters to print in French, Spanish, Italian and German.

THE FEATURES

Automatic justification of the right margin
The electronics of the TYPRINTER 221 have made right hand justification a simple, automatic operation.

Phrase and format storage
Phrases, dates, addresses, data, etc. that may be stored in your computer's memory may be sent over to the printer and stored in one of the "memory bins" of the printer. This information may then be used by the operator in the manual mode. This can save you hours when trying to get a form "just right."

Automatic centering
The TYPRINTER 221 will not only center any title between the pre-set margins, but will also center over one or more columns, or over any specific point and will even align copy with the right margin independent of the left margin.

Automatic vertical lines
A command from the computer enables an automatic feature which prints vertical lines at any point on the paper.

Automatic tab sequence recall
With the TYPRINTER 221 you may store and recall the most frequently needed margin and tab sequences for applications such as daily correspondence, statistical reports, etc. This guarantees consistent high quality appearance of each document.

Paragraph indent
A computer command instantly sets a temporary margin in order to print one or more indented paragraphs with respect to the right margin.

Automatic decimal point location
No matter how many figures to either the left or right of the decimal point, the TYPRINTER 221 will automatically line up the figures with the decimal point in any position you choose. Statistical printing has never been easier.

Column layout
This feature allows you to obtain automatic and perfect distribution of spaces between columns in respect to the margins. A perfect page balance is assured without the need to carry out calculations or additional operations.

There is a wide variety of options that you can add to TYPRINTER 221.

By now you are probably convinced that **we are sold** on our machine, and we hope you can understand why. In fact, why don't you use these facts to measure against any and/or all the other computer printers on the market.

When you do, you will realize the TYPRINTER 221 is an intelligent electronic typewriter, a text formatter — and a brilliant computer printer — available at a suggested list price of only $2850.

TYPRINTER 221 is available at your local computer shop — or we'll tell you where you can see and try one if you call us at

HOWARD INDUSTRIES
2051 E. CERRITOS AVE., 8C
ANAHEIM, CA 92806
714/778-3443

The Mysterious And Unpredictable RND

Bob Albrecht and
George Firedrake

From a book of the same name by Dymax Publishing Company; copyright ©1980 Dymax. Permission to reprint by teachers for classroom use is granted.

Editor's Note: You may reach Bob & George by mail at: P.O. Box 310 Menlo Park, CA 94025

> *Editor's Note:*
> *We conclude our presentation of The Mysterious And Unpredictable RND with this installment. We expect to make the series available to teachers, in booklet form, within the next few months.*
> *...RCL*

Solutions and Stuff

Here are our solutions. Yours may be different. That's OK, as long as they solve the problem. One really nice thing about computers: There are many ways to write a program that works!

Exercise 1.

(a) The smallest RND number is .0103099732 in the first sample.

(b) The largest RND number is .984101932 in second sample.

Exercise 2. Smallest RND Number In A Sample

```
100 REM****SMALLEST RND NUMBER IN A SAMPLE

200 REM****FIND OUT HOW BIG A SAMPLE
210 PRINT "[CLR]" ;
220 PRINT
230 INPUT "HOW MANY RND NUMBERS" ; N

300 REM****SET SMALL EQUAL TO FIRST RND NUMBER
310 SMALL = RND(1)

400 REM****DO REST OF SAMPLE. COMPARE EACH RND
410 REM****NUMBER WITH SMALL. IF SMALLER, REPLACE.
420 FOR K = 1 TO N - 1
430   X = RND(1)
440   IF X < SMALL THEN SMALL = X
450 NEXT K

500 REM****PRINT SMALL AND GO BACK FOR MORE
510 PRINT "LARGEST NUMBER IN SAMPLE IS" SMALL
520 GOTO 220

999 END
```

Exercise 3. The Small And Big

In this program, we first set *both* SMALL and BIG to the *same* first RND number (lines 310 and 320).

```
100 REM****SMALLEST AND LARGEST RND NUMBER IN SAMPLE

200 REM****FIND OUT HOW BIG A SAMPLE
210 PRINT "[CLR]" ;
220 PRINT
230 INPUT "HOW MANY RND NUMBERS" ; N

300 REM****SET SMALL AND BIG EQUAL FIRST RND NUMBER
310 SMALL = RND(1)
320 BIG = SMALL

400 REM****DO REST OF SAMPLE. COMPARE EACH RND
410 REM****NUMBER WITH SMALL AND BIG.
420 FOR K = 1 TO N - 1
430   X = RND(1)
440   IF X < SMALL THEN SMALL = X
450   IF X > BIG THEN BIG = X
460 NEXT K

500 REM****PRINT SMALL AND BIG, GO BACK FOR MORE
510 PRINT "SMALLEST NUMBER IN SAMPLE IS" SMALL
520 PRINT "LARGEST NUMBER IN SAMPLE IS" BIG
530 GOTO 220

999 END
```

The following method might not work. Why not?

310 SMALL = RND(1)
320 BIG = RND(1)

Exercise 4.

(a) 7 (b) 5 (c) 0

The integer part of .328904955 even though it isn't printed.

Exercise 5.

(a) 2 (b) 0 (c) 7

Exercise 6.

(a) 220 PRINT INT(2*RND(1)),
(b) 220 PRINT INT(6(RND(1)),
(c) 220 PRINT INT(100*RND(1)),

Exercise 7.

(a) 220 PRINT INT(2*RND(1)) + 1,
(b) 220 PRINT INT(8*RND(1)) + 1,
(c) 220 PRINT INT(100*RND(1)), + 1,
(d) 220 PRINT INT(2*RND(1)) + 2,
(e) 220 PRINT INT(3*RND(1)) + 3,
(f) 5, 6, 7, or 8
(g) 2, 4, or 6

Exercise 8.

430 IF COIN = 0 THEN T = T + 1
440 IF COIN = 1 THEN H = H + 1

Exercise 9.

There are many ways to write this program. Here are two ways.

```
100 REM****COIN FLIPPER #4

200 REM****FIND OUT HOW MANY FLIPS
210 PRINT "[CLR]" ;
220 INPUT "HOW MANY COIN FLIPS" ; N

400 REM****FLIP TWO COINS N TIMES
410 FOR K = 1 TO N

420   C1 = INT(2*RND(1))
430   C2 = INT(2*RND(1))

440   IF C1 = 1 AND C2 = 1 THEN PRINT "HH" ,
450   IF C1 = 1 AND C2 = 0 THEN PRINT "HT" ,
460   IF C1 = 0 AND C2 = 1 THEN PRINT "TH" ,
470   IF C1 = 0 AND C2 = 0 THEN PRINT "TT" ,

480 NEXT K
490 PRINT

999 END
```


```
100 REM****COIN FLIPPER #4A
110 A$(0)="TT" : A$(1)="TH" ; A$(2)="HT" : A$(3)="HH"

200 REM****FIND OUT HOW MANY FLIPS
210 PRINT "[CLR]" ;
220 INPUT "HOW MANY FLIPS" ; N

400 REM****FLIP TWO COINS N TIMES
410 FOR K = 1 TO N

420   C1 = INT(2*RND(1))
430   C2 = INT(2*RND(1))

440   PRINT A$(2*C1 + C2)

450 NEXT K
460 PRINT

999 END
```

Exercise 10.

We did it by modifying our first program of Exercise 9. Make these changes and additions to COIN FLIPPER 4.

```
100 REM****COIN FLIPPER #5

300 REM****SET FLIP COUNTERS TO ZERO
310 HH = 0
320 HT = 0
330 TH = 0
340 TT = 0

440   IF C1 = 1 AND C2 = 1 THEN HH = HH + 1
450   IF C1 = 1 AND C2 = 0 THEN HT = HT + 1
460   IF C1 = 0 AND C2 = 1 THEN TH = TH + 1
470   IF C1 = 0 AND C2 = 0 THEN TT = TT + 1

500 REM****PRINT RESULTS OF N FLIPS
510 PRINT "OUTCOME", "NUMBER OF TIMES"
520 PRINT " HH    ", HH
530 PRINT " HT    ", HT
540 PRINT " TH    ", TH
550 PRINT " TT    ", TT
```

Exercise 11.

Program to roll two dice, N times.

```
100 REM****DICE ROLLER #2

200 REM****FIND OUT HOW MANY ROLLS
210 PRINT "[CLR]" ;
220 INPUT "HOW MANY DICE ROLLS" ; N

400 REM****ROLL TWO DICE N TIMES
410 FOR K = 1 TO N
420   D1 = INT(6*RND(1)) + 1
430   D2 = INT(6*RND(1)) + 1
440   SUM = D1 + D2
450   PRINT SUM,
460 NEXT K
470 PRINT

999 END
```

Exercises 12 and 13.

OUTCOME	NUMBER OF WAYS	PROPORTION
2	1	1/36 = .0278
3	2	2/36 = .0556
4	3	3/36 = .0833
5	4	4/36 = .1111
6	5	5/36 = .1389
7	6	6/36 = .1667
8	9	5/36 = .1389
9	4	4/36 = .1111
10	3	3/36 = .0833
11	2	2/36 = .0556
12	1	1/36 = .0278

Exercise 14.

In this program, we use FREQUENCY to mean NUMBER OF TIMES.

```
100 REM***DICE ROLLER #3

200 REM***FIND OUT HOW MANY FOLLS
210 PRINT " [CLR] " ;
220 INPUT "HOW MANY ROLLS" ; N

300 REM***SET OUTCOME COUNTS TO ZERO
310 DIM F(12)
320 FOR X = 2 TO 12
330    F(X) = 0          <----    F(X) will be the number of
340 NEXT X                        times outcome X occurred.

400 REM***ROLL DICE, COUNT OUTCOMES
410 FOR K = 1 TO N
420    D1 = INT(6*RND(1)) + 1  }
430    D2 = INT(6*RND(1)) + 1  } X is sum of two dice, D1 and D2
440    X = D1 + D2             }

450    F(X) = F(X) + 1 <----     Increase count for outcome X by 1

460 NEXT K

500 REM***PRINT COUNTS AND PROPORTIONS
510 PRINT
520 PRINT "OUTCOME, "FREQUENCY", "PROPORTION"
530 PRINT
540 FOR X = 2 TO 12
550    PRINT X, F(X), F(X)/N
560 NEXT X

999 END
```

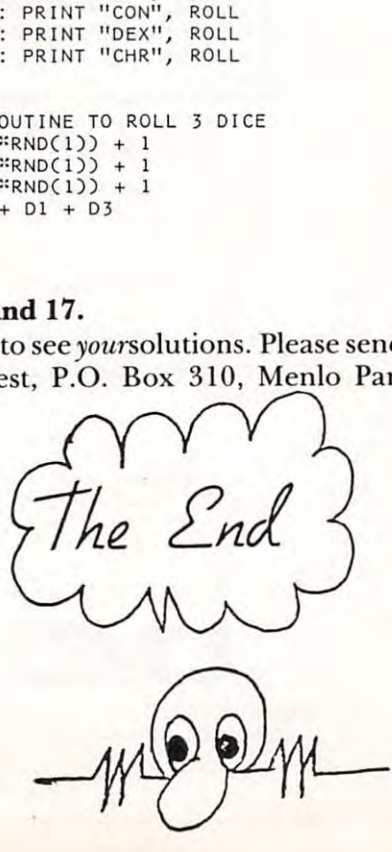
Exercise 15.

Since we had to roll three dice six times, we used a *subroutine* to roll the dice.

```
100 REM***CREATE AN ADVENTURER
110 PRINT " [CLR] " ;

200 REM***ROLL = SUM OF THREE DICE
210 GOSUB 310 : PRINT "STR", ROLL
220 GOSUB 310 : PRINT "IQ", ROLL
230 GOSUB 310 : PRINT "LK", ROLL
240 GOSUB 310 : PRINT "CON", ROLL
250 GOSUB 310 : PRINT "DEX", ROLL
260 GOSUB 310 : PRINT "CHR", ROLL
270 STOP

300 REM***SUBROUTINE TO ROLL 3 DICE
310 D1 = INT(6*RND(1)) + 1
320 D2 = INT(6*RND(1)) + 1
330 D3 = INT(6*RND(1)) + 1
340 ROLL = D1 + D1 + D3
350 RETURN

999 END
```

Exercises 16 and 17.

We would like to see *your* solutions. Please send them to DragonQuest, P.O. Box 310, Menlo Park, CA 94025.



DYNACOMP

Quality software for⁺:

ATARI	**TRS-80 (Level II)***
PET	**NORTH STAR**
APPLE II Plus	**CP/M 8" Disk**

GAMES, SIMULATIONS, EDUCATION and MISCELLANEOUS

BRIDGE 2.0 (Available for all computers) — **Price: $17.95 Cassette/$21.95 Diskette**
An all-inclusive version of this most popular of card games. This program both BIDS and PLAYS either contract or duplicate bridge. Depending on the contract, your computer opponents will either play the offense OR defense. If you bid too high, the computer will double your contract! BRIDGE 2.0 provides challenging entertainment for advanced players and is an excellent learning tool for the bridge novice.

HEARTS 1.5 (Available for all computers) — **Price: $14.95 Cassette/$18.95 Diskette**
An exciting and entertaining computer version of this popular card game. Hearts is a trick-oriented game in which the purpose is not to take any hearts or the queen of spades. Play against two computer opponents who are armed with hard-to-beat playing strategies.

CRIBBAGE (TRS-80 only) — **Price: $14.95 Cassette/$18.95 Diskette**
This is simply the best cribbage game available. It is an excellent program for the cribbage player in search of a worthy opponent as well as for the novice wishing to improve his game. The graphics are superb and assembly language routines provide rapid execution.

STUD POKER (ATARI only) — **Price: $11.95 Cassette/$15.95 Diskette**
This is the classic gambler's card game. The computer deals the cards one at a time and you (and the computer) bet on what you see. The computer does not cheat and *usually* bets the odds. However, it sometimes bluffs! Also included is a five card draw poker betting practice program. This package will run on a 16K ATARI. Color, graphics, sound.

POKER PARTY (Available for all computers) — **Price: $17.95 Cassette/$21.95 Diskette**
POKER PARTY is a draw poker simulation based on the book, POKER, by Oswald Jacoby. This is the most comprehensive version available for microcomputers. The party consists of yourself and six other (computer) players. Each of these players (you will get to know them) has a different personality in the form of a varying propensity to bluff or fold under pressure. Practice with POKER PARTY before going to that expensive game tonight! Apple Cassette and diskette versions require a 32 K (or larger) Apple II.

NOMINOES JIGSAW (Atari, Apple and TRS-80 only) — **Price: $16.95 Cassette/$20.95 Diskette**
A jigsaw puzzle on your computer! Complete the puzzle by selecting your pieces from a table consisting of 60 different shapes. NOMINOES JIGSAW is a virtuoso programming effort. The graphics are superlative and the puzzle will challenge you with its three levels of difficulty. Scoring is based upon the number of guesses taken and by the difficulty of the board set-up.

CRANSTON MANOR ADVENTURE (North Star only) — **Price: $19.95**
At last! A comprehensive Adventure game for the North Star. CRANSTON MANOR ADVENTURE takes you in mysterious CRANSTON MANOR where you attempt to gather fabulous treasures. Lurking in the manor are wild animals and robots who will not give up the treasures without a fight. The number of rooms is greater and the associated descriptions are much more elaborate than the current popular series of Adventure programs, making this game the top in its class. Play can be stopped at any time and the status stored on diskette. Requires 32K.

VALDEZ (Available for all computers) — **Price: $14.95 Cassette/$18.95 Diskette**
A simulation of supertanker navigation in the Prince William Sound and Valdez Narrows. The program uses an extensive 256X256 element radar map and employs physical models of ship response and tidal patterns. Chart your own course through ship and iceberg traffic. Any standard terminal may be used for display.

FLIGHT SIMULATOR (Available for all computers) — **Price: $17.95 Cassette/$21.95 Diskette**
A realistic and extensive mathematical simulation of take-off, flight and landing. The program utilizes aerodynamic equations and the characteristics of a real airfoil. You can practice instrument approaches and navigation using radials and compass headings. The more advanced flyer can also perform loops, half-rolls and similar aerobatic maneuvers.

STARTREK 3.2 (Available for all computers) — **Price: $ 9.95 Cassette/$13.95 Diskette**
This is the classic Startrek simulation, but with several new features. For example, the Klingons now shoot at the Enterprise without warning while also attacking starbases in other quadrants. The Klingons also attack with both light and heavy cruisers and move when shot at! The situation is hectic when the Enterprise is besieged by three heavy cruisers and a starbase S.O.S. is received! The Klingons get even!

CHESS MASTER (North Star and TRS-80 only) — **Price: $19.95 Cassette/$23.95 Diskette**
This complete and very powerful program provides five levels of play. It includes castling, en passant captures and the promotion of pawns. Additionally, the board may be preset before the start of play, permitting the examination of "book" plays. To maximize execution speed, the program is written in assembly language (by SOFTWARE SPECIALISTS of California). Full graphics are employed in the TRS-80 version, and two widths of alphanumeric display are provided to accommodate North Star users.

BLACK HOLE (Apple only) — **Price: $14.95 Cassette/$18.95 Diskette**
This is an exciting graphical simulation of the problems involved in closely observing a black hole with a space probe. The object is to enter and maintain, for a prescribed time, an orbit close to a small black hole. This is to be achieved without coming so near the anomaly that the tidal stress destroys the probe. Control of the craft is realistically simulated using side jets for rotation and main thrusters for acceleration. This program employs Hi-Res graphics and is educational as well as challenging.

HODGE PODGE (Apple only, 48K Integer BASIC) — **Price: $19.95 Cassette/$23.95 Diskette**
Let HODGE PODGE be your child's baby sitter. Pressing any key on your Apple will result in a different and intriguing "happening" related to the letter or number of the chosen key. The program's graphics, color and sound are a delight for children from ages 1½ to 9. HODGE PODGE is a non-intimidating teaching device which brings a new dimension to the use of computers in education. HODGE PODGE requires a 48K Apple running with Integer BASIC.

TEACHER'S PET I (Available for all computers) — **Price: $ 9.95 Cassette/$13.95 Diskette**
This is the first of DYNACOMP's educational packages. Primarily intended for pre-school to grade 3, TEACHER'S PET provides the young student with counting practice, letter-word recognition and three levels of math skill exercises.

SPACE TILT (Apple only) — **Price: $10.95 Cassette/$14.95 Diskette**
Use the game paddles to tilt the plane of the TV screen to "roll" a ball into a hole in the screen. Sound simple? Not when the hole gets smaller and smaller! A built-in timer allows you to measure your skill against others in this habit-forming action game.

MOVING MAZE (Apple only) — **Price: $10.95 Cassette/$14.95 Diskette**
MOVING MAZE employs the games paddles to direct a puck from one side of a maze to the other. However, the maze is dynamically (and randomly) built and is continually being modified. The objective is to cross the maze without touching (or being hit by) a wall. Scoring is by an elapsed time indicator, and three levels of play are provided.

ALPHA FIGHTER (Atari only) — **Price: $14.95 Cassette/$18.95 Diskette**
Two excellent graphics and action programs in one! ALPHA FIGHTER requires you to destroy the alien starships passing through your sector of the galaxy. ALPHA BASE is in the path of an alien UFO invasion; let five UFO's get by and the game ends. Both games require the joystick and get progressively more difficult the higher you score!

INTRUDER ALERT (Atari only) — **Price: $16.95 Cassette/$20.95 Diskette**
This is a fast paced graphics game which places you in the middle of the "Dreadstar" having just stolen its plans. The droids have been alerted and are directed to destroy you at all costs. You must find and enter your ship to escape with the plans. Five levels of difficulty are provided. INTRUDER ALERT requires a joystick and will run on 16K systems.

GIANT SLALOM (Atari only) — **Price: $14.95 Cassette/$18.95 Diskette**
This real-time action game is guaranteed addictive! Use the joystick to control your path through slalom courses consisting of both open and closed gates. Choose from different levels of difficulty, race against other players or simply take practice runs against the clock. GIANT SLALOM will run on 16K systems.

CRYSTALS (ATARI only) — **Price: $ 9.95 Cassette/$13.95 Diskette**
A unique algorithm randomly produces fascinating graphics displays accompanied with tones which vary as the patterns are built. No two patterns are the same, and the combined effect of the sound and graphics are mesmerizing. CRYSTALS has been used in local stores to demonstrate the sound and color features of the Atari.

CHOMP-OTHELLO (Atari only) — **Price: $11.95 Cassette/$15.95 Diskette**
CHOMP-OTHELLO! It's really two challenging games in one. CHOMP is similar in concept to NIM; you must bite off part of a cookie, but avoid taking the poisoned portion. OTHELLO is the popular board game set to fully utilize the Atari's graphics capability. It is also very hard to beat! This package will run on a 16K system.

GAMES PACK I (Available for all computers) — **Price: $9.95 Cassette/$13.95 Diskette**
GAMES PACK I contains the classic computer games of BLACKJACK, LUNAR LANDER, CRAPS, HORSERACE, SWITCH and more. These games have been combined into one large program for ease in loading. They are individually accessed by a convenient menu.

GAMES PACK II (Available for all computers) — **Price: $9.95 Cassette/$13.95 Diskette**
GAMES PACK II includes the games CRAZY EIGHTS, JOTTO, ACEY-DUCEY, LIFE, WUMPUS and others. As with GAMES PACK I, all the games are loaded as one program and are called from a menu.

Why pay $7.95 or more per program when you can buy a DYNACOMP collection for just $9.95?

NORTH STAR SOFTWARE EXCHANGE (NSSE) LIBRARY — **Price $9.95 Diskette**
DYNACOMP now distributes the 20+ volume NSSE library. Most of these diskettes offer an outstanding value for the purchase price. Write for details regarding the contents of this library and quantity (four or more) purchases.

Availability

DYNACOMP software is supplied with complete documentation containing clear explanations and examples. Unless otherwise specified, all programs will run within 16K program memory space (ATARI requires 24K). Except where noted, programs are available on ATARI, PET, TRS-80 (Level II) and Apple (Applesoft) cassette and diskette as well as North Star single density (double density compatible) diskette. Additionally, most programs can be obtained on standard (IBM format) 8" CP/M floppy disks for systems running under MBASIC.

BUSINESS and UTILITIES

MAIL LIST 2.2 (Apple, Atari and North Star diskette only) — **Price: $34.95**
This program is unmatched in its ability to store a maximum number of addresses on one diskette (minimum of 1100 per diskette, more than 2200 for "double density" systems!). Its many features include alphabetic and zip code sorting, label printing, merging of files and a unique keyword seeking routine which retrieves entries by a virtually limitless selection of user defined codes. A very valuable program!

FORM LETTER SYSTEM (FLS) (Apple and North Star diskette only) — **Price: $21.95**
Use FLS to create and edit form letters and address lists. Form letters are produced by automatically inserting each address into a predetermined portion of your letter. FLS is completely compatible with MAIL LIST 2.2, which may be used to manage your address files.

FLS and MAIL LIST 2.2 are available as a combined package for $49.95.

PERSONAL FINANCE SYSTEM (ATARI only) — **Price: $34.95 Diskette**
PFS is a single disk menu oriented system composed of 10 programs designed to organize and simplify your personal finances. Features include a 300 transaction capacity; fast access; 26 optional user codes; data retrieval by month, code or payee; optional printing of reports; checkbook balancing; bar graph plotting and more. Also provides on the diskette is ATARI DOS 2.

FINDIT (North Star only) — **Price: $19.95**
This is a three-in-one program which maintains information accessible by keywords of three types: Personal (eg: last name), Commercial (eg: plumbers) and Reference (eg: magazine articles, record albums, etc). In addition to keyword searches, there are birthday, anniversary and appointment searches for the personal records and appointment searches for the commercial records. Reference records are accessed by a single keyword or by cross-referencing two or three keywords.

DFILE (North Star only) — **Price: $19.95**
This handy program allows North Star users to maintain a specialized data base of all files and programs in the stack of disks which invariably accumulates. DFILE is easy to set up and use. It will organize your disks to provide efficient locating of the desired file or program.

GRAFIX (TRS-80 only) — **Price: $12.95 Cassette/$16.95 Diskette**
This unique program allows you to easily create graphics directly from the keyboard. You "draw" your figure using the program's extensive cursor controls. Once the figure is made, it is automatically appended to your BASIC program as a string variable. Draw a "happy face", call it H$ and then print it from your program using PRINT H$! This is a very easy way to create and save graphics.

TIDY (TRS-80 only) — **Price: $10.95 Cassette/$14.95 Diskette**
TIDY is an assembly language program which allows you to renumber the lines in your BASIC programs. TIDY also removes unnecessary spaces and REMark statements. The result is a compacted BASIC program which uses much less memory space and executes significantly faster. Once loaded, TIDY remains in memory; you may load any number of BASIC programs without having to reload TIDY!

STATISTICS and ENGINEERING

DATA SMOOTHER (Not available for ATARI) — **Price: $14.95 Cassette/$18.95 Diskette**
This special data smoothing program may be used to rapidly derive useful information from noisy business and engineering data which are equally spaced. The software features choice in degree and range of fit, as well as smoothed first and second derivative calculation. Also included is automatic plotting of the input data and smoothed results.

FOURIER ANALYZER (Available for all computers) — **Price: $14.95 Cassette/$18.95 Diskette**
Use this program to examine the frequency spectra of limited duration signals. The program features automatic scaling and plotting of the input data and results. Practical applications include the analysis of complicated patterns in such fields as electronics, communications and business.

TFA (Transfer Function Analyzer) — **Price: $19.95 Cassette/$23.95 Diskette**
This is a special software package which may be used to evaluate the transfer functions of systems such as hi-fi amplifiers and filters by examining their response to pulsed inputs. TFA is a major modification of FOURIER ANALYZER and contains an engineering-oriented decibel versus log-frequency plot as well as data editing features. Whereas FOURIER ANALYZER is designed for educational and scientific use, TFA is an engineering tool. Available for all computers.

HARMONIC ANALYZER (Available for all computers) — **Price: $24.95 Cassette/$28.95 Diskette**
HARMONIC ANALYZER was designed for the spectrum analysis of repetitive waveforms. Features include data file generation, editing and storage/retrieval as well as data and spectrum plotting. One particularly unique facility is that the input data need not be equally spaced or in order. The original data is sorted and a cubic spline interpolation is used to create the data file required by the FFT algorithm.

FOURIER ANALYZER, TFA and HARMONIC ANALYZER may be purchased together for a combined price of $44.95 (three cassettes) and $56.95 (three diskettes).

REGRESSION I (Available for all computers) — **Price: $19.95 Cassette/$23.95 Diskette**
REGRESSION I is a unique and exceptionally versatile one-dimensional least squares "polynomial" curve fitting program. Features include very high accuracy; an automatic degree determination option; an extensive internal library of fitting functions; data editing; automatic data and curve plotting; a statistical analysis (eg: standard deviation, correlation coefficient, etc.) and much more. In addition, new fits may be tried without reentering the data. REGRESSION I is certainly the cornerstone program in any data analysis software library.

REGRESSION II (PARAFIT) (Available for all computers) — **Price: $19.95 Cassette/$23.95 Diskette**
PARAFIT is designed to handle those cases in which the parameters are imbedded (possibly nonlinearly) in the fitting function. The user simply inserts the functional form, including the parameters (A(1), A(2), etc.) as one or more BASIC statement lines. Data and results may be manipulated and plotted as with REGRESSION I. Use REGRESSION I for polynomial fitting, and PARAFIT for those complicated functions.

MULTILINEAR REGRESSION (MLR) (Available for all computers) — **Price: $24.95 Cassette / $28.95 Diskette**
MLR is a professional software package for analyzing data sets containing two or more linearly independent variables. Besides performing the basic regression calculation, this program also provides easy to use data entry, storage, retrieval and editing functions. In addition, the user may interrogate the solution by supplying values for the independent variables. The number of variables and data size is limited only by the available memory.

REGRESSION I, II and MULTILINEAR REGRESSION may be purchased together for $49.95 (three cassettes) or $61.95 (three diskettes).

BASIC SCIENTIFIC SUBROUTINES, Volume I (Not available for ATARI)
DYNACOMP is the exclusive distributor for the software keyed to the text *BASIC Scientific Subroutines, Volume I* by F. Ruckdeschel (see the BYTE/McGraw-Hill advertisement in BYTE magazine, January 1981). These subroutines have been assembled according to chapter. Included with each collection is a menu program which selects and demonstrates each subroutine.

Collection #1: Chapters 2 and 3: Data and function plotting, complex variables
Collection #2: Chapter 4: Matrix and vector operations
Collection #3: Chapters 5 and 6: Random number generators, series approximations
Price per collection: $14.95 Cassette/$18.95 Diskette

All three collections are available for $39.95 (three cassettes) and $49.95 (three diskettes).

Because the text is a vital part of the documentation, *BASIC Scientific Subroutines, Volume I* is available from DYNACOMP for $19.95 plus 75¢ postage and handling.

ROOTS (Available for all computers) — **Price $9.95 Cassette/$13.95 Diskette**
In a nutshell, ROOTS simultaneously determines all the zeroes of a polynomial having real coefficients. There is no limit on the degree of the polynomial, and because the procedure is iterative, the accuracy is generally very good. No initial guesses are required as input, and the calculated roots are substituted back into the polynomial and the residuals displayed.

Ordering Information

All orders are processed and shipped postpaid within 48 hours. Please enclose payment with order along with computer information. If paying by VISA or Master Card, include all numbers on card. For orders outside North America add 10% for shipping and handling.

Add $2.50 to diskette price for 8" floppy disk (IBM format soft sectored, CP/M, Microsoft BASIC)
*TRS-80 diskettes are not supplied with DOS or BASIC.

Deduct 10% when ordering 3 or more programs.

Ask for **DYNACOMP** programs at your local software dealer. Write for detailed descriptions of these and other programs from **DYNACOMP**.

DYNACOMP, Inc.
1427 Monroe Avenue
Rochester, New York 14618
24 hour mail order phone: (716)586-7579
Office phone (9AM-5PM EST): (716)442-8960

VISA MasterCard

New York State residents please add 7% NYS sales tax.

⁺ *ATARI, PET, APPLE II, TRS-80, NORTH STAR, CP/M and IBM are registered trade names and/or trademarks.*

WHY BUY FROM THE BEST?

NEECO

Service... Support... Software...

CBM™ 8000 SERIES BUSINESS COMPUTERS

The new Commodore 8000 series computers offer a wide screen display to show you up to 80-character lines of information. Text editing and report formatting are faster and easier with the new wide-screen display. The 8000 series also provides a resident Operating System with expanded functional capabilities. You can use BASIC on the 8000 computers in both interactive and program modes, with expanded commands and functions for arithmetic, editing, and disk file management. The CBM 8000 series computers are ideally suited for the computing needs of the business marketplace.

SCREEN
2000 character display, organized into twenty-five 80-column lines
64 ASCII, 64 graphic characters
3 x 8 dot matrix characters
Green phosphor screen
Brightness control
Line spacing: 1½ in Text Mode
1 in Graphics Mode

KEYBOARD
73-key typewriter style keyboard with graphic capabilities
Repeat key functional with all keys

MEMORY
CBM 8016: 16K (15359 net) random access memory (RAM)
CBM 8032: 32K (31743 net) random access memory (RAM)

POWER REQUIREMENTS
Volts: 110V
Cycles: 60 Hz
Watts: 100

SCREEN EDITING CAPABILITIES
Full cursor control (up, down, right, left)
Character insert and delete
Reverse character fields
Overstriking
Return key sends entire line to CPU regardless of cursor position

INPUT/OUTPUT
Parallel port
IEEE-488 bus
2 cassette ports
Memory and I/O expansion connectors

FIRMWARE
24K or ROM contains:
BASIC (version 4.0) with direct (interactive) and indirect (program) modes
9-digit floating binary arithmetic
Tape and disk file handling software

CBM 8032 Computer $1795

CBM™ 8050 DUAL DRIVE FLOPPY DISK

The CBM 8050 Dual Drive Floppy Disk in an enhanced version of the intelligent CBM 2040 Disk Drive. The CBM 8050 has all of the features of the CBM 2040, and provides more powerful software capabilities, as well as nearly one megabyte of online storage capacity. The CBM 8050 supplies relative record files and automatic diskette initialization. It can copy all the files from one diskette to another without copying unused space. The CBM 8050 also offers improved error recovery and the ability to append to sequential files.

HARDWARE SPECIFICATIONS
Dual Drives
Two microprocessors
974K Bytes storage on two 5.25" diskettes (ss)
Tracks 70
Sectors 17-21
Soft sector format
IEEE-488 interface
Combination power (green) and error (red) indicator lights
Drive Activity indicator lights
Disk Operating System Firmware (12K ROM)
Disk Buffer (4K RAM)

FIRMWARE
DOS version 2.0
Sequential file manipulation
Sequential user files
Relative record files
Append to sequential files
Improved error recovery
Automatic diskette initialization
Automatic directory search
Command parser for syntax validation
Program load and save

CBM 8050 Dual Price $1795

CBM	PRODUCT DESCRIPTION	PRICE
4016	16K RAM-Graphics(N) or Business(B) Keyboard	$ 995.00
4032	32K RAM-Graphics(N) or Business(B) Keyboard	$1295.00
8032	32K RAM-80 Col. Screen-Business Keyboard	$1795.00
4022	Tractor Feed Printer	$ 795.00
4040	Dual Floppy-343K-DOS 2.0	$1295.00
8050	Dual Floppy-974K-DOS 2.0	$1795.00
4010	Voice Synthesizer	$ 395.00
8010	300 Baud IEEE Modem	$ 279.95
C2N Cassette	External Cassette Drive	$ 95.00
CBMto IEEE	CBM to 1st IEEE Peripheral	$ 39.95
IEEE to IEEE	IEEE to 2nd IEEE Peripheral	$ 49.95
2.1 DOS	DOS Upgrade for 2040	$ 100.00
4.0 DOS	O/S Upgrade for 40 Column Computer	$ 100.00
Word Pro 4+	Word Processing Software used w/8032	$ 450.00

*CBM is a registered trademark of Commodore. All prices and specifications are subject to change without notice.

commodore

NEECO

679 HIGHLAND AVE.
NEEDHAM, MA 02194

NEECO WILL MATCH ANY ADVERTISED PRICE ON CBM EQUIPMENT.

(617) 449-1760
Telex: 951021

MON-FRI 9:00 - 5:00
MasterCharge and VISA Accepted

NEECO announces the MIPLOT Plotter from WATANABE

MIPLOT WX4671

	Code(ASCII)	Name	Function
Vector commands	D	DRAW	Draw a straight line to the point specified by absolute coordinates.
	I	RELATIVE DRAW	Draw a straight line to the point specified by relative coordinates.
	M	MOVE	Move with pen up to the point specified by absolute coordinates.
	R	RELATIVE MOVE	Move with pen up to the point specified by relative coordinates.
	L	LINE TYPE	Specify solid or broken line.
	B	LINE SCALE	Specify the pitch of a broken line (0.1 – 12.7mm).
	X	AXIS	Draw X or Y coordinate axis.
	H	HOME	Return to the origin with the pen up.
Character commands	S	ALPHA SCALE	Specify character size (1 to 16 times basic 0.7mm x 0.4mm)
	Q	ALPHA ROTATE	Specify character orientation. (Four directions)
	P	PRINT	Draw ASCII code characters.
	N	MARK	Draw mark centered on the pen position. (Six kinds)

- 26 cm X 36 cm Plotting area
- .1 mm Addressing Resolution
- Full ASCII in any of 15 different character sizes
- Uses standard felt tip pens

$1250.00*

*Includes interface to CBM, Atari®, Apple, or TRS•80**. Please specify.

Please CALL or WRITE for specifications and information.

®Atari is a registered trademark
**TRS•80 is a registered trademark of Tandy Corp.

ORIGINAL 8K PET 2001* OWNERS TAKE NOTE!

The following peripherals and accessories are IN STOCK AT NEECO:

1. AXIOM PRINTER

- Complete PET graphics
- Plug compatible
- Electrostatic paper
- 40 or 80 columns

$299.00

2. 16 or 24K EXPANDAMEM

INTERNAL MEMORY EXPANSION UNIT

- Plug compatible
- Dynamic low heat memory
- Proven reliability
- No adaptor needed

16K - $299.00
24K - $379.00

3. FULL SIZE KEYBOARD

- Complete PET graphics
- Separate keypad
- Plug compatible
- With cover

$99.95

*8K-2001 with original keyboard and built-in cassette

NEECO
879 HIGHLAND AVE.
NEEDHAM, MA 02194

NEW ENGLAND ELECTRONICS CO., INC.
"NEW ENGLAND's Largest Computer Showroom"

(617) 449-1760
MASTERCHARGE OR VISA ACCEPTED
TELEX NUMBER 951021, NEECO
MON-FRI. 9:00-5:30. E.S.T.

Land of the Lost

A Program For A Cassette Filing System

Steve Michel
Sterling, IL 61081

One day I pushed myself back from the green glow of the PET CRT and was struck by a fact that has been apparent to my family (translated here as wife) for quite some time. My office had become a jungle of little white plastic cases.

The major source of confusion was my cassette filing system. I HAD NONE!! There were some 200 plus programs strewn around on 100 plus C-10 cassettes. (I still drool over ads for floppy disk drives.) The disarray of cassettes was not so much a bother as was my MTBF. In most computer circles that stands for Mean Time Between Failure. In my case it stood for *Mean Time Between Finding.* It usually took me 2-3 times longer to find a particular program than it did to LOAD it. I decided it was a case of survival – find my way out now or be forever lost among those sequential magnetic I/O storage devices.

The ultimate solution was two pronged. The first step was to place each program into one of three categories:

1) EDUCATIONAL – I teach high school science.
2) UTILITIES – renumber, merger, business applications
3) GAMES – Need I say anything here?

These classifications covered the range of my programs fairly well.

The groups were then placed into appropriately labeled boxes. I have found that the boxes used to package those self-adhesive mailing labels that arrive on so much of our mail are an ideal size. They are exactly the right width and will hold about 15 cassettes. I get my boxes from a local industry that sends out mass mailings. The DP manager was more than happy to provide the empty boxes.

The last step in finding my way out of this "cassette block" was to devise a method for cataloging the programs, providing a short description of each, updating these as necessary and producing a final listing of the library contents. This effort resulted in the following program.

I tried to take an example from some of the larger computer systems and wrote a menu-driven program. This means that the operator is given a display on the screen which lists various options that can be selected by the pressing of a single key. After the option is complete, the user is then returned to the same or another menu to make another selection.

...It usually took me 2–3 times longer to find a particular program than it did to LOAD it...

The main advantage of this type of approach is that it allows people with little or no computer experience to feel comfortable and confident about running a particular job. It also cuts down on the chance of operator error because of the reduced input requirements.

PROGRAM DISSECTION:

Variables Used:

E$,U$,G$ ------ arrays that hold program names and the description of the programs
EX$ ------------- array used to LOAD and SAVE each of the individual categories
NM -------------- holds the total number of records LOADED or SAVED in each category
F ----------------- F 0 – return to SAVE MENU
F 1 – return to LOAD MENU
EN -------------- number of entry currently being edited
II ---------------- position in string that is beinng entered or edited
EE, EU, EG ---- number of titles *entered* from the keyboard for each category
LE, LU, LG ---- number of titles *loaded* from cassette file for each category
DN -------------- devise number on which final printed output will appear

Program Segments:

100 -------------- sets array sizes
105–1016 ------- **MAIN MENU**-listing of options
1. enter data from keyboard
2. save data file to cassette
3. load data file from cassette
4. print listing of titles
5. edit any previously entered data
6. end program

2000–2136 ----- **EXCHANGE ROUTINE**-this routine is used just prior to the SAVE routine which employs the general variable EX$-each category is transferred into EX$ before SAVE-ing.

3000–2136 ----- **SORT ROUTINE**-this is used to sort each category before it is saved to tape. It is a quick sort taken from **COMPUTE!**, issue 2, pg. 12.

4000–4391 ----- **EDIT ROUTINE**-this section allows any previously entered data to be reviewed or corrected.
It displays the entry and cursor by use of the cursor left and cursor right keys. Corrections are made by typing over the existing entry. No provisions were made for the insertion or deletion of characters.

5000–5331 ----- **INPUT ROUTINE**-this is the heart of the program. It provides both the enter data and edit data function. I first started with an INPUT statement here but it wasn't flexible enough (or I wasn't smart enough) to accomodate the edit function. The routine gets one character at a time, checks the value of the character, branches to appropriate routines for cursor control, or adds the character to the correct spot in the input string. The routine wraps around both forward and backward and allows up to 75 characters per entry. A more detailed explanation follows:

- **5000** ------- clears keyboard buffer of accidental keystrokes
- **5010** ------- prints flashing underline cursor
- **5015** ------- get keystroke
- **5020** ------- checks for cursor left key
- **5025** ------- increments position of cursor in string
- **5030** ------- checks for cursor right key
- **5040** ------- checks for wrap around in forward direction
- **5090-5100**- prints character on screen and adds it to the correct position in the string being entered or edited
- **5300-5330**- performs cursor left and reverse wrap around

10000-16031 -- **ENTER DATA MENU**

19000-20121 -- **SAVE ROUTINE**-saves the selected group from previously saved file. Also gives messages for tape handling.

30000-32041 -- **PRINT ROUTINE**-prints list of selected titles. User defines whether output is to screen or to printer in lines 42000-42060. Devise number 4 for printer, 3 for screen. The output file is then opened to the correct devise in line 40075, the file is printed and then closed in 41002. Figure 1 shows a sample of the output.

A few words about program modifications. The LOAD and SAVE routines should be easily modified to accomodate those lucky disk users out there. I strongly recommend that all REMs be omitted from the program when typing in because they take up an extra 2.5K of memory.

Well, that is the way it works. Some corollary, somewhere, must say, "It always looks easier after its done." At this point, all of my programs are neatly stacked in 5 well labeled boxes, every program has been backed up on a master tape, every program has its listing filed in an appropriate folder and I have an alphabetical list and description of every program in my library. It feels great to be back in civilization again.

```
5 REM    STEVE MICHEL
10 REM    STERLING HIGH SCHOOL
15 REM    STERLING IL    61081
20 REM
100 DIMG$(150),E$(150),U$(150),EX$(150)
105 PRINT"ħ↓↓              ɼMAIN MENUf̂↓↓"
110 PRINT"ɼ1f̂.→ENTER DATA          ɼ2f̂. ¬
     ¬SAVE FILE"
112 PRINT"↓ɼ3f̂. LOAD FILE          ɼ4f̂. ¬
     ¬PRINT LIST"
113 PRINT"↓ɼ5f̂. EDIT TITLES        ɼ6f̂. ¬
     ¬END PROGRAM"
115 GETA$:IFA$=""THENGOSUB1000:GOTO115
116 A=VAL(A$)
120 ONAGOTO10000,19000,30000,40000,4000,
     ¬18000
125 GOTO115
1000 TT=TT+1
1002 IFTT/2=INT(TT/2)THENR$="ɼ":GOTO1010
1005 R$="f̂"
1010 PRINT"ħ↓↓↓↓↓↓↓↓↓↓↓↓↓↓              ¬
     ¬";R$;"ENTER CHOICE"
1015 FORJ=1TO500:NEXTJ:RETURN
2000 A=VAL(A$):ONAGOTO2010,2020,2030
2010 FORJ=1TONM:EX$(J)=E$(J):NEXTJ:
     ¬GOSUB3000:FORJ=1TONM:E$(J)=EX$(J):
     ¬NEXTJ
2015 GOTO2100
2020 FORJ=1TONM:EX$(J)=U$(J):NEXTJ:
     ¬GOSUB3000:FORJ=1TONM:U$(J)=EX$(J):
     ¬NEXTJ
2025 GOTO2100
2030 FORJ=1TONM:EX$(J)=G$(J):NEXTJ:
     ¬GOSUB3000:FORJ=1TONM:G$(J)=EX$(J):
     ¬NEXTJ
2100 PRINT"ħ↓↓  REWIND DATA TAPE ¬
     ¬ɼ";NM$"f̂."
2105 PRINT"↓↓  PRESS '*' TO RETURN TO ¬
     ¬MENU."
2110 PRINT"↓↓  PRESS ANY KEY WHEN DONE."
2120 GETA$:IFA$=""THEN2120
2130 IFA$="*"THENIFF=0THEN19000
2133 IFA$="*"THENIFF=1THENF=0:GOTO30000
2134 IFF=1THENF=0:GOTO31000
2135 GOTO20020
3000 PRINT"ħ↓↓NOW SORTING ";NM$;" ¬
     ¬PROGRAMS."
3100 TP=1:LOWER(1)=1:UPPER(1)=NM
3120 IFTP<=0THENRETURN
3140 LB=LOWER(TP):UB=UPPER(TP):TP=TP-1
3160 IFUB<=LBTHEN3120
3180 I=LB:J=UB:TEMP$=EX$(I)
3200 IFJ<1THEN3260
3220 IFTEMP$>=EX$(J)THEN3260
3240 J=J-1:GOTO3200
3260 IFJ<=ITHENEX$(I)=TEMP$:GOTO3400
3280 EX$(I)=EX$(J):I=I+1
3300 IFI>NMTHEN3360
3320 IFEX$(I)>=TEMP$THEN3360
3340 I=I+1:GOTO3300
3360 IFJ>ITHENEX$(J)=EX$(I):J=J-1:
     ¬GOTO3220
3380 EX$(J)=TEMP$:I=J
3400 TP=TP+1
3420 IFI-LB<UB-ITHENLOWER(TP)=I+1:
     ¬UPPER(TP)=UB:UB=I-1:GOTO3160
3440 LOWER(TP)=LB:UPPER(TP)=I-1:LB=I+1
3460 GOTO3160
4000 PRINT"ħ↓↓         ɼEDIT FILE MENUf̂"
4010 PRINT"↓↓EDIT WHICH SET OF PROGRAM ¬
     ¬TITLES?"
4020 PRINT"↓ɼ1f̂. EDUCATIONAL      ɼ2f̂. ¬
     ¬UTILITIES"
4030 PRINT"↓ɼ3f̂. GAMES            ɼ4f̂. ¬
     ¬MAIN MENU"
4050 GETA$:IFA$=""THENGOSUB1000:GOTO4050
4060 A=VAL(A$)
4070 ONAGOTO4100,4200,4300,105
4080 GOTO4050
4100 PRINT"ħ↓↓WHICH EDUCATIONAL TITLE ¬
     ¬TO EDIT ?"
4110 INPUT"NUMBER";EN
4115 PRINT"ħ    1    5    10    15    20    ¬
     ¬25    30    35"
```

```
4120 PRINT"    ";LEFT$(E$(EN),35)
4125 LR=LEN(E$(EN)):IFLR>35THENPRINT"↓";
   ¬RIGHT$(E$(EN),LR-35)
4130 PRINT"h↓↓↓↓↓↓HERE IS THE LINE AS ¬
   ¬IT EXISTS NOW."
4140 PRINT"↓SIMPLY EDIT OVER THE ¬
   ¬MISTAKES AND"
4150 PRINT"PRESS RETURN WHEN DONE."
4160 PRINT"h":PRINT">>>";:B$=E$(EN):
   ¬II=0:GOSUB5007:T$=B$
4170 PRINT"h↓↓↓↓↓↓↓↓↓↓↓↓↓IT THAT CORRECT ¬
   ¬NOW r(Y OR N)?"
4175 GETA$:IFA$=""THEN4175
4180 IFA$="Y"THENE$(EN)=T$:GOTO4000
4185 IFA$="N"THEN4115
4190 GOTO4175
4200 PRINT"ĥ↓↓WHICH UTILITY TITLE TO ¬
   ¬EDIT ?"
4210 INPUT"NUMBER";EN
4215 PRINT"ĥ    1    5    10    15    20   ¬
   ¬25   30   35"
4220 PRINT"    ";LEFT$(U$(EN),35)
4225 LX=LEN(U$(EN)):IFLX>35THENPRINT"↓";
   ¬RIGHT$(U$(EN),LX-35)
4230 PRINT"h↓↓↓↓↓↓HERE IS THE LINE AS ¬
   ¬IT EXISTS NOW."
4240 PRINT"↓SIMPLY EDIT OVER THE ¬
   ¬MISTAKES AND"
4250 PRINT"PRESS RETURN WHEN DONE."
4260 PRINT"h":PRINT">>>";:B$=U$(EN):
   ¬II=0:GOSUB5007:T$=B$
4270 PRINT"h↓↓↓↓↓↓↓↓↓↓↓↓↓IT THAT CORRECT ¬
   ¬NOW r(Y OR N)?"
4275 GETA$:IFA$=""THEN4275
4280 IFA$="Y"THENU$(EN)=T$:GOTO4000
4285 IFA$="N"THEN4215
4290 GOTO4275
4300 PRINT"ĥ↓↓WHICH GAME TITLE TO EDIT ¬
   ¬?"
4310 INPUT"NUMBER";EN
4315 PRINT"ĥ    1    5    10    15    20   ¬
   ¬25   30   35"
4320 PRINT"    ";LEFT$(G$(EN),35)
4325 LR=LEN(G$(EN)):IFLR>35THENPRINT"↓";
   ¬RIGHT$(G$(EN),LR-35)
4330 PRINT"h↓↓↓↓↓↓HERE IS THE LINE AS ¬
   ¬IT EXISTS NOW."
4340 PRINT"↓SIMPLY EDIT OVER THE ¬
   ¬MISTAKES AND"
4350 PRINT"PRESS RETURN WHEN DONE."
4360 PRINT"h":PRINT">>>";:B$=G$(EN):
   ¬II=0:GOSUB5007:T$=B$
4370 PRINT"h↓↓↓↓↓↓↓↓↓↓↓↓↓IT THAT CORRECT ¬
   ¬NOW r(Y OR N)?"
4375 GETA$:IFA$=""THEN4375
4380 IFA$="Y"THENG$(EN)=T$:GOTO4000
4385 IFA$="N"THEN4315
4390 GOTO4375
5000 GETA$:IFA$<>""THEN5000
5005 II=0:B$=""
5007 PRINT">";
5010 PRINT"↓#↑";:FORI=1TO30:NEXTI:
   ¬PRINT"↓< <↑";
5015 GETA$:IFA$=""THEN5010
5020 A=ASC(A$):IFA=157THEN5300
5025 II=II+1
5030 IFA=29THENPRINT">";:GOTO5040
5035 GOTO5050
5040 IFII=35THENPRINT:PRINT
5045 GOTO5010
5050 IFA=13ORA=141THENPRINT:RETURN
5070 IFA=160THENA=32:A$=CHR$(A)
5080 IF(A>95ANDA<160)OR(A<32)OR(A=34)OR(
   ¬A=20)THENII=II-1:GOTO5010
5090 IFII=1THENPRINTA$;:B$=A$+MID$(B$,2,
   ¬LEN(B$)):GOTO5010
5095 IFII=75THENPRINTA$:B$=B$+A$:RETURN
5100 PRINTA$;:B$=LEFT$(B$,II-1)+A$+MID$(
   ¬B$,II+1,LEN(B$))
5110 IFII=35THENPRINT:PRINT
5120 GOTO5010
5300 IFII=0THEN5010
5310 II=II-1
5320 IFII=34THENPRINT"<↑";
5330 PRINT"<";:GOTO5010
10000 PRINT"ĥ↓↓          rENTER DATA ¬
   ¬MENUr̂↓↓"
10010 PRINT"↓↓↓r1r̂. EDUCATIONAL      ¬
   ¬r2r̂. UTILITIES"
10020 PRINT"↓r3r̂. GAMES             r4r̂. ¬
   ¬MAIN MENU"
10030 GETA$:IFA$=""THENGOSUB1000:
   ¬GOTO10030
10035 A1=VAL(A$)
10040 ONA1GOTO10100,14000,16000,105
10045 GOTO10030
10100 PRINT"ĥ↓↓ENTER ED. PROGRAMS (0 ¬
   ¬WHEN DONE)"
10103 PRINT"h↓↓↓    1    5    10    15   ¬
   ¬20   25   30   35"
10105 EE=EE+1
10110 PRINTEE+LE;:GOSUB5000:E$=B$
10115 IFE$="0"THENEE=EE-1:GOTO105
10117 IFE$="*"THENEE=EE-1:GOTO10110
10120 E$(EE+LE)=E$
13000 GOTO10105
14000 PRINT"ĥ↓↓ENTER UTILITY PROGS. (0 ¬
   ¬WHEN DONE)"
14003 PRINT"ĥ    1    5    10    15    20   ¬
   ¬ 25   30   35"
14005 EU=EU+1
14010 PRINTEU+LU;:GOSUB5000:U$=B$
14015 IFU$="0"THENEU=EU-1:GOTO105
14017 IFU$="*"THENEU=EU-1:GOTO14010
14020 U$(EU+LU)=U$
14103 PRINT"h↓↓↓    1    5    10    15   ¬
   ¬20   25   30   35"
15000 GOTO14005
16000 PRINT"ĥ↓↓ENTER GAME PROGRAMS  (0 ¬
   ¬WHEN DONE)"
16003 PRINT"ĥ    1    5    10    15    20   ¬
   ¬ 25   30   35"
16005 EG=EG+1
16010 PRINTEG+LU;:GOSUB5000:G$=B$
16015 IFG$="0"THENEG=EG-1:GOTO105
16017 IFG$="*"THENEG=EG-1:GOTO16010
16020 G$(EG+LG)=G$
16030 GOTO16005
16103 PRINT"h↓↓↓    1    5    10    15   ¬
   ¬20   25   30   35"
18000 PRINT"ĥ↓↓↓WANT TO SAVE YOUR DATA ¬
   ¬FIRST(Y OR N)?"
18020 GETA$:IFA$=""THEN18020
18030 IFA$="Y"THEN19000
18040 IFA$="N"THENEND
18050 GOTO18020
19000 PRINT"ĥ↓↓          rSAVE FILE MENUr̂"
20000 PRINT"↓↓SAVE WHICH SET OF PROGRAM ¬
   ¬TITLES?"
20001 PRINT"↓r1r̂. EDUCATIONAL     r2r̂. ¬
   ¬UTILITIES"
```

```
20003 PRINT"↓ṟ3r̂. GAMES              ṟ4r̂. ¬
      ¬MAIN MENU"
20004 GETA$:IFA$=""THENGOSUB1000:
      ¬GOTO20004
20005 A=VAL(A$)
20006 ONAGOTO20010,20014,20017,105
20007 GOTO20004
20010 IFA$="1"THENNM$="EDUCATIONAL":
      ¬NM=EE+LE:GOTO2000
20014 IFA$="2"THENNM$="UTILITIES":
      ¬NM=EU+LU:GOTO2000
20017 IFA$="3"THENNM$="GAMES":NM=EG+LG:
      ¬GOTO2000
20019 GOTO20004
20020 OPEN1,1,1,NM$
20022 PRINT"↓↓NOW WRITING ";NM$;" FILE."
20025 PRINT#1,NM
20030 FORQ=1TONM
20050 PRINT#1,EX$(Q)
20060 NEXTQ
20070 CLOSE1
20080 PRINT"ĥ↓↓ A TOTAL OF";NM;NM$;" ¬
      ¬TITLES WERE"
20090 PRINT"↓SAVED."
20100 PRINT"↓↓PRESS ANY KEY TO RETURN ¬
      ¬TO MAIN MENU"
20110 GETA$:IFA$=""THEN20110
20120 GOTO105
30000 PRINT"ĥ↓↓          ṟLOAD FILE ¬
      ¬MENUr̂↓↓"
30010 PRINT"↓↓LOAD WHICH SET OF PROGRAM ¬
      ¬TITLES?"
30015 PRINT"↓ṟ1r̂. EDUCATIONAL       ṟ2r̂. ¬
      ¬UTILITIES"
30020 PRINT"↓ṟ3r̂. GAMES             ṟ4r̂. ¬
      ¬MAIN MENU"
30030 GETL$:IFL$=""THENGOSUB1000:
      ¬GOTO30030
30035 L=VAL(L$)
30036 ONLGOTO30040,30050,30060,105
30037 GOTO30030
30040 IFL$="1"THENNM$="EDUCATIONAL":
      ¬LE=0:F=1:GOTO2100
30050 IFL$="2"THENNM$="UTILITIES":LU=0:
      ¬F=1:GOTO2100
30060 IFL$="3"THENNM$="GAMES":LG=0:F=1:
      ¬GOTO2100
30070 GOTO30030
31000 OPEN1,1,0,NM$
31005 PRINT"↓↓FOUND ";NM$;".  NOW ¬
      ¬LOADING."
31010 INPUT#1,NM
31020 FORJ=1TONM
31030 INPUT#1,EX$(J)
31040 NEXTJ
31045 CLOSE1
31050 ONLGOTO31060,31070,31080
31060 LE=NM:FORJ=1TONM:E$(J+EE)=EX$(J):
      ¬NEXTJ:GOTO32000
31070 LU=NM:FORJ=1TONM:U$(J+EU)=EX$(J):
      ¬NEXTJ:GOTO32000
31080 LG=NM:FORJ=1TONM:G$(J+EG)=EX$(J):
      ¬NEXTJ
32000 PRINT"ĥ↓↓A TOTAL OF ";NM;NM$;" ¬
      ¬TITLES WERE"
32010 PRINT"↓LOADED."
32020 PRINT"↓↓    PRESS ANY KEY TO ¬
      ¬CONTINUE."
32030 GETA$:IFA$=""THEN32030
32040 GOTO105
40000 PRINT"ĥ↓↓HAVE YOU SAVED THE ¬
      ¬TITLES ON TAPE(YORN)?"
40010 GETA$:IFA$=""THEN40010
40020 IFA$="Y"THEN42000
40025 IFA$="N"THEN40030
40029 GOTO40010
40030 PRINT"↓↓ THEY NEED TO BE SAVED ¬
      ¬(SORTED) BEFORE"
40035 PRINT"↓THEY CAN BE PRINTED. SAVE ¬
      ¬THEM FIRST!!"
40040 PRINT"↓↓ PRESS ANY KEY TO GO TO ¬
      ¬SAVE FILE MENU."
40045 GETA$:IFA$=""THEN40045
40049 GOTO19000
40050 PRINT"ĥ↓↓          ṟPRINT LIST ¬
      ¬MENUr̂↓↓"
40060 PRINT"↓↓↓ṟ1r̂. EDUCATIONAL        ¬
      ¬ṟ2r̂. UTILITIES"
40070 PRINT"↓ṟ3r̂. GAMES  "
40075 OPEN1,DN,1
40080 GETA$:IFA$=""THENGOSUB1000:
      ¬GOTO40080
40085 A1=VAL(A$)
40090 ONA1GOTO40100,40200,40300
40095 GOTO40080
40100 PRINT#1,"ĥ          EDUCATIONAL ¬
      ¬TITLES"
40105 PRINT#1:PRINT#1
40110 FORJ=1TOEE+LE
40120 PRINT#1,MID$(STR$(J),2);". ¬
      ¬";E$(J):NEXTJ:GOTO41000
40200 PRINT#1,"ĥ          UTILITIES ¬
      ¬TITLES"
40205 PRINT#1:PRINT#1
40210 FORJ=1TOEU+LU
40220 PRINT#1,MID$(STR$(J),2);". ¬
      ¬";U$(J):NEXTJ:GOTO41000
40300 PRINT#1,"ĥ              GAMES ¬
      ¬TITLES"
40305 PRINT#1:PRINT#1
40310 FORJ=1TOEG+LG
40320 PRINT#1,MID$(STR$(J),2);". ¬
      ¬";G$(J):NEXTJ:GOTO41000
41000 PRINT#1:PRINT#1:PRINT#1,"SM = ¬
      ¬STEVE MICHEL     CC = CREATIVE ¬
      ¬COMPUTING"
41002 CLOSE1,DN,1
41005 PRINT"↓↓↓ PRESS ANY KEY TO RETURN ¬
      ¬TO MAIN MENU"
41010 GETA$:IFA$=""THEN41010
41020 GOTO105
42000 PRINT"ĥ↓↓↓↓SELECT OUTPUT DEVICE ¬
      ¬DESIRED"
42010 PRINT"↓↓    ṟPr̂RINTER"
42020 PRINT"↓↓    ṟSr̂CREEN"
42030 GETA$:IFA$=""THEN42030
42040 IFA$="P"THENDN=4:GOTO40050
42050 IFA$="S"THENDN=3:GOTO40050
42060 GOTO42030
```

©

Odds & Ends on the 2040 Disk

Jim Butterfield

WARNING: If you get an unclosed file — which shows up with an asterisk on the directory — do not scratch it; you may harm other files. Instead, do a Verify (called COLLECT on 4.0 systems). ©

Using The 6522 to drive a Printer

Edward H. Carlson
Okemos, MI

Low price compatible with good quality. If you are reaching the edge of your budget, the fifty dollars you can save by buying the parallel version of a printer may loom large. I wanted a printer for word processing and chose the Comprint 912P as suitable for rough draft printing. I was confident that the 6522 VIA on the CPU board of my Ohio Scientific C2-4P could handle the parallel interfacing. VIA stands for Versatile Interface Adaptor, and it can easily be configured to handle all the handshaking involved in the parallel transfer of data.

This article will describe how to wire the 6522 to the printer and will give a machine language program to drive it. The discussion is not at all restricted to OSI computers, nor even to the Comprint printer since the same principles apply to interfacing to other printers.

You may be interested in the features of the Comprint that appealed to me for word processing. It is fast, quiet and simple in design. The letter quality is high for a dot matrix printer as it has a 9x12 matrix. It is quiet because it is an electrostatic printer. This technology uses rolls of black paper which are coated with aluminum. The print head sparks holes through the aluminum to expose the black color below. The silvery paper is low in cost, thin and somewhat of a nuisance to handle. However, it Xeroxes very well. The 912 prints 3 lines a second of 80 characters each.

The Comprint has a variety of parallel options including the IEEE-488 convention and both wide and narrow strobe modes. I purchased the Comprint soon after it appeared on the market and made the modifications they suggested to operate with the Apple II Parallel Interface Card. (Since I also have an Apple, the same printer serves both computers.) The signal lines into the printer include seven parallel lines for the ASCII data and one line for DAV which is a narrow (one clock cycle is enough) strobe that tells the printer when valid data is on the 7 line bus. Signal lines from the Comprint include NDAC which goes low to acknowledge that the printer has accepted the character, and NRFD (not ready for data) which goes high when the printer's data buffer is full.

The 6522 VIA has two 8-bit ports, A and B, each with two control lines. The two ports are not identical and for no good reason I use the B port for the seven line ASCII bus. Since the eighth line is not needed for ASCII, I use it for the "busy" signal (NRFD). The B port control lines CB1 and CB2 are used for NDAC and DAV respectively.

The listing shows a subroutine, OUTCHR, that prints one character. Also included is a DRIVER that uses some subroutines in the OSI BASIC ROM's to read tape so its contents can be sent to the printer. Of course, this driver will need to be altered if your computer is not an OSI machine.

Implementing a 6522 can be a frustrating experience because of its many options. It has 16 registers of which we need 5. Three of the registers need be set only once. but we have plenty of time per character, and it is simpler to set these registers each time the subroutine is entered. Line 160

...implementing a 6522 can be a frustrating experience because of its many options...

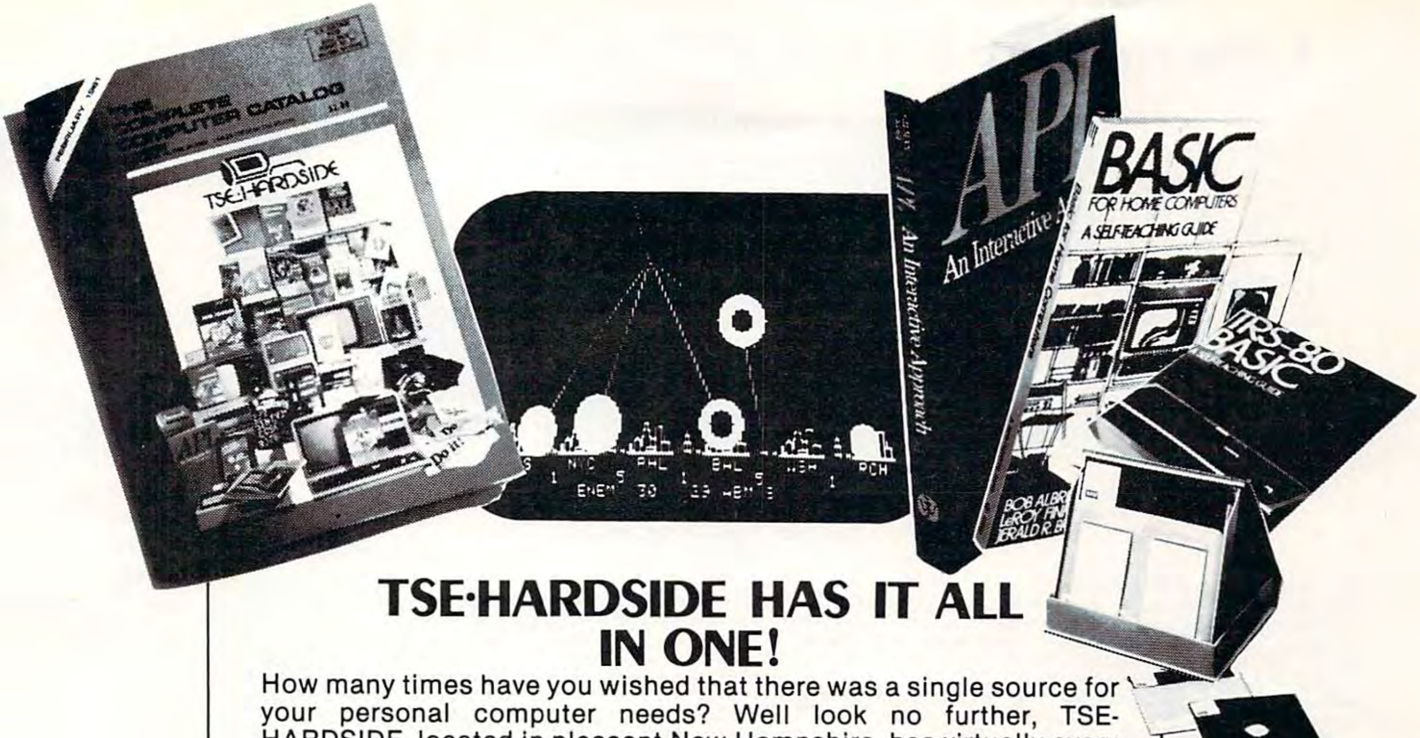
enables the B port by setting bit 1 in the Auxiliary Control Register. In line 170, the Data Direction Register for B port is loaded such that lines 0 to 6 are output (for the ASCII character) and line 7 as input (for the DAV signal). Finally, the Peripheral Control Register must be tickled so that CB1 and CB2 know what is expected of them. This is done in line 210. Bits 7, 6, 5 are set to 100 so that CB2 will pulse low when the CPU writes to the VIA, (the strobe). Setting bit 4 tells the VIA to raise a flag when CB1 makes a low to high transition (the acknowledgement).

When the subroutine is entered, the accumulator A holds the character to be printed. It is saved by pushing it on the stack. Then the three registers mentioned above are configured. Next the VIA looks for the "busy" signal in lines 220 to 240. Upon finding a non-busy status, the character is pulled from the stack and sent to the B Output Register, and on to the printer. The last event is to detect the DAV acknowledgement. When it comes in on CB1, it sets a flag in the Interrupt Flag Register. Detecting this flag allows an exit from the loop of lines 300 to 330, and then exit from the subroutine.

There you have it. If you are interfacing to some other printer, the main thing to watch for is the polarity of the signal lines. Consult your 6522 data sheets for the code needed to reverse the polarity of the handshake signals. If by chance you have a Comprint 912P and have not configured it for Apple compatibility, I have written a program for that case too. An article describing it has been accepted for publication by BYTE. A copy of the program may be obtained by writing me at 3872 Raleigh Drive, Okemos, MI, 48864.

```
  1 0000              ;        ***   TAPE TO COMPRINT 912P    ***
  2 0000              ;
 10 C000               *       =$C000
 20 C000 2007BF  DRIVER JSR $BF07       GET CHAR. FROM TAPE PORT
 25 C003 8D00D2         STA $D200       STORE CHAR. ON SCREEN
 30 C006 200CC0         JSR OUTCHR      PRINT CHAR.
 40 C009 4C00C0         JMP DRIVER
 41 C00C              ;
 42 C00C              ;        MY ADDRESSES, SEE FOOTNOTE
 43 C00C              ;
 44 C00C               VIA   =$F700 ADDRESS OF 6522 IS $F7XX
 46 C00C               AUX   =$0E   AUXILIARY CTRL REGISTER
 48 C00C               BDD   =$08   B DATA DIRECTION REGISTER
 50 C00C               BPORT =$00   OUTPUT REGISTER FOR I/O PORT B
 52 C00C               PCTRL =$03   PERIPHERAL CONTROL REGISTER
 54 C00C               IFLAG =$07   INTERRUPT FLAG REGISTER
 60 C00C              ;
 61 C00C              ;        STANDARD ADDRESSES
 63 C00C              ;
 64 C00C              ; VIA     PER YOUR MACHINE
 66 C00C              ; AUX     =%1011
 68 C00C              ; BDD     =%0010
 70 C00C              ; BPORT   =%0000
 72 C00C              ; PCTRL   =%1100
 74 C00C              ; IFLAG   =%1101
134 C00C              ;
140 C00C 48      OUTCHR PHA                 A CONTAINS CHARACTER
150 C00D A902           LDA #%00000010  ENABLE B PORT OF 6522
160 C00F 8D0EF7         STA VIA+AUX     AUX CTRL REGISTER
170 C012 A97F           LDA #%01111111  DATA DIRECTION
180 C014 8D08F7         STA VIA+BDD     B PORT DATA DIR REGISTER
190 C017 8D07F7         STA VIA+IFLAG   CLEAR INTERRUPT FLAGS
200 C01A A9B0           LDA #%10110000  PREPARE CB1 AND CB2
210 C01C 8D03F7         STA VIA+PCTRL   CB2 IS STROBE, PULSES LO
220 C01F AD00F7  BUSY   LDA VIA+BPORT   READ B PORT INPUT
230 C022 2980           AND #%10000000  BIT 7 IS NRFD OF COMPRINT
240 C024 30F9           BMI BUSY        BUSY IF BIT 7 IS HI
250 C026 68             PLA             LOAD CHAR. IN A
270 C027 8D00F7         STA VIA+BPORT   OUTPUT TO PRINTER
300 C02A AD07F7  ACK    LDA VIA+IFLAG   LOOK FOR NDAC ON CB1
310 C02D 2910           AND #%00010000  MASK OUT DESIRED FLAG
320 C02F C910           CMP #%00010000  NDAC IS ACKNOWLEDGE
330 C031 D0F7           BNE ACK         IF NOT FOUND, LOOK AGAIN
340 C033 60             RTS
350 C034              ;
400 C034              ; COMPRINT PARALLEL I/O BOARD (PBC 1184 Rev C)
405 C034              ;   HAS BEEN MODIFIED TO OPERATE WITH THE APPLE II
410 C034              ;   PARALLEL PRINTER INTERFACE CARD
415 C034              ;
420 C034              ; THE 6522 HAS ADDRESS LINES 0,1 CONNECTED TO
422 C034              ;   ADDRESSES 2,3 AND VICE VERSA
```

©



TSE·HARDSIDE HAS IT ALL IN ONE!

How many times have you wished that there was a single source for your personal computer needs? Well look no further, TSE-HARDSIDE, located in pleasant New Hampshire, has virtually every conceivable item for your micro. Whether you're shopping for your Apple, Pet, TRS-80 or Atari, TSE-HARDSIDE has it all. We stock hardware, software, books, magazines and specialty items for all of the popular machines. So the next time you're out shopping for your system don't be surprised, be satisfied. Remember the name TSE-HARDSIDE as your choice for quality, service and reliability.

TSE·HARDSIDE
6 South St., Milford, NH 03055 (603) 673-5144
TOLL FREE OUT-OF-STATE 1-800-258-1790

Save $4.00

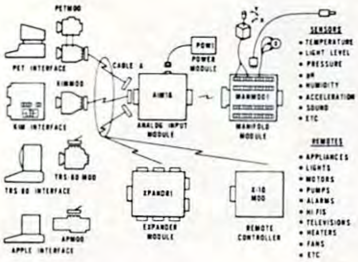
If you or a friend would like to subscribe to our TSE/HARDSIDE Complete Computer Catalog, simply send $2.00 and receive a $4.00 certificate good toward your first purchase of software, hardware, books or specialty items from TSE/HARDSIDE. I have a (check one) ☐ TRS-80, ☐ APPLE ☐ ATARI ☐ PET. A photo-copy of this coupon will be accepted.

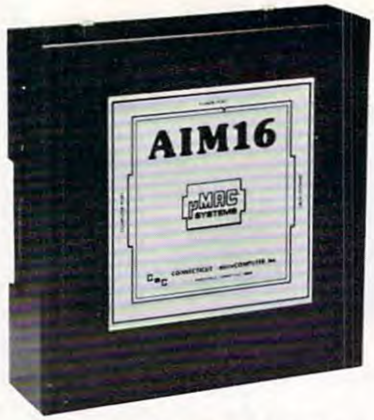
Name ____________________
Address ____________________ State ____________
City ____________________ Zip ____________

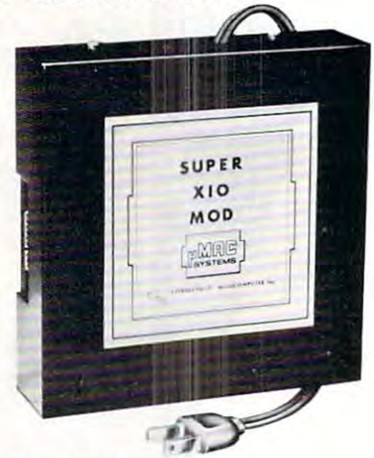
MAIL TO: TSE/HARDSIDE
6 South Street
Milford, NH 03055

CAPR81

Microcomputer Measurement And Control For PET,APPLE,KIM and AIM65







The world we live in is full of variables we want to measure. These include weight, temperature, pressure, humidity, speed and fluid level. These variables are continuous and their values may be represented by a voltage. This voltage is the analog of the physical variable. A device which converts a physical, mechanical or chemical quantity to a voltage is called a sensor.

Computers do not understand voltages: They understand bits. Bits are digital signals. A device which converts voltages to bits is an analog-to-digital converter. Our AIM 16 (Analog Input Module) is a 16 input analog-to-digital converter.

The goal of Connecticut microComputer in designing the uMAC SYSTEMS is to produce easy to use, low cost data acquisition and control modules for small computers. These acquisition and control modules will include digital input sensing (e.g. switches), analog input sensing (e.g. temperature, humidity), digital output control (e.g. lamps, motors, alarms), and analog output control (e.g. X-Y plotters, or oscilloscopes).

Connectors

The AIM 16 requires connections to its input port (analog inputs) and its output port (computer interface). The ICON (Input CONnector) is a 20 pin, solder eyelet, edge connector for connecting inputs to each of the AIM16's 16 channels. The OCON (Output CONnector) is a 20 pin, solder eyelet edge connector for connecting the computer's input and output ports to the AIM16.

The MANMOD1 (MANifold MODule) replaces the ICON. It has screw terminals and barrier strips for all 16 inputs for connecting pots, joysticks, voltage sources, etc.

CABLE A24 (24 inch interconnect cable) has an interface connector on one end and an OCON equivalent on the other. This cable provides connections between the uMACSYSTEMS computer interfaces and the AIM 16 or XPANDR1 and between the XPANDR1 and up to eight AIM 16s.

Analog Input Module .

The AIM 16 is a 16 channel analog to digital converter designed to work with most microcomputers. The AIM 16 is connected to the host computer through the computer's 8 bit input port and 8 bit output port, or through one of the uMAC SYSTEMS special interfaces.

The input voltage range is 0 to 5.12 volts. The input voltage is converted to a count between 0 and 255 (00 and FF hex). Resolution is 20 millivolts per count. Accuracy is 0.5% ± 1 bit. Conversion time is less than 100 microseconds per channel. All 16 channels can be scanned in less than 1.5 milliseconds.

Power requirements are 12 volts DC at 60 ma.

POW1

The POW1 is the power module for the AIM16. One POW1 supplies enough power for one AIM16, one MANMOD1, sixteen sensors, one XPANDR1 and one computer interface. The POW1 comes in an American version (POW1a) for 110 VAC and in a European version (POW1e) for 230 VAC.

TEMPSENS

This module provides two temperature probes for use by the AIM16. This module should be used with the MANMOD1 for ease of hookup. The MANMOD1 will support up to 16 probes (eight TEMPSENS modules). Resolution for each probe is 1°F.

Remote Controller-Clock and Calendar

AN INEXPENSIVE CONTROL SOLUTION FOR

HOME SECURITY • ENERGY CONSERVATION • GREENHOUSES
• ENVIRONMENTAL CONTROL
• INDUSTRIAL CONTROL
• LABORATORIES

SUPER X-10 MOD SPECS

1. Remote controller
   Controls up to 256 different remote devices by sending signals over the house wiring to remote modules. Uses BSR remote modules available all over the USA (Sears, Radio Shack, etc.). Does not require BSR control module. Does not use sonic link.
2. Clock/calendar
   Time of day - hours, minutes, seconds
   Date - month, day - automatically corrects for 28,29,30 and 31 day months. Day of the week.
3. Digital input/outputs
   8 inputs - TTL levels or switch closures.
   Can be used as a trigger for a stored sequence.
   8 outputs - TTL levels
   Power supply included 110VAC only.

XPANDR1

The XPANDR1 allows up to eight Input/Output modules to be connected to a computer at one time. The XPANDR1 is connected to the computer in place of the AIM16 or X10 MOD. Up to eight AIM16s or seven Aim 16s and one X10 MOD are then connected to each of the eight ports provided using a CABLE A24 for each module.

For your convenience the AIM16 and the X10 MOD come as part of a number of sets. The minimum configuration for a usable system is the AIM16 Starter Set 1 which includes one AIM16, one POW1, one ICON and one OCON. The AIM16 Starter Set 2 includes a MANMOD1 in place of the ICON. The minimum configuration for a usable system is the X10 MOD Starter Set which includes one X10 MOD, one ICON and one OCON. These sets require that you have a hardware knowledge of your computer and of computer interfacing.

For simple plug compatible systems we also offer computer interfaces and sets for many computers.

AIM16 .... **179.00**
SUPER X10 MOD (110 VAC only) **249.00**
POW1a (POWer module-110 VAC) **14.95**
POW1e (POWer module-230 VAC) **24.95**
ICON (Input CONnector) **9.95**
OCON (Output CONnector) **9.95**
MANMOD1 (MANifold MODule) **59.95**
CABLE A24 (24 inch interconnect cable) **19.95**
XPANDR1 (allows up to 8 Input or Output modules to be connected to a computer at one time) **59.95**
TEMPSENS2P1 (two temperature probes, -10°F to 160°F) **69.95**
LIGHTSENS1P1 (light level probe) **89.95**

The following sets include one AIM16, one POW1, one OCON and one ICON.
AIM16 Starter Set 1a (110 VAC) **189.00**
AIM16 Starter Set 1e (230 VAC) **199.00**

All prices and specifications subject to change without notice. Our 30-day money back guarantee applies.

The following sets include one AIM16, one POW1, one OCON and one MANMOD1.
AIM16 Starter Set 2a (110 VAC) **239.00**
AIM16 Starter Set 2e (230 VAC) **249.00**

The following modules plug into their respective computers and, when used with a CABLE A24, eliminate the need for custom wiring of the computer interface.
PETMOD (Commodore PET) **49.95**
KIMMOD (KIM,SYM) **39.95**
APMOD (APPLE II) **59.95**
TRS-80 MOD (Radio Shack TRS-80) **59.95**
AIM65 MOD (AIM 65) **39.95**

The following sets include one AIM16, one POW1, one MANMOD1, one CABLE A24 and one computer interface module
PETSET1a (Commodore PET - 110 VAC) **295.00**
PETSET1e (Commodore PET - 230 VAC) **305.00**
KIMSET1a (KIM,SYM,AIM65 - 110 VAC) **285.00**
KIMSET1e (KIM,SYM,AIM65 - 230 VAC) **295.00**
APSET1a(APPLE II - 110 VAC) **295.00**
APSET1e(APPLE II - 230 VAC) **305.00**
TRS-80 SET1a (Radio Shack TRS-80 - 110 VAC) **295.00**
TRS-80 SET1e(Radio Shack TRS-80 - 230 VAC) **305.00**
AIM65 SET1a(AIM65-110 VAC) **285.00**
AIM65 SET1e(AIM65-230 VAC) **295.00**

The following sets include one X10 MOD, one CABLE A24, one ICON and one computer interface module.
PETSET2(Commodore PET) **295.00**
KIMSET2(KIM,SYM) **285.00**
APSET2(APPLE II) **295.00**
TRS-80 SET2 (Radio Shack TRS-80) **295.00**
AIM65 SET2 (AIM65) **285.00**
SUPER X10 MOD/XPANDR1 SET2 (if you already hav[illegible]) .... **[illegible]**

Printer And Communication Interfaces For The CBM/PET

SADI For Serial Two-way Communication, Parallel Printers, and Serial Printers

SADI - The microprocessor based serial and parallel interface for the Commodore PET. SADI allows you to connect your PET to parallel and serial printers, CRT's, modems, acoustic couplers, hard copy terminals and other computers. The serial and parallel ports are independent allowing the PET to communicate with both peripheral devices simultaneously or one at a time. In addition, the RS-232 device can communicate with the parallel device.

Special Features for the PET interface include:
- Conversion to true ASCII both in and out
- Cursor controls and function characters specially printed
- Selectable reversal of upper and lower case
- Addressable - works with other devices

Special Features for the serial interface include:
- Baud rate selectable from 75 to 9600
- Half or full duplex
- 32 character buffer
- X-ON, X-OFF automatically sent
- Selectable carriage return delay

Special Features for the parallel interface include:
- Data strobe - either polarity
- Device ready - either polarity
- Centronics compatible

Complete with power supply, PET IEEE cable, RS-232 connector, parallel port connector and case. Assembled and tested.
SADIa (110VAC) $295
SADIe (230VAC) $325

ADA1600 • For Parallel NEC and Centronics Standard Printers

The ADA1600 is a low cost easy to use interface for the Commodore Computers. It allows the PET and CBM computers to use standard Centronics type printers (including the NEC 5530) for improved quality printing. The ADA1600 has a two foot cable which plugs into the PET IEEE port. Another IEEE card edge connector is provided for connecting disks and other peripherals to the PET. The ADA1600 is addressable and does not tie up the bus. The address is switch selectable. A four foot cable with a standard 36 pin Centronics connector is provided. A switch selects upper/lower case, upper/lower case reversed (needed for some Commodore machines) and upper case only for clearer program listings. Works with WORDPRO, BASIC and other software. No special programming is required. The case measures 3 1/2 x 5 3/4 inches. Comes complete, assembled and tested, with case and cables. Power is obtained from the printer or an external power supply may be used. Retail price for the ADA1600 is $129.

ADA1450 • Serial Printer Adapters

The ADA1450 is a low cost, easy to use serial interface for the Commodore Computers. It allows the PET and CBM computers to use standard serial printers for improved quality printing. The ADA1450 has a two foot cable which plugs into the PET IEEE port. Another IEEE card edge connector is provided for connecting disks and other peripherals to the PET. The ADA1450 is addressable and does not tie up the bus. The address is switch selectable. A six foot RS-232 cable is provided with a DB25 connector. Pin 3 is data out. Pins 5,6 and 8 act as ready lines to the printer. Pins 4 and 20 act as ready lines from the printer. These lines can be switched for non-standard printers. Baud rate is selectable to 9600 baud. A switch selects upper/lower case, upper/lower case reversed (needed for some Commodore machines) and upper case only for clearer program listings. Works with WORDPRO, BASIC and other software. No special programming is required. The case measures 3 1/2 x 5 3/4 inches. Comes complete, assembled and tested, with case, cables, power supply and software on cassette for graphing functions, formatting data etc. The ADA1450 has a female DB25 connector at the end of the RS-232 cable for most standard printers. The ADA1450N has a male DB25 at the end of the RS-232 cable for the DIABLO serial printers. Retail price for the ADA1450 or 1450N is $149.

ADA730 Parallel • For the Centronics 730 and 737 Printers

The ADA730 is a low cost easy to use interface for the Commodore Computers. It allows the PET and CBM computers to use Centronics type 730 and 737 printers. The ADA730 has a two foot cable which plugs into the PET IEEE port. Another IEEE card edge connector is provided for connecting disks and other peripherals to the PET. The ADA730 is addressable and does not tie up the bus. The address is switch selectable. A cable with a 36 pin card edge connector is provided. A switch selects upper/lower coase, upper/lower case reversed (needed for some Commodore machines) and upper case only for clearer program listings. Works with WORDPRO, BASIC and other software. No special programming is required. The case measures 3 1/2 x 5 3/4 inches. Comes complete, assembled and tested, with case and cables. Power is obtained from the printer or an external power supply may be used. Retail price for the ADA is $129.

Word Processor Program •

PET Word Processor. On tape - $39.50, On disk - 49.50
For 8K Pets 29.50
For 16K and 32K Pets 39.50
Compose and print letters, flyers, ads, manuscripts, etc. Uses disk or tape. 30 page manual included.

ADA400 RS-232 To Current Loop Adapter

RS-232 to current loop adapter (ADA 400). $29.50
Two circuits - 1 each direction. Run an RS-232 device off a computer's teletype port or vice versa. Optoisolated.

ADA4005 - Solder Pads 24.50
ADA400B - Barrier Strips with screw terminals 29.50

QUANTITY	DESCRIPTION	PRICE	TOTAL

SUBTOTAL
Handling and shipping — add per order $4.00
Foreign orders add 10% for AIR postage
Conn. residents add 7% sales tax
TOTAL ENCLOSED

NAME ____
COMPANY ____
ADDRESS ____
CITY ____
STATE ____ ZIP ____
VISA O M/C O Expiration date ____
Card number ____

Mention this magazine and deduct 2% from the TOTAL

In the US order from: Connecticut MicroComputer, Inc.
34 Del Mar Drive Brookfield, CT 06804 (203) 775-4595
In Canada order from: Batteries Included, LTD
71 McCaul St. F6 Toronto, Canada M5T2X1 (416)596-1405

CmC

Connecticut microComputer, Inc.
34 Del Mar Drive, Brookfield, CT 06804
203 775-4595 TWX: 710 456-0052

Using The Aim 65 As A Remote Terminal For An Apple

Tony Davis and Marvin L. De Jong
Department of Mathematics-Physics
The School of the Ozarks
Pt. Lookout, MO 65726

In the March issue of **COMPUTE!** (page 28 – Computer Communications Experiments) a circuit using the 6551 ACIA (Asynchronous Communications Interface Adapter) and a RS-232C interface to a modem were described. We have used this same interface to a NOVATION CAT modem on the AIM 65 to operate an Apple II over a telephone link. The Apple was equipped with a Hayes micromodem. The Apple was used to run BASIC programs, but its monitor can also be used to load machine language programs or data.

The circuit will not be repeated here, but we will provide the listing of the simple program that we used on the AIM 65. The Hayes Micromodem comes with its own firmware.

We operated the 6551 in the mode where a received character produces an interrupt. The interrupt routine simply prints the character on the display by jumping to an AIM 65 monitor subroutine. The program runs at 300 or 110 Baud. In Listing 1 we show the 6551 initialized to run at 300 Baud. Note that in either case the AIM 65 thermal printer was not used because its print time is so long that several characters are missed. To use it one would have to write a routine to buffer the incoming data. Our ultimate goal is to use the AIM 65 to access the college's big IBM mainframe. I am especially interested in being able to calculate my own salary and print my own paycheck at the end of each month.

Listing 1. Program to operate an Apple from an AIM 65 over a telephone line.

```
$0F00 58              START  CLI            Allow interrupts.
 0F01 D8                     CLD
 0F02 A9 09                  LDA  #09       Set up the 6551
                                            command register.
 0F04 8D 02 94               STA  CMNDREG
 0F07 A9 16                  LDA  #$13      Set up the control
                                            register for
 0F09 8D 03 94               STA  CNTREG    300 Baud.
 0F0C 20 3C E9        CHAR   JMP  READ      Get character from
                                            AIM keyboard.
 0F0F 8D 00 94               STA  DATA      Output data to the 6551.
 0F12 AD 01 94        CHECK  LDA  STATUS    Check the status register
 0F15 29 10                  AND  #$10      Check bit four.
 0F17 FO F9                  BEQ  CHECK     Wait for data to be
                                            transmitted.
 0F19 D0                     BNE  CHAR      Then get another
                                            character.
                             **********
Interrupt Routine
$0E00 48              IRQ    PHA            Save the accumulator.
 0E01 AD 00 94               LDA  DATA      Get character that
                                            was sent.
 0E04 20 7A E9               JMP  OUTPUT    Output character to
                                            display.
 0E07 AD 01 94               LDA  STATUS    Clear IRQ flag.
 0E0A 68                     PLA
 0E0B 40                     RTI
```

Be sure to load the interrupt vector $0E00.

ONE GREAT SHOW DESERVES ANOTHER

Since the COMDEX show was first established two years ago, and particularly after the recent overwhelming success of COMDEX '80, many vendors and ISOs alike have been telling us that this unique conference and exposition once a year just isn't enough.

Hello there, COMDEX/SPRING!

The big push is on to computerize America, boost productivity in every office and shop across the land. That's why Independent Sales Organizations (ISOs) are springing up like dandelions after the rain. Vendors are accelerating their schedule of new product introductions, and their intense search for independent sales representation.

Thus, COMDEX/SPRING, in New York City, June 23-25, 1981. If you're a computer industry marketer, whether vendor or ISO, you've now got TWO opportunities each year to get together with the people who really count. You've got COMDEX/SPRING in New York in June, and you've got COMDEX '81 in Las Vegas in November. That's a one-two marketing punch guaranteed to get the job done!

Who exhibits at the COMDEX shows?

Leading vendors of small computer and word processing systems, peripherals, software and media-supplies-services have been flocking to COMDEX since its inception in 1979. Write or call us immediately for detailed information on COMDEX/SPRING and, among other things, we'll send you a list of more than 360 such companies which exhibited at COMDEX '80.

ISO, as defined in our dictionary.

ISO (ai-ess-oh) Independent Sales Organization. Acronym coined by The Interface Group in summer 1979 as convenient umbrella for all independent third-party sellers of small systems and related products and services. Such as: *Dealers, distributors, systems houses, commercial OEMs, computer retailers, manufacturers' reps, turnkey vendors, office machines/products dealers, software houses, etc.* The acronym has gained widespread acceptance, following its introduction by COMDEX.

COMDEX/SPRING'81™

June 23-25, 1981 • New York Coliseum, New York, New York

For further information, write to COMDEX/SPRING '81,
160 Speen Street, Framingham, MA 01701.
Or call us toll-free: 800-225-4620 (in Mass., 617-879-4502).

Another Conference and Exposition from THE INTERFACE GROUP
Producers of: INTERFACE, INTERFACE WEST, FEDERAL DP EXPO, COMDEX, COMDEX/SPRING, THE NATIONAL SMALL COMPUTER SHOW

NEW— LOW COST DISK FOR PET CBM

PEDISK II from cgrs MICROTECH is a new floppy disk controller board for the Commodore PET-/CBM (tm) computer. PEDISK II contains control ROM and all logic required to read, write, and control up to two standard floppy disk drives (either 5.25" or 8"). The PEDISK II controller mounts inside the PET and occupies previously unused memory space. The circuit board plugs into an existing ROM socket in your machine, and is connected by cable to an external disk drive.

A complete disk system includes the PEDISK II Controller, Disk Operating System software, connecting cable, and an external disk drive set. Simply plug in the controller board, and your disk system is ready to run.

The PEDISK II system offers speed, reliability, and IBM compatibility.

Complete System prices with DOS and cable:

5.25" 40 track, 1 drive, 143K	$560
5.25" 80 track, 1 drive, 286K	690
8" IBM 3740 format, 77 track, 250K	995

Individual Component Prices:

PEDISK II Controller Card	$229
DOS for PEDISK II	75
Cable assembly for 2 drives	35

CBM SOFTWARE

Legal Time Accounting Package	
Medical Accounting Package	
Complete General Accounting Package	
Comprehensive Investment Analysis Package	
Dow Jones Portfolio Management	$135
Personal Tax Calculator	65
Tax Preparation System	445
Information Retrieval and Management Aid	400
Wordcraft 80 Wordprocessor Package	325
Pascal Development Package	295
Assembler Development Package	99
Intelligent Terminal Emulator	30

SOFTPACK-1 from Competitive Software — 25
16 games and utilities for PET

MICROREVERSI for PET by Michael Riley — 10
Super Machine Language Version of Othello

TUNNEL VISION/KAT & MOUSE by Michael Riley — 10
Two excellent machine language maze programs.

PERSONAL SOFTWARE

Microchess	17.00
Checker King	17.00
Gammon Gambler	17.00
Time Trek	13.45
Bridge Partner	13.45

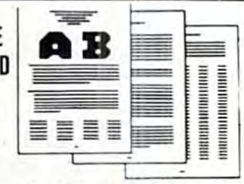
HAYDEN SOFTWARE

Complex Mathematics	12.70
Engineering Mathematics	12.70
General Mathematics	12.70
MCAP: Circuit Analysis Program	21.00
Energy Miser	24.50

JUKEBOX SERIES FOR PET by L.C. Cargile
Excellent 4 part harmony music—write for list

Automated Simulations (EPYX) Fantasy Games

PAPER-MATE 60 COMMAND WORD PROCESSOR



Paper-Mate is a full-featured word processor for $29.00 by Michael Riley. Paper-Mate incorporates 60 commands to give you full screen editing with graphics for all 16k or 32K PETs, all printers, and disk or tape drives. It also includes most features of the CBM WordPro III, plus many additional features.

For writing text, Paper-Mate has a definable keyboard so you can use either Business or Graphics machines. Shift lock on letters only, or use keyboard shift lock. All keys repeat.

Paper-Mate text editing includes floating cursor, scroll up or down, page forward or back, and repeating insert and delete keys. Text Block handling includes transfer, delete, append, save, load, and insert.

All formatting commands are imbedded in text for complete control. Commands include margin control and release, column adjust, 9 tab settings, variable line spacing, justify text, center text, and auto print form letter (variable block). Files can be linked so that one command prints an entire manuscript. Auto page, page headers, page numbers, pause at end of page, and hyphenation pauses are included.

Unlike most word processors, PET graphics as well as text can be used. Paper-Mate can send any ASC11 code over any secondary address to any printer.

Paper-Mate works on 16K or 32K PETs with any ROM, cassette or disk, and CBM or non-CBM printers. An 8K version is in the planning.

To order Paper-Mate, specify machine and ROM type.

On Tape (with manual):	$29.95
On Disk (with manual):	$32.95
Manual Separate:	$ 1.00

KMMM Pascal for PET/CBM $75

A subset of standard Pascal with extensions • Machine Language Pascal Source Editor • Machine Language P-Code Compiler • P-Code Interpreter (for debugging and learning) • P-Code to machine language translator for optimized object code • Run-time package • Floating point capability • User manual and sample programs • Includes source code editor
Specify ROM version (16K minimum), disk or tape.

EARL for PET (disk file based) $65

Editor, Assembler, Relocater, Linker. Generates relocatable object code using MOS Technology mnemonics.
Disk file input (can edit files larger than memory). Links multiple object programs as one memory load. Listing output to screen or printer. Enhanced editor operates in both command mode and cursor oriented "window" mode.

SUPERSORT by James Strasma $35

Supersort is an excellent general purpose machine language sort routine for PET/CBM computers. Sorts both one and two dimensioned arrays at lightning speed in either ascending or descending order. Other fields can be subsorted when a match is found, and fields need not be in any special order. Sort arrays may be specified by name, and fields are random length. Allows sorting by bit to provide 8 categories per byte. The routine works with all PET BASICs, adjusts to any memory size, and can co-exist with other programs in high memory.

FLEX-FILE For CBM/PET Computers

Self Calculating
DATA BASE
REPORT WRITER
MAILING LIST

Flex File is a set of flexible, friendly programs to allow you to set up and maintain a data base as well as print files with a versatile Report Writer or a Mail Label routine. Programmers will find it easy to add subroutines to their own programs to make use of Data Base files.

RANDOM ACCESS DATA BASE

Record size limit is 250 characters. The number of records per disk is limited only by the size of each record and the amount of free space on the disk. File maintenance lets you step forward or backward through a file, add, delete or change a record, go to a numbered record, or find a record from a specified field. The Find command locates any record when you enter all (or a portion of) the desired key field. Field lengths can vary from record to record provided the sum of the fields does not exceed the size of the record. This allows maximum packing of information. The file can be sorted by any field. Any field can be specified as a key field at any time. Sequential files from other programs can be converted to random files, and random can be converted to sequential. Maximum record size, fields per record, and order of fields can be changed at any time.

MAILING LABELS

When record size is 127 characters (typical for mailing list), each disk can handle over 1000 records (about 2800 with the 8050 drive). Labels can be printed any number of labels across, and in any column position. Any number of fields can be printed on a label in any order, and two or three fields can be joined together on one line (like first name, last name, and title). A "type of customer" field allows selective printing.

REPORT WRITER

The contents of any field can be placed in any column. Numerics can be decimal point justified and rounded to any accuracy. Any column can be defined as a series of mathematical functions performed on other columns. These functions may include +, −, x, /, %, and various log and trig functions. Results of operations such as running total may be passed from row to row. At the end of the report a total and/or average can be calculated for any column. Complete record selection, including field within range, pattern match, and logical functions can be specified individually or in combination with other parameters.

Flex File was developed by Michael Riley.

Flex File System — **$60**

Specify machine size (32K recommended) and ROM type for both disk and computer.

Good things coming!

VIC 20 Computer with Color Graphics

CBM 8096

CBM Single Disk Drive

SM-KIT–a "super toolkit" for PET/CBM 40/80 column BASIC 4.0 machines. Includes all the standard functions in much enhanced fashion, plus disk commands (including Disk Merge and DOS Support), Screenprint, etc.

SAVE ENERGY with Commodore Programmable Thermostat

Installs easily in place of your existing unit to save up to 30% on your heating/cooling bills.

Introductory SPECIAL — **$114**

115 E. Stump Road
Montgomeryville, PA 18936 215-822-7727 **A B Computers** WRITE FOR CATALOG
Add $1.25 per order for shipping. We pay balance of UPS surface charges on all prepaid orders.

THE STAR MODEM

From Livermore Data Systems

RS232 MODEM	SALE $135
IEEE 488 MODEM	**SALE $199**
RS232 CCITT	$170
IEEE 488 CCITT	$280

STAR Modem is the price performance leader with a full 2 YEAR FACTORY WARRANTY.

WE CARRY THE BLACK APPLE

BELL & HOWELL
MADE EXCLUSIVELY FOR BELL & HOWELL BY
apple computer inc.

EBS Business System for PET/CBM $795
Extremely comprehensive package for small business. Fully integrated inventory and accounts receivable system including invoices, packing slips, mail labels, statements, bank deposit slips, and 17 reports.

Tiny Mail List for PET BASIC 3.0 $12
Clever disk based package by L.C.Cargile. Uses overlays to avoid garbage collection delays.

DISK MASTER by Robert Baker $12
Comprehensive diskette cataloging package to keep track of all programs and files on up to 140 diskettes. Specify 2040/4040/8050.

PROGRAM YOUR OWN EPROMS
Branding Iron for PET/CBM $79
EPROM Programmer with software for all ROM versions. Includes all necessary hardware and software to program or copy 2716 and 2532 EPROMS.

FORTH for PET/CBM or Apple $65
A full-featured FORTH with extensions conforming to Forth Interest Group standards. Includes assembler, string processing capabilities, disk virtual memory, multiple dimensioned arrays, floating point and integer processing.

MIPLOT Intelligent Plotter
by Watanabe Instruments (Digiplot)

SPECIAL $1145

Has all intelligent functions for producing graphs and drawings including 8 vector and 4 character commands. Solid and broken lines can be specified. Character generator for alpha, numeric, and symbols. Characters can be rotated in 4 orientations, and can be 16 sizes. Coordinate axes drawn by specifying graduation interval and number of repetitions. Parallel ASCII interface, 11 x 17 paper.

SPECIALS

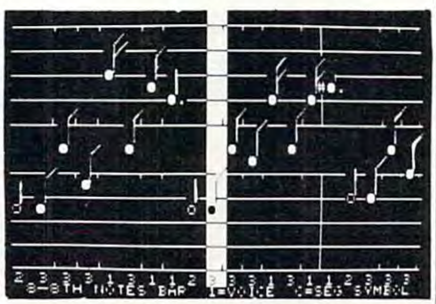
EPSON MX-80 Printer	
EPSON MX-70 Printer	
STARWRITER Daisy Wheel Printer	$1390
Centronics 737 Printer	725
NEC Spinwriter	2500
Diablo 630 Daisy Wheel	2150
Leedex Video 100 12" Monitor	135
ZENITH DATA SYSTEMS	
Z19 Video Terminal (factory assem.)	729
Z89 with 48K (factory assem.)	2150
SYM-1	209
SYM BAS-1 BASIC or RAE-1/2 Assembler	85
KTM-2/80 Synertek Video Board	349
Seawell Motherboard - 4K RAM	195
Seawell 16K Static RAM - KIM, SYM, AIM	320

6502	7.45	10/6.95	50/6.55	100/6.15
6502A	8.40	10/7.95	50/7.35	100/6.90
6520 PIA	5.15	10/4.90	50/4.45	100/4.15
6522 VIA	6.45	10/6.10	50/5.75	100/5.45
6532	7.90	10/7.40	50/7.00	100/6.60
2114-L200 ns RAM		3.75	25/3.50	100/3.25
2114-L300 ns RAM		3.15	25/2.90	100/2.65
2716 EPROM (5 volt)		9.90	5/9.45	10/8.90
TMS 2532 EPROM				23.50
4116-200 ns RAM				**8 for 29.00**
S-100 Wire Wrap				2.65

CASSETTES – AGFA PE-611 PREMIUM
High output, low noise, 5 screw housing, labels.

C-10	10/5.65	50/25.00	100/48.00
C-30	10/7.30	50/34.00	100/66.00

All other lengths available. Write for price list.



4 PART HARMONY MUSIC SYSTEM for PET
The Visible Music Monitor, by Frank Levinson, allows you to easily enter, display, edit, and play 4 part harmony music. Includes whole notes thru 64ths (with dotted and triplets), tempo change, key signature, transpose, etc. The KL-4M unit includes D to A converter and amplifier (add your own speaker).
KL-4M Music Board with VMM Program $59.90

commodore

CBM-PET SPECIALS

FREE **Up to $315 free merchandise with purchase of one of following CBM-PET items:**

		FREE
8032 32K - 80 Column CRT	$1795	315
8050 Dual Disk Drive - 950,000 bytes	1795	315
4016 Full Size Graphics Keyboard	995	170
4032 Full Size Graphics Keyboard	1295	230
32B Business Keyboard -	SPECIAL	800
2040/4040 Dual Disk Drive - 340K	1295	230
4022 Tractor Feed Printer	795	140
CBM Voice Synthesizer	395	50
C2N External Cassette Deck	95	12
Used CBM/PET Computers		CALL
VIC Personal Computer	300	25

WRITE FOR SYSTEM PRICES

***** EDUCATIONAL DISCOUNTS *****
Buy 2 PET/CBM Computers, receive 1 FREE

WordPro 3+ – 32K CBM, disk, printer	250
WordPro 4+ – 8032, disk, printer	385
OZZ Data Base System for CBM 8032	335
VISICALC for PET or ATARI	170
BPI General Ledger, A/P, A/R for PET/CBM	270
Programmers Toolkit – PET ROM Utilities	34.90
PET Spacemaker II ROM Switch	36.00
2 Meter PET to IEEE or IEEE to IEEE Cable	40.00
Dust Cover for PET	6.90
IEEE-Parallel Printer Interface for PET	110.00
IEEE-RS232 Printer Interface for PET	120.00
The PET Revealed	17.00
Library of PET Subroutines	17.00
Source Hookup over 1000 programs/services	**88**

DISK SPECIALS

SCOTCH (3M) 5¼"	10/2.85	50/2.75	100/2.65
SCOTCH (3M) 8"	10/2.90	50/2.80	100/2.70
Verbatim 5¼"	10/2.45	50/2.40	100/2.35
(add 1.00 for 5¼" Verbatim plastic storage box)			
Verbatim 8" Dbl. Dens.	10/3.45	50/3.35	100/3.25
BASF 5¼"	10/2.60	20/2.50	100/2.40
BASF 8"	10/2.65	20/2.55	100/2.45

WE STOCK MAXELL DISKS

Diskette Storage Pages		10 for 3.95
Disk Library Cases	8" - 2.85	5" - 2.15

ATARI 800 $745
All Atari Modules 20% OFF

ATARI EDUCATIONAL PLAN Write for details.

A P Products 15% OFF
A P Hobby-Blox 15% OFF

ALL BOOK and SOFTWARE PRICES DISCOUNTED

The 8086 Book (Osborne)	14.00
Z8000 Assembly Language Programming	16.90
PET Personal Computer Guide (Osborne)	12.75
PET and the IEEE-488 Bus (Osborne)	13.60
6502 Assembly Language (Osborne)	14.45
Programming the 6502 (Zaks)	10.45
6502 Applications Book (Zaks)	10.45
6502 Software Cookbook (Scelbi)	9.45
CP/M Handbook (w/ MP/M) Zaks	11.85
Practical BASIC Programs (Osborne)	13.60
Some Common BASIC Programs (Osborne)	12.75

115 E. Stump Road
Montgomeryville, PA 18936 215-822-7727 **A B Computers**

WRITE FOR CATALOG.
Add $1.25 per order for shipping. We pay balance of UPS surface charges on all prepaid orders. Prices listed are on cash discount basis. Regular prices slightly higher.

EPIDEMIC

A Simulation Of An Epidemic In A Closed Community

Andy Gamble
Computer Science Instructor
Columbia College
Vancouver, BC V6J 2A2

It seems that the programs most readily accepted by students, with good reason, are the ones that involve an element of competition. How many times have you seen programs that, while advertised as simulations, are no more than textbooks written for the screen? The amount of student involvement is often limited to a 'Press any key to continue' or to a few multiple-choice questions.

EPIDEMIC operates in the realm of instruction and hypothesis-testing and, although the subject matter would seem to belie it, a certain amount of competitive fun. This competition, by the way, is on an individual basis, a kind of see-if-I-can-beat-the-machine, similar to the way HAMURABI works.

The program, through the use of PET graphics, illustrates how a disease could spread in a closed community. Given such a community, as for example an island with no physical connection to the world outside (are there any?), what are the parameters affecting the epidemic? The islanders move about randomly, infecting others if able to do so. The disease itself lasts for a specified amount of time, otherwise it is certain that all islanders will contract it. After this time, an infected person will become uncontagious, and also immune from further infection.

The RUN of the program prompts the user for such input as the number of inhabitants on the island, the number originally infected and the time for which the disease is contagious (lines 180-250).

Each person on the island is inspected to see if:

1) he is starting his period of infection. At this point a random move (or no move at all) is made (lines 500-520, 900-980 and 1000-1040). Note that no one is allowed to move off the island.

2) he is able to infect his immediate neighbors (lines 540 and 1060-1140)

3) he is infected by his neighbors (lines 560 and 1160-1240). This will only happen if he has not yet been infected.

4) his period of infection has finished (lines 580-620). He now passes into the immune category.

The program continues until there are no more infected people on the island (line 660). A bar chart is then presented which summarizes the history of the disease (lines 680-810).

The shape of the island is obtained from the DATA statements 1670-1710, and can easily be changed to suit your locale if you wish.

A further change could be a random element acting so that it is not absolutely certain that an islander will become infected if in contact with a diseased neighbor. Lines 1160-1230 would become

1160 IFPEEK(PP(I)–41)=COTHENIFRND-(1)».5THENPOKEPP(I),CO etc.

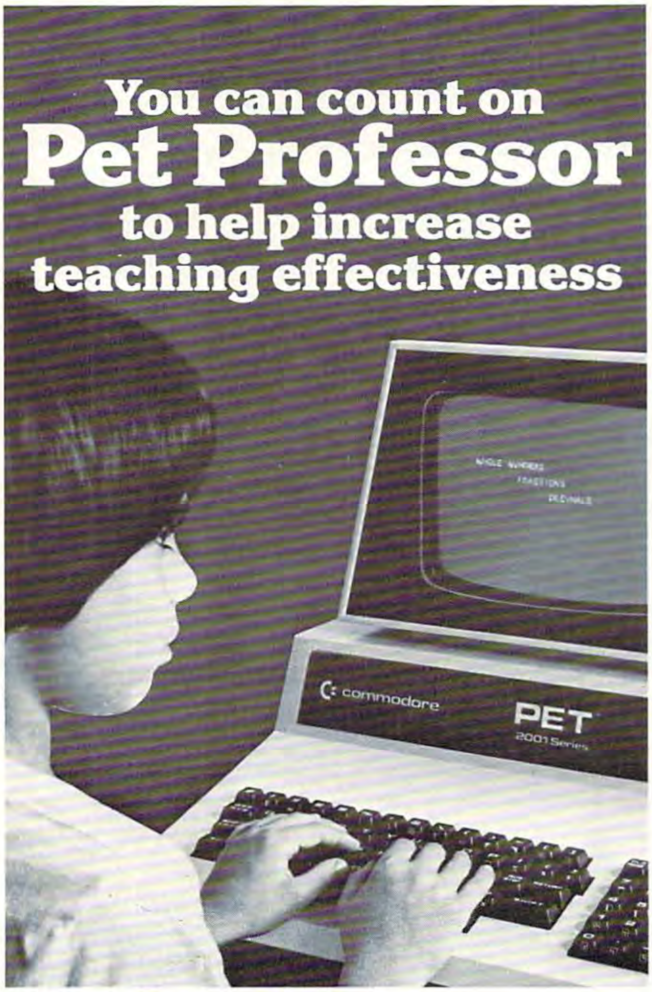
for a 50% chance of being infected.

Here is the challenge: given a constant number of inhabitants, what is the smallest number of infected people which insures that all will become infected? This makes a nice problem in statistics, if you want to go that far, but it is enjoyable to obtain experimental evidence from this program. That, after all, is what simulations are for.

The program will run on new and old roms, and uses less than 6K as given.

```
100 REM EPIDEMIC : ANDY GAMBLE, AUG 80
110 REM COLUMBIA COLLEGE,1619 W10 AVE    ¬
      ¬ VANCOUVER BC V6J 2A2
120 I=RND(-RND(0)):POKE59468,12
130 POKE59458,62:REM SPEED POKE
140 T$="h↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓"
150 QP=515:QA=126:IFPEEK(50000)THENQP=15
      ¬1:QA=44
160 IG=160:VI=215:CO=209:GI=170
170 GOTO1360
180 PRINTCHR$(147)"HOW MANY ISLANDERS ¬
      ¬(<=100)?";:GOSUB1720:NP=VAL(Z1$)
190 IFNP=0THEN180
200 IFNP>100THENPRINT"↑↑";:GOTO180
210 PRINT"↓HOW MANY INFECTED AT ¬
      ¬START?";:GOSUB1720:NI=VAL(Z1$)
220 IFNI=0THENPRINT"↑↑";:GOTO210
230 IFNI>NPTHENPRINT"↑↑";:GOTO210
240 PRINT"↓CONTAGION TIME (DAYS)?";:
      ¬GOSUB1720:CT=VAL(Z1$)
250 IFCT=0THENPRINT"↑↑";:GOTO240
260 RESTORE
270 PRINTCHR$(147):FORI=1TO18:READIL,IR
280 FORJ=ILTOIR:POKEJ,IG:NEXT:NEXT
290 PRINT"hrWr̂=UNINFECTED  rOr̂=CONTAGIOU
      ¬S  r*r̂=IMMUNE"
300 PRINT"h↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓CONTAGION"
310 PRINT"h↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓TIME "CT;
320 IFCT>1THENPRINT" DAYS":GOTO340
330 PRINT" DAY"
340 REM GARBAGE COLLECTION FOR ARRAYS
350 POKEQA+2,PEEK(QA):POKEQA+3,PEEK(QA+1
      ¬):Z9=FRE(0)
360 DIMPP(NP),PG(NP),DC(NP),NI(50),
      ¬IM(50)
370 FORI=1TONP:PG(I)=VI:NEXT
380 FORI=1TONI:PG(I)=CO:NEXT
390 FORI=1TONP:DC(I)=0:NEXT
400 FORI=1TONI:DC(I)=CT:NEXT
410 FORI=1TONP
```

```
420 PP(I)=INT(690*RND(1)+32901):
    ¬IFPEEK(PP(I))<>IGTHEN420
430 POKEPP(I),PG(I):NEXT
440 ND=0
450 IM=0:PRINT"h↓"TAB(7)NP-NI-IM;TAB(20)
    ¬NI;TAB(30)IM
460 PRINT"h↓↓↓DAY";ND
470 ND=ND+1:NI=0:IM=0
480 FORI=1TONP
490 REM NEW CONTAGIOUS FROM LAST TIME
500 IF(PEEK(PP(I))=CO)AND(PG(I)=VI)THEND
    ¬C(I)=CT
510 PG(I)=PEEK(PP(I))
520 ONINT(9*RND(1)+1)GOSUB900,910,920,
    ¬930,940,950,960,970,980
530 REM INFECTING?
540 IFPG(I)=COTHENGOSUB1060
550 REM INFECTED?
560 IFPG(I)=VITHENGOSUB1160
570 REM ONE DAY LESS
580 IFPG(I)=COTHENDC(I)=DC(I)-1
590 REM END CONTAGION
600 IFDC(I)<0THENPG(I)=GI:POKEPP(I),GI
610 IFPEEK(PP(I))=COTHENNI=NI+1
620 IFPEEK(PP(I))=GITHENIM=IM+1
630 NEXT
640 IFND<=50THENNI(ND)=NI:IM(ND)=IM
650 PRINT"h↓"TAB(7)NP-NI-IM"←  ";TAB(20)
    ¬NI"←  ";TAB(30)IM"←  "
660 IFNI>0THEN460
670 PRINTT$;:GOSUB1250
680 PRINTCHR$(147)"DAY"TAB(5)"INFECTION"
    ¬:PRINT
690 FORND=1TO50
700 FORWT=1TO150:NEXT
710 PRINTND;TAB(4);
720 PRINTNP-NI(ND)-IM(ND);NI(ND);IM(ND):
    ¬PRINT
730 IFNP-NI(ND)-IM(ND)=0THEN750
740 FORI=1TONP-NI(ND)-IM(ND):PRINT"rW";:
    ¬NEXT
750 IFNI(ND)=0THEN770
760 FORI=1TONI(ND):PRINT"rQ";:NEXT
770 IFIM(ND)=0THEN790
780 FORI=1TOIM(ND):PRINT"r*";:NEXT
790 PRINT
800 IFNI(ND)=0THEN820
810 PRINT:NEXTND
820 PRINT"↓LIKE TO SEE THE CHART AGAIN ¬
    ¬(Y/N)?";:GOSUB1720:Q$=Z1$
830 IFQ$=""THENPRINT"↑↑";:GOTO820
840 IFLEFT$(Q$,1)="Y"THEN680
850 PRINTCHR$(147)LEFT$(T$,10)"WANT ¬
    ¬ANOTHER TRY (Y/N)?";:GOSUB1720:
    ¬Q$=Z1$
860 IFQ$=""THEN850
870 IFLEFT$(Q$,1)="Y"THEN180
880 END
890 REM MOVE S/R'S
900 MV=39:GOSUB1000:RETURN
910 MV=40:GOSUB1000:RETURN
920 MV=41:GOSUB1000:RETURN
930 MV=-1:GOSUB1000:RETURN
940 MV=0:GOSUB1000:RETURN
950 MV=1:GOSUB1000:RETURN
960 MV=-41:GOSUB1000:RETURN
970 MV=-40:GOSUB1000:RETURN
980 MV=-39:GOSUB1000:RETURN
990 REM MAKE MOVE
1000 IFPEEK(PP(I)+MV)<>IGTHENRETURN
1010 PG(I)=PEEK(PP(I))
1020 POKEPP(I),IG
1030 PP(I)=PP(I)+MV
1040 POKEPP(I),PG(I):RETURN
1050 REM INFECTING OTHERS
1060 IFPEEK(PP(I)-41)=VITHENPOKEPP(I)-41
     ¬,CO
1070 IFPEEK(PP(I)-40)=VITHENPOKEPP(I)-40
     ¬,CO
1080 IFPEEK(PP(I)-39)=VITHENPOKEPP(I)-39
     ¬,CO
1090 IFPEEK(PP(I)- 1)=VITHENPOKEPP(I)- ¬
     ¬1,CO
1100 IFPEEK(PP(I)+ 1)=VITHENPOKEPP(I)+ ¬
     ¬1,CO
1110 IFPEEK(PP(I)+39)=VITHENPOKEPP(I)+39
     ¬,CO
1120 IFPEEK(PP(I)+40)=VITHENPOKEPP(I)+40
     ¬,CO
1130 IFPEEK(PP(I)+41)=VITHENPOKEPP(I)+41
     ¬,CO
1140 RETURN
1150 REM INFECTION FROM OTHERS
1160 IFPEEK(PP(I)-41)=COTHENPOKEPP(I),CO
1170 IFPEEK(PP(I)-40)=COTHENPOKEPP(I),CO
1180 IFPEEK(PP(I)-39)=COTHENPOKEPP(I),CO
1190 IFPEEK(PP(I)- 1)=COTHENPOKEPP(I),CO
1200 IFPEEK(PP(I)+ 1)=COTHENPOKEPP(I),CO
1210 IFPEEK(PP(I)+39)=COTHENPOKEPP(I),CO
1220 IFPEEK(PP(I)+40)=COTHENPOKEPP(I),CO
1230 IFPEEK(PP(I)+41)=COTHENPOKEPP(I),CO
1240 RETURN
1250 PRINT"     rPRESS ANY KEY TO ¬
     ¬CONTINUE"
1260 GETQ$:IFQ$=""THEN1260
1270 RETURN
1280 REM*******VARIABLES**************
1290 REM NP=# OF PEOPLE,NI=#  INFECTED  ¬
     ¬          IM=# OF IMMUNE
1300 REM CT=DAYS FOR CONTAGIOUS,
     ¬IG=                ISLANDGRAPHIC,
     ¬PP=POS OF PEOPLE
1310 REM ND=# OF DAYS,VI=NOTYETINFECTED
1320 REM CO=CONTAGIOUS,PG=PEOPLEGRAPHIC
1330 REM GI=GRAPHIC IMMUNE,MV=MOVE,
     ¬DC=             DAYS OF CONTAGION ¬
     ¬LEFT
1340 REM******************************
1350 REM TITLES
1360 PRINT"ĥ":FORI=32768TO32807:
     ¬POKEI,224:POKEI+960,224:NEXT
1370 FORI=32808TO33688STEP40:POKEI,224:
     ¬POKEI+39,224:NEXT
1380 PRINT"h↓↓↓↓"
1390 PRINTTAB(9)"  O# $'       L
1400 PRINTTAB(9)"  L$::::::MNLLLLNM
1410 PRINT"h↓↓↓↓↓↓↓↓↓↓"TAB(14)"rEPIDEMIC
     ¬"
1420 PRINT"h↓↓↓↓↓↓↓↓↓↓↓↓"TAB(25)"ANDY ¬
     ¬GAMBLE"
1430 PRINTLEFT$(T$,20)TAB(5);"DO YOU ¬
     ¬NEED INSTRUCTIONS (Y/N)?";:
     ¬GOSUB1720:Q$=Z1$
1440 IFQ$=""THEN1430
1450 IFLEFT$(Q$,1)<>"Y"THEN180
1460 REM INSTRUCTIONS
1470 PRINTCHR$(147)"AN EPIDEMIC HAS ¬
     ¬BROKEN OUT ON A SMALL "
1480 PRINT"ISLAND. THE DISEASE IS NOT ¬
     ¬FATAL, AND"
```



It reteaches the 4 basic arithmetic operations step-by-step.

PET PROFESSOR includes 77 programs that reteach the four fundamental arithmetic operations for whole numbers, fractions and decimals by providing more than just practice drills. Each program includes a complete tutorial sequence that takes a problem apart, then leads the student step-by-step through solving it.

Each program covers a single objective to allow the teacher to select the skill to be reinforced, It states the objective, supplies vocabulary, displays an example, then leads the student step-by-step through other examples. When the student is ready, a quiz requiring writing is given, then scored by the computer. Every program has been carefully selected. All problems were written by teachers and tested in classrooms..

Use PET PROFESSOR to supplement classroom instruction in individualized or traditional programs. Available for PET 2001 Series. Will run within 8K on any PET.

- Whole numbers—32 programs @ $235.00
- Fractions—24 programs @ $175.00
- Decimals—21 programs @ $175.00

Buy all 3 for $499.00. Save $86.00!

Send $5.00 for more info., sample tape & shipping (deducted from total purchase).

List of other educational programs available on request.

Melad Associates, Inc.
66 Main Street, South River, N.J. 08882 • (201) 238-7330

By Netronics

ASCII/BAUDOT, STAND ALONE Computer Terminal

COMPLETE FOR ONLY . . . $149.95

ASCII/BAUDOT
Computer Terminal Kit

The Netronics ASCII/BAUDOT Computer Terminal Kit is a microprocessor-controlled, stand alone keyboard/terminal requring no computer memory or software. It allows the use of either a 64 or 32 character by 16 line professional display format with selectable baud rate, RS232-C or 20 ma. output, full cursor control and 75 ohm composite video output.

The keyboard follows the standard typewriter configuration and generates the entire 128 character ASCII upper/lower case set with 96 printable characters. Features include onboard regulators, selectable parity, shift lock key, alpha lock jumper, a drive capability of one TTY load, and the ability to mate directly with almost any computer, including the new Explorer/85 and ELF products by Netronics.

The Computer Terminal requires no I/O mapping and includes 1k of memory, character generator, 2 key rollover, processor controlled cursor control, parallel ASCII/BAUDOT to serial conversion and serial to video processing—fully crystal controlled for superb accuracy. PC boards are the highest quality glass epoxy for the ultimate in reliability and long life.

VIDEO DISPLAY SPECIFICATIONS

The heart of the Netronics Computer Terminal is the microprocessor-controlled Netronics Video Display Board (VID) which allows the terminal to utilize either a parallel ASCII or BAUDOT signal source. The VID converts the parallel data to serial data which is then formatted to either RS232-C or 20 ma. current loop output, which can be connected to the serial I/O on your computer or other interface, i.e., Modem.

When connected to a computer, the computer must echo the character received. This data is received by the VID which processes the information, converting to data to video suitable to be displayed on a TV set (using an RF modulator) or on a video monitor. The VID generates the cursor, horizontal and vertical sync pulses and performs the housekeeping relative to which character and where it is to be displayed on the screen.

Video Output: *1.5 P/P into 75 ohm (EIA RS-170)* • **Baud Rate:** *110 and 300 ASCII* • **Outputs:** *RS232-C or 20 ma. current loop* • **ASCII Character Set:** *128 printable characters—*

!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~

BAUDOT Character Set: *A B C D E F G H I J K L M N O P Q R S T U V W X Y Z - ? : • 3 $ # ( ) . , 9 0 1 4 ! 5 7 ; 2 / 6 8* • **Cursor Modes:** *Home, Backspace, Horizontal Tab, Line Feed, Vertical Tab, Carriage Return. Two special cursor sequences are provided for absolute and relative X-Y cursor addressing* • **Cursor Control:** *Erase, End of Line, Erase of Screen, Form Feed, Delete* • **Monitor Operation:** *50 or 60Hz (jumper selectable.*

Continental U.S.A. Credit Card Buyers Outside Connecticut
CALL TOLL FREE 800-243-7428
To Order From Connecticut Or For Technical Assistance, Etc. **Call (203) 354-9375**

Netronics R&D Ltd., Dept.
333 Litchfield Road, New Milford, CT 06776

Please send the items checked below—

- ☐ **Netronics Stand Alone ASCII Keyboard/Computer Terminal Kit, $149.95** plus $3.00 postage & handling.
- ☐ **Deluxe Steel Cabinet** for Netronics Keyboard/Terminal In Blue/Black Finish, **$19.95** plus $2.50 postage and handling.
- ☐ **Video Display Board Kit** alone (less keyboard), **$89.95** plus $3 postage & handling.
- ☐ **12" Video Monitor** (10 MHz bandwidth) fully assembled and tested, **$139.95** plus $5 postage and handling.
- ☐ **RF Modulator Kit** (to use your TV set for a monitor), **$8.95** postpaid.
- ☐ **5 amp Power Supply Kit** In Deluxe Steel Cabinet (±8VDC @ 5 amps, plus 6-8 VAC), **$39.95** plus $2 postage & handling.

Total Enclosed (Conn. res. add sales tax) $______
By—
☐ Personal Check ☐ Cashiers Check/Money Order
☐ **Visa** ☐ **Master Charge** (Bank # ____)
Acct. # ______
Signature ______ Exp. Date ______
Print Name ______
Address ______
City ______
State ______ Zip ______
☐ **Send Me More Information**

NEW!!!-THE ELECTRIC MOUTH*

for S100, Elf II, Apple, TRS-80 Level II*

From $99.95 kit

Now — teach your computer to talk, dramatically increasing the interaction between you and your machine.

That's right: the ELECTRIC MOUTH actually lets your computer talk! Installed and on-line in just minutes, it's ready for spoken-language use in office, business, industrial and commercial applications, in games, special projects, R&D, education, security devices — there's no end to the ELECTRIC MOUTH's usefulness. Look at these features:

- Supplied with 143 words/letters/phonemes/numbers, capable of producing hundreds of words and phrases.
- Expandable on-board up to thousands of words and phrases (just add additional speech ROMs as they become available).
- Four models, which plug directly into S100, Apple, Elf II and TRS-80 Level II computers.
- Get it to talk by using either Basic or machine language (very easy to use, complete instructions with examples included).
- Uses National Semiconductor's "Digitalker" system.
- Includes on-board audio amplifier and speaker, with provisions for external speakers and amplifier.
- Adds a new dimension and excitement to programming; lets you modify existing programs and games to add spoken announcements of results, warnings, etc.
- Installs in just minutes.

"Elf II" and "The Electric Mouth" are reg. trademarks of Netronics R&D Ltd. "Apple" is a reg. trademark of Apple Computer Inc. "TRS-80 Level II" is a reg. trademark of Tandy Corp.

Principle of Operation The ELECTRIC MOUTH stores words in their digital equivalents in ROMs. When words, phrases, and phonemes are desired, they are simply called for by your program and then synthesized into speech. The ELECTRIC MOUTH system requires none of your valuable memory space except for a few addresses if used in memory mapped mode. In most cases, output ports (user selectable) are used.

Spoken Material Included

ONE	SEVENTY	CONTROL	LEFT	POUND
TWO	EIGHTY	DANGER	LESS	PULSES
THREE	NINETY	DEGREE	LESSER	RATE
FOUR	HUNDRED	DOLLAR	LIMIT	RE
FIVE	THOUSAND	DOWN	LOW	READY
SIX	MILLION	EQUAL	LOWER	RIGHT
SEVEN	ZERO	ERROR	MARK	SS
EIGHT	AGAIN	FEET	METER	SECOND
NINE	AMPERE	FLOW	MILE	SET
TEN	AND	FUEL	MILLI	SPACE
ELEVEN	AT	GALLON	MINUS	SPEED
TWELVE	CANCEL	GO	MINUTE	STAR
THIRTEEN	CASE	GRAM	NEAR	START
FOURTEEN	CENT	GREAT	NUMBER	STOP
FIFTEEN	400HERTZ TONE	GREATER	OF	THAN
SIXTEEN	80HERTZ TONE	HAVE	OFF	THE
SEVENTEEN	20MS SILENCE	HIGH	ON	TIME
EIGHTEEN	40MS SILENCE	HIGHER	OUT	TRY
NINETEEN	80MS SILENCE	HOUR	OVER	UP
TWENTY	160MS SILENCE	IN	PARENTHESIS	VOLT
THIRTY	320 MS SILENCE	INCHES	PERCENT	WEIGHT
FORTY	CENTI	IS	PLEASE	
FIFTY	CHECK	IT	PLUS	
SIXTY	COMMA	KILO	POINT	

Also, the complete alphabet, spoken letter by letter.

Continental U.S.A. Credit Card Buyers Outside Conn. **TO ORDER CALL TOLL FREE 800-243-7428**
To Order From Connecticut Or For Technical Assistance, Etc., **Call (203) 354-9375**

NETRONICS R&D LTD. Dept.
333 Litchfield Road, New Milford, CT 06776

Please send the items checked below:

- ☐ S100 "Electric Mouth" kit $99.95
- ☐ Elf II "Electric Mouth" kit $99.95
- ☐ Apple "Electric Mouth" kit $119.95
- ☐ TRS-80 Level II "Electric Mouth" kit $119.95

Add $20.00 for wired & tested units. All plus $3.00 postage & insurance. Conn. res. add sales tax.

Total Enclosed $______

☐ Personal Check ☐ Cashier's Check/Money Order
☐ Visa ☐ Master Charge (Bank No. ______)
Acct. No. ______
Signature ______ Exp. Date ______
Print Name ______
Address ______
City ______
State ______ Zip ______

```
1490 PRINT"ONCE CONTRACTED PROVIDES ¬
     ¬IMMUNITY FOR   LIFE."
1500 PRINT:PRINT"YOU ARE ALLOWED TO ¬
     ¬CHOOSE CERTAIN "
1510 PRINT"INITIAL CONDITIONS:":PRINT:
     ¬PRINT:PRINTTAB(5)"THE ISLAND ¬
     ¬POPULATION"
1520 PRINT:PRINTTAB(5)"THE NUMBER ¬
     ¬ORIGINALLY INFECTED"
1530 PRINT:PRINTTAB(5)"THE NUMBER OF ¬
     ¬DAYS FOR WHICH THE"
1540 PRINTTAB(5)"DISEASE IS CONTAGIOUS ¬
     ¬- THIS IS"
1550 PRINTTAB(5)"ALSO THE DURATION OF ¬
     ¬THE DISEASE."
1560 PRINT:PRINT:PRINT"THE ISLANDERS ¬
     ¬WILL MOVE ABOUT RANDOMLY,"
1570 PRINT"INFECTING OTHERS IF CONTAGIOU
     ¬S.";
1580 PRINT"THOSE WHOARE IMMUNE WILL NOT ¬
     ¬BE  INFECTED. "
1590 PRINT: GOSUB1250
1600 PRINTCHR$(147)"THIS WILL CONTINUE ¬
     ¬UNTIL THE DISEASE HASRUN ITS ¬
     ¬COURSE ";
1610 PRINT"(UNTIL THERE ARE NO MORE ¬
     ¬INFECTED PERSONS)."
1620 PRINT:PRINT"YOU WILL THEN BE GIVEN ¬
     ¬A DAY-BY-DAY BAR CHART OF THE ";
1630 PRINT"HISTORY OF THE EPIDEMIC, "
1640 PRINT"UP TO A MAXIMUM OF 50 DAYS."
1650 PRINT:GOSUB1250
1660 GOTO180
1670 DATA32902,32911,32940,32955
1680 DATA32978,32996,33015,33040,33050,
     ¬33083,33090,33123,33130,33162
1690 DATA33171,33203,33213,33244,33254,
     ¬33285,33297,33325
1700 DATA33337,33362,33376,33398,33417,
     ¬33437,33459,33475,33501,33515
1710 DATA33544,33556,33585,33592
1720 Z$="":Z1$=""
1730 PRINT"&<";:FORI=1TO50:NEXTI
1740 PRINT" <";:FORI=1TO50:NEXTI
1750 GETZ$:IFZ$=""THEN1730
1760 IFZ$<>CHR$(20)THEN1810
1770 IFZ$=""THEN1730
1780 ZZ=LEN(Z1$):IFZZ<1THEN1730
1790 Z1$=LEFT$(Z1$,ZZ-1):PRINT"<";
1800 GOTO1730
1810 IFZ$=CHR$(13)ORZ$=CHR$(141)THEN1850
1820 PRINTZ$;
1830 Z1$=Z1$+Z$
1840 GOTO1730
1850 FORI=1TO10:GETZ$:NEXTI
1860 PRINT
1870 RETURN
```

©

Odds & Ends on the 2040 Disk

Jim Butterfield

Verify works this way: all blocks on disk are freed. Then the processor works through the files in the directory, one by one, and re-allocates the blocks it finds in use there. You should not verify a disk that contains direct files unless special provision is made to allow this. Otherwise, the blocks will be freed and not re-allocated — which means trouble.

©

A Floating Point Multiplication Routine

Marvin L. De Jong
Department of Mathematics-Physics
The School of the Ozarks
Pt. Lookout, MO 65726

Introduction

In two previous articles in **COMPUTE!** we have described:

1) A routine that inputs any signed number with magnitude between 1.70141183*E38 to 1.46936795*E-39 and converts it to a floating-point binary number.

2) A routine that outputs a signed floating-point binary number to an output device in BCD code.

In this article we add a floating-point multiplication routine to this set of routines that will eventually become a four-function floating-point package with nine digit accuracy.

The Floating-Point Multiplication Routine

A floating-point multiplication routine is given in Listing 1, and its flowchart is shown in Figure 1. The flowchart is essentially the same as that of B. Hashizume (**BYTE, V2**, Number 11, November 1977, p76). Studying the flowchart and the program comments should make the process understandable.

The multiplication routine uses three accumulators. Accumulator A occupies locations $0000 through $0003 with the most-significant byte in location $0000. Since the mantissa is normalized, there will always be a one in Bit 7 of location $0000, unless the mantissa is identical to zero. Location $0004 is used as a "guard" byte to do a 40-bit multiplication. The 40-bit result is rounded to 32 bits, giving approximately nine-digit decimal accuracy. Accumulator B occupies locations $0020 through $0023, with a guard byte in location $0024, an exponent (twos complement code) in location $0025, and a sign ($FF for minus, $00 for plus) in location $0027. The routine multiplies the contents of accumulator A with the contents of accumulator B. Intermediate results are stored in RES from $0010 to $0014.

The accumulator architecture just described proved to be very convenient for the multiplication routine. However, it differs slightly from the accumulator architecture used in the routines described in previous articles of this series. Rather than modify those two routines, which would not be difficult if you wish to try, we have included a little subroutine in Listing 2 that adjusts the accumulator used by the input routine to conform to the accumulator used in the multiply routine. Thus, after the BCD to Floating-Point Binary routine is called, the subroutine in Listing 2 must be called.

Once the accumulator is properly adjusted, it is moved to Accumulator B to await multiplication. The BCD to Floating-Point Binary routine is then called again to get the second number. Its accumulator is again adjusted to make it Accumulator A. Then the multiply routine is called, and finally the Floating-Point Binary to BCD routine is called to output the answer. This entire process is accomplished by the program in Listing 5, and this program can be used to test all three programs for proper operation.

One very important note. The BCD to Floating-Point Binary routine must be modified with the instruction listed in Listing 4 in order for it to work with the multiplication routine. The change is simple. Modify the byte at $0E02 from $20 to $1F. This prevents Accumulator B's most significant byte from being cleared whenever the BCD to Floating-Point Binary routine is called.

And a final note. If the combination of exponents to form the exponent of the result produces an overflow (exponent larger than 127 or exponent smaller than –128), the multiplication routine executes a BRK instruction. Normally this will send control back to the monitor, but one could write an interrupt routine to signal an overflow or an underflow.

Listing 1. The Floating-Point Multiplication Routine.

```
$0000 = ACCA; Most-significant byte of accumulator A.
$0005 = ACCX; Exponent for accumulator A.
$0007 = ACCS; Sign byte for accumulator A.
$0020 = ACCB; Most-significant byte of accumulator B.
$0025 = BCCX; Exponent for accumulator B.
$0027 = BCCS; Sign byte for accumulator B.
$0010 = RES; Most-significant byte of "result" accumulator.
$0014 = GRDR; "Guard" byte for "result" accumulator.
$0C28 A5 07   START   LDA ACCS       Determine the sign of the result.
 0C2A 45 27           EOR BCCS       Positive sign if signs are alike,
 0C2C 85 07           STA ACCS       negative otherwise.
 0C2E 18              CLC            To multiply, add exponents.
 0C2F A5 05           LDA ACCX
 0C31 65 25           ADC BCCX
 0C33 50 01           BVC ARND       Break to monitor if an exponent
 0C35 00              BRK            overflow (or underflow) results.
 0C36 85 05   ARND    STA ACCX       Store result into EXPONENT.
 0C38 A2 04           LDX #$04       Clear the locations that store
 0C3A A9 00           LDA #$00       the result for the mantissa.
 0C3C 05 10   HERE    STA RES,X
 0C3E CA              DEX
 0C3F 10 FB           BPL HERE
 0C41 A0 28           LDY #$28       Do a 40 ($28) bit multiplication
 0C43 A2 FB   BR2     LDX #$FB       starting here.
 0C45 18              CLC
 0C46 76 25   BACK    ROR ACCB+5,X   Rotate Multiplier right into carry.
 0C48 E8              INX
 0C49 D0 FB           BND BACK
 0C4B 90 0C           BCC PAST       No carry; don't add.
 0C4D A2 04           LDX #04        Add Multiplicand to Result.
 0C4F 18              CLC
 0C50 B5 00   MORE    LDA ACCA,X
 0C52 75 10           ADC RES,X
 0C54 95 10           STA RES,X
 0C56 CA              DEX
 0C57 10 F7           BPL MORE
 0C59 A2 FB   PAST    LDX #$FB       Shift Result right one bit.
 0C5B 76 15   BR1     ROR RES+5,X
 0C5D E8              INX
 0C5E D0 FB           BNE BR1
 0C60 88              DEY            Back for another bit in the
 0C61 D0 E0           BNE BR2        multiplier?
 0C63 A5 10   BR4     LDA RES        Check for zero result.
 0C65 F0 3F           BEQ OUT        If so, get out.
 0C67 30 14           BMI DETOUR     Check if mantissa is already
 0C69 A2 04           LDX #04        normalized.
 0C6B 18              CLC
 0C6C A5 05           LDA ACCX       For each shift left, decrement
 0C6E E9 00           SBC #00        exponent.
 0C70 50 01           BVC BR8        Overflow set?
 0C72 00              BRK            Yes, go to monitor.
 0C73 85 05   BR8     STA ACCX
 0C75 18              CLC
 0C76 36 10   BR3     ROL RES,X
 0C78 CA              DEX
 0C79 10 FB           BPL BR3
 0C7B 30 E6           BMI BR4
 0C7D A5 14   DETOUR  LDA GRDR       If most-significant bit of guard
```

```
0C7F 10 1C          BPL  BR5        byte is one, then round up.
0C81 38             SEC
0C82 A2 03          LDX  #03
0C84 B5 10    BR6   LDA  RES,X
0C86 69 00          ADC  #00
0C88 95 10          STA  RES,X
0C9A CA             DEX
0C9B 10 F7          BPL  BR6
0C8D 90 0E          BCC  BR5        Did rouding produce a carry from
0C8F A9 80          LDA  #$80       the mantissa?
0C91 85 10          STA  RES        Yes. Fix mantissa.
0C93 38             SEC             And adjust exponent.
0C94 A5 05          LDA  ACCX
0C96 69 00          ADC  #00
0C98 50 01          BVC  BR9        Check for overflow.
0C9A 00             BRK             Jump to monitor on overflow.
0C9B 85 05    BR9   STA  ACCX
0C9D A2 03    BR5   LDX  #03        Move result to accumulator for
0C9F B5 10    BR7   LDA  RES,X      the output (Binary to BCD) routine.
0CA1 95 01          STA  ACCA+1,X
0CA3 CA             DEX
0CA4 10 F9          BPL  BR7
0CA6 60       OUT   RTS             Get out.
```

Listing 2. A Subroutine to Modify the Accumulator of the BCD-to-Binary Routine.

```
$0FB0 A0 08   SUB1  LDY  #08        Rotate the accumulator one byte
 0FB2 A2 04   B2    LDX  #04        (eight bits) left.
 0FB4 18            CLC
 0FB5 36 00   B1    ROL  ACCA,X
 0FB7 CA            DEX
 0FB8 10 FB         BPL  B1
 0FBA 88            DEY
 0FBB D0 F5         BNE  B2
 0FBD 60            RTS
```

Listing 3. A Subroutine to Transfer Accumulator A to Accumulator B.

```
$0FC0 A2 07   SUB2  LDX  #07        Move ACCA to ACCB.
 0FC2 B5 00   B3    LDA  ACCA,X
 0FC4 95 20         STA  ACCB,X
 0FC6 CA            DEX
 0FC7 10 F9         BPL  B3
 0FC9 60            RTS
```

Listing 4. An IMPORTANT Modification to the BCD-to-Binary Routine.

```
$0E01 A2 1F   MODIFY  LDX  #$1F     The multiply routine will not work
                                    without this modification.
```

Listing 5. An Input/Output/Multiply Calling Program.

```
$0050 20 00 0E    JSR  INPUT        Call the BCD to Floating-Point Binary Routine.
 0053 20 B0 0F    JSR  SUB1         Call the subroutine to modify the accumulator.
 0056 20 C0 0F    JSR  SUB2         Transfer ACCA to ACCB (it takes two to multiply),
 0059 20 00 0E    JSR  INPUT        and get the second number.
 005C 20 B0 0F    JSR  SUB1         Fix the accumulator again.
 005F 20 28 0C    JSR  MULTIPLY     Multiply the two numbers using
 0062 20 00 0B    JSR  OUTPUT       Listing 1 in this article. Then
                                    output the result using the Floating-Point
 0065 00          BRK               Binary to BCD Routine.
```

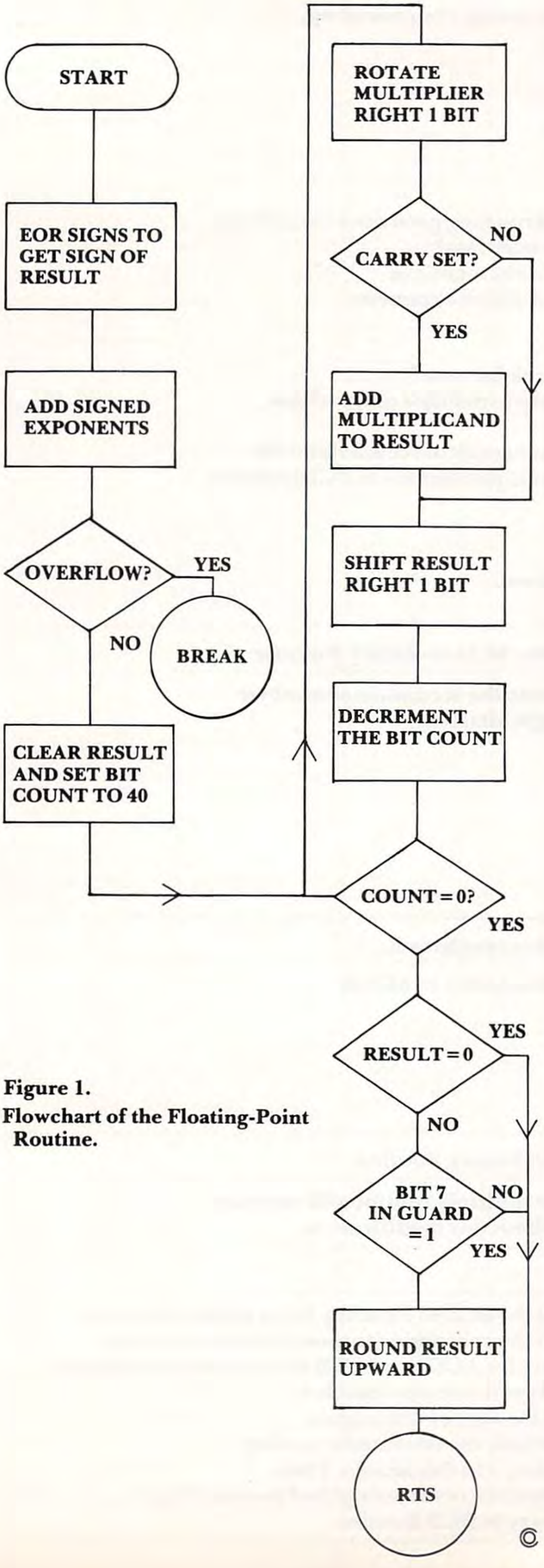


Figure 1.
Flowchart of the Floating-Point Routine.

©

Professional Business Software

For The Commodore 32K Microcomputer System With 2040 Dual Drive Disk & 2022 Tractor Feed Printer

General Ledger

- Holds Up To 300 Accounts.
- Accepts Up To 3000 Transactions Per Month.
- Cash Disbursements Journal, Cash Receipts Journal, and Petty Cash Journal for simplified data entry.
- Maintains Account Balances For Present Month, Present Quarter, Present Year, Three Previous Quarters, And Previous Year.
- Complete Financial Reports Including Trial Balance, Balance Sheet, Profit & Loss Statement, Cash Receipts Journal, Cash Disbursements Journal, Petty Cash Journal and more.
- Accepts Postings From External Sources Such As Accounts Payable, Accounts Receivable, Payroll, Etc. . . . .

Accounts Payable

- Interactive Data Entry With Verified Input And Complete Operator Prompting.
- Automatic Application Of Credit And Debit Memos.
- Maintains Complete Purchase Records For Up To 200 Vendors.
- Invoice File Accepts Up To 400 Invoices.
- Random Access File Organization Allows Fast Individual Record Updating.
- Multiple Reports Provide A Complete Audit Trail.
- Check Printing With Full Invoice Detail.
- Full Invoice Aging.
- Automatic Posting To General Ledger . . . .

Accounts Receivable

- Maintains Invoice File For Up To 300 Invoices.
- Accomodates Full Or Partial Invoice Payments.
- Customer File Maintains Purchase Information For Up To 1000 Customers.
- Allows For Automatic Progress Billing.
- Provides For Credit And Debit Memos As Well As Invoices.
- Prints Individualized Customer Statements.
- Interactive Data Entry With FullOperator Prompting.
- Complete Data Input Verification And Formating.
- Automatic Posting To General Ledger . . . .

Payroll

- Maintains Monthly, Quarterly, And Yearly Cumulative Totals For Each Employee.
- Payroll Check Printing With Full Deduction And Pay Detail.
- Sixteen Different Reports Including W2 And 941.
- Interactive Data Entry With Easy Correction Of Entry Errors.
- Automatic Data Verification.
- Complete Job Costing Option With Cumulative Totals And Overhead Calculations.
- Random Access File Organization For Fast Updating Of Individual Records.
- Automatic Posting To General Ledger . . . .

Structured around the time tested and reliability proven series of business software systems developed by Osborne and Associates, these programs have been designed to fill the need of a comprehensive accounting package for the new Commodore PET micro computer system. Each program can either stand alone, or be integrated with the others in a total software system.

Designed with the first time user in mind, these programs lead the operator through step by step, verified data entry. It is impossible to 'crash' a program due to operator error or invalid data input. Design consistency has been maintained from program to program to greatly increase operator familiarity and confidence.

Documentation, normally a problem for small systems users, is provided by the comprehensive series of Osborne and Associates user manuals. These three manuals together total over 800 pages of detailed step by step instructions written at three levels for DP Department Managers, Data Entry Operators, and Programmers. You don't have to worry about getting 'promises' instead of documentation because the documentation was written before the programs were developed. A second set of manuals details any changes required during conversion. Each program provided on disk with complete documentation. Packaged in a handsome three ring binder with pockets and twelve monthly dividers for convenient storage of reports.

See your nearest Commodore dealer for a demonstration of this outstanding business software system.

CMS Software Systems

5115 MENEFEE DRIVE • DALLAS, TX 75227 • 214-381-0690

Naming Compounds

Tony A. Hartman
Texarkana, AR

Chemistry students seem to have less trouble 're-membering' names of elements and radicals when seated in front of a computer. The prefixes, suffixes and symbols used in nomenclature seem less confusing. Students seem to be able to calculate subscripts faster when challenged by the 'answer machine'. Students begin to rely less on lists of valences and sometimes need not even consult a periodic chart for the proper valences.

Try this program after you have 'hammered away' at valences and 'harped on' using the correct suffix in naming. In this program, answers are typed in exactly as they would be written on paper, except for the placement of subscripts on the screen (on the screen, SUBSCRIPTS are on the same line as the symbol). I think the program can best be utilized after practice and drill on naming compounds and writing formulas. I have found that students working in pairs, carefully selected, have shown the best response. The tendency to 'let the machine answer the hard ones' is lessened when working in pairs.

The following program was written on a PET computer for use in high school chemistry classes. As written, the program uses about 6K of memory. It will run as is on any model PET – original, upgrade, or 4.0 ROM. There are many statements which could be omitted or combined if you are interested in making it more compact.

The elements and radicals used in the compounds are some of the more commonly encountered ones. Students should be familiar with most of the symbols and valences. The names of elements and radicals used in the program can be changed easily as you will see later.

Well, enough of that. I am sure you will find an effective and practical way to use the program. Here is a summary of the program by line numbers:

30-130	Prints title, gives choice of writing names or formulas
140-170	Randomly chooses a name (called from line 880 & 990)
180-200	Delay a few seconds (used in the instructions)
210-250	Prints message and waits for space bar (called throughout)
260-310	Reads data statements
320-450	Compares valences and assigns subscripts
460-510	Displays 'correct' on the screen and increments correct answer counter
520-730	Instructions for writing formulas
740-860	Prints compound name on screen and asks for formula
870-990	Sets number of elements, calls subroutine to choose name and assign subscripts, sets the correct formula
1000-1280	Instructions for writing names of compounds
1290-1430	Uses subroutine 870 to randomly choose a compound
1440-1500	Prints student average and comment
1510-1580	Additional instructions
1590-1650	Comments on scores
1660-1760	Additional instructions
1770-1930	Data statements containing metal groups
1940-2060	Data statements containing nonmetal groups

The following is a summary of the variables used. Hopefully, this will help you to interpret and adapt the program a little easier if that is what you want to do.

c	number of correct answers
e$	name of element
s$	symbol of element
v%	valence of element
e1$	name of metal group
s1%	symbol of metal group
v1%	valence of metal group
e2$	name of nonmetal group
s2$	symbol of nonmetal group
v2%	valence of nonmetal group
n	number of metal/nonmetal ions listed in data statements
f$	formula of compound given by student input
f1$	correct formula of compound calculated by PET
n$	name of compound given by student input
n1$	correct name of compound
1$	line of graphic symbols printed on screen
s1%	subscript of metal group
so%	student score as a percent
t	try (student gets two tries to answer correctly)
x	random number
z%	number read to keep data statement pointer at the right spot
z$	strings read to keep data statement pointer at the right spot

What about personalizing the program? The statements which print the directions can be changed to 'your language'. You can change or take out the delay loop. Change the data statements to include more or different elements or radicals. If you change the number of elements, be sure to change the value of the variable n in line 880 to correspond to the number of metal groups and the value of n in line 990 to correspond to the number of nonmetal groups. Also, changing the comments to your own witty remarks will spark some interest.

One final note. I was reluctant to send an article to a nationally known magazine. I felt less competent than some because of a lack of formal computer training. But I am convinced that for educators to share their ideas on computers, programs and the use of these, we must all put aside our feelings of inadequacy and start sharing what we have. I look forward to seeing more science programs (or any programs for that matter) from you educators who have been holding back!

Editor's Note: Me too! RCL

```
VARIABLE REFERENCES

     c    490   500   530  1450
    e$    280   760   890  1310
   e1$    760   890  1310
    f$    780   790   800
   f1$    800   830   950   960   970   980  1330
     i     40    50    60    70   190   270   300   460
    l$     20   530   610   630   710   730   770   820   840  1010
    l$   1090  1100  1180  1190  1270  1320  1340  1390  1410  1460
    l$   1470  1490  1510  1570  1670  1750
     n    150   160   290   300   880   900
    n$   1350  1360  1370
   n1$   1310  1370  1400
     q    130   720   760   850  1280  1330  1420
    q$    110   120   130   230   240
    s$    280   890   940   950   960   970   980
    s1    340   390   430   450   930   950   960   970   980
   s1$    890   930   950   960   970   980
    s2    340   390   430   450   940   950   960   970   980
   sc%   1450  1460  1600  1610  1620  1630  1640
     t    490   760   800   820   830  1330  1370  1390  1400
    ti     20
    v%    280   890   910
   v1%    330   360   370   390   410   430   450   890
   v2%    330   360   370   390   410   430   450   910
     x     20   150   160   270   290   300
    z$    300
    z%    300
```

```
LINE REFERENCES
  40  1500
 110   110   120
 140   880   900
 150   160
 180   640   650   660   670   680   690   700  1110  1120  1130
 180  1140  1150  1160
 210   610   710   840  1090  1180  1270  1410  1500  1580  1750
 230   230   240
 260   880   900
 310   290
 320   920
 350   400   440   450
 360   330
 390   370
 410   360
 430   410
 450   380   420
 460   810  1380
 510   490
 520   130
 730   850
 780   790   820
 820   800
 840   810
 870   750  1300
1000   130
1300  1420
1350  1360  1390
1390  1370
1410  1380
1440   860  1430
1510   720  1280
1590  1480
1660   620
```

```
10 POKE59468,14
20 L$="((((((((((((((((((((((((((((((((((((((
     ¬(((((((":X=RND(-TI)
30 REM PRINTS BOX CLOCKWISE
40 PRINT"ĥ";:FORI=1TO39:PRINT"&";:NEXT
50 FORI=1TO23:PRINT"&←↓";:NEXT
60 FORI=1TO39:PRINT"&←←";:NEXT
70 FORI=1TO24:PRINT"&↑←";:NEXT
80 PRINT"h↓↓↓↓"TAB(12)"NAMING COMPOUNDS"
90 PRINT"↓↓"TAB(6)"CHOOSE ONE OF THE ¬
     ¬FOLLOWING:"
100 PRINT"↓↓"TAB(12)"1. WRITE FORMULAS":
     ¬PRINT"↓↓"TAB(12)"2. WRITE NAMES"
110 GETQ$:IFQ$=""THEN110
120 IFQ$<>"1"ANDQ$<>"2"THEN110
130 Q=VAL(Q$):ONQGOTO520,1000
140 REM SUBRTN TO CHOOSE A NAME 0<X<18
150 X=INT(N*RND(1)+0.5)
160 IFX>NORX<=0THEN150
170 RETURN
180 REM SUBRTN TO DELAY A FEW SECONDS
190 FORI=1TO1000:NEXT
200 RETURN
210 REM SUBRTN TO PRINT MESSAGE AND WAIT
220 PRINT"↓PRESS THE rSPACEr̂ BAR TO ¬
     ¬CONTINUE.
```

```
230 GETQ$:IFQ$=""THEN230
240 IFQ$<>CHR$(32)THEN230
250 RETURN
260 REM SUBRTN TO READ NAMES, FORMULAS
270 FORI=1TOX
280 READE$,S$,V%:NEXT
290 IFX=NTHEN310
300 FORI=X+1TON:READZ$,Z$,Z%:NEXT
310 RETURN
320 REM SUBRTN TO COMPARE VALENCE AND ¬
    ¬RETURN SUBSCRIPTS
330 IFV1%<>V2%THEN360
340 S1=1:S2=1
350 RETURN
360 IFV1%>V2%THEN410
370 IFV2%/V1%=2ORV2%/V1%=3ORV2%/V1%=4THE
    ¬N390
380 GOTO450
390 S1=V2%/V1%:S2=1
400 GOTO350
410 IFV1%/V2%=2ORV1%/V2%=3ORV1%/V2%=4THE
    ¬N430
420 GOTO450
430 S2=V1%/V2%:S1=1
440 GOTO350
450 S1=V2%:S2=V1%:GOTO350
460 FORI=1TO20:IFINT(I/2)=I/2THENPRINT"r
    ¬";
470 PRINT"*CORRECT*r̂
480 PRINT"↑↑":NEXT
490 IFT=1THENC=C+1:GOTO510
500 C=C+0.5
510 RETURN
520 REM WRITE FORMULAS WHEN GIVEN NAME
530 C=0:PRINT"ĥ↓"L$"↓"
540 PRINT"YOU WILL BE GIVEN THE NAME OF ¬
    ¬A COMPOUND
550 PRINT"AND ASKED TO WRITE THE ¬
    ¬FORMULA.  YOU
560 PRINT"↓MUST USE A SPECIFIC FORM IN ¬
    ¬ANSWERING
570 PRINT"↓THESE.  TO WRITE THE ¬
    ¬FORMULAS ON THE
580 PRINT"↓SCREEN, YOU CANNOT USE ¬
    ¬SUBSCRIPTS.
590 PRINT"↓YOU MUST TYPE IN THE ¬
    ¬SUBSCRIPT ON THE
600 PRINT"↓SAME LINE AS THE ELEMENT ¬
    ¬SYMBOL.↓          (INSTRUCTIONS ¬
    ¬CONTINUED)
610 PRINTL$:GOSUB210
620 GOSUB1660
630 PRINT"ĥ↓"L$"↓FOR EXAMPLE, TO WRITE ¬
    ¬THE FORMULA FOR
640 PRINT"↓THE COMPOUND CALCIUM ¬
    ¬CHLORIDE, YOU WOULD↓WRITE:↓":
    ¬GOSUB180:GOSUB180
650 PRINT"C";:GOSUB180:PRINT"A";:
    ¬GOSUB180:PRINT"C";:GOSUB180:
    ¬PRINT"L";:GOSUB180:PRINT"2":
    ¬GOSUB180
660 PRINT"↓FOR THE COMPOUND POTASSIUM ¬
    ¬SULFATE:↓":GOSUB180
670 PRINT"K";:GOSUB180:PRINT"2";:
    ¬GOSUB180:PRINT"S";:GOSUB180:
    ¬PRINT"O";:GOSUB180:PRINT"4":
    ¬GOSUB180
680 PRINT"↓FOR THE COMPOUND ZINC ¬
    ¬NITRATE:↓":GOSUB180
690 PRINT"Z";:GOSUB180:PRINT"N";:
```

```
      ¬GOSUB180:PRINT"(";:GOSUB180:
      ¬PRINT"N";:GOSUB180:PRINT"Q";
700 GOSUB180:PRINT"3";:GOSUB180:
      ¬PRINT")";:GOSUB180:PRINT"2":
      ¬GOSUB180
710 PRINT"↓"L$:GOSUB210
720 Q=0:GOSUB1510
730 PRINT"ĥ↓"L$
740 REM SELECT NAME AND ASK FOR FORMULA
750 GOSUB870
760 Q=Q+1:PRINTQ". "E1$" "E$:T=0
770 PRINT"↓"L$"↓"
780 INPUT"FORMULA  _←←←";F$
790 IFF$="_"THENPRINT"↑↑":GOTO780
800 T=T+1:IFF$<>F1$THEN820
810 PRINT"↓":GOSUB460:GOTO840
820 IFT=1THENPRINT"↓WRONG!   TRY ¬
      ¬AGAIN.":PRINT"↓"L$:GOTO780
830 IFT=2THENPRINT"↓WRONG AGAIN!    ¬
      ¬FORMULA IS "F1$
840 PRINT"↓"L$:GOSUB210
850 IFQ<10THEN730
860 GOTO1440
870 REM SUBRTN TO CHOOSE NAME AND FORM
880 N=17:GOSUB140:GOSUB260
890 E1$=E$:S1$=S$:V1%=V%
900 N=13:GOSUB140:GOSUB260
910 RESTORE:V2%=ABS(V%)
920 GOSUB320
930 IFLEN(S1$)>2ANDS1>1THENS1$="("+S1$+"
      ¬)"
940 IF(LEN(S$)>2ANDS2>1)OR(S$="OH"ANDS2>
      ¬1)THENS$="("+S$+")"
950 IFS1<>1ANDS2<>1THENF1$=S1$+RIGHT$(ST
      ¬R$(S1),1)+S$+RIGHT$(STR$(S2),1)
960 IFS1<>1ANDS2=1THENF1$=S1$+RIGHT$(STR
      ¬$(S1),1)+S$
970 IFS1=1ANDS2<>1THENF1$=S1$+S$+RIGHT$(
      ¬STR$(S2),1)
980 IFS1=1ANDS2=1THENF1$=S1$+S$
990 RETURN
1000 REM WRITE NAMES WHEN GIVEN FORMULAS
1010 PRINT"ĥ↓"L$"↓YOU WILL BE GIVEN A ¬
      ¬FORMULA AND YOU
1020 PRINT"↓WILL BE ASKED TO WRITE THE ¬
      ¬NAME OF THE
1030 PRINT"↓COMPOUND.  SPELLING ¬
      ¬DEFINITELY COUNTS.
1040 PRINT"↓SO YOU WILL NEED TO BE ¬
      ¬CAREFUL WITH THE
1050 PRINT"↓ENDINGS SUCH AS 'ITE' AND ¬
      ¬'ATE' AND ALL
1060 PRINT"↓OTHER SPELLINGS AS WELL.  ¬
      ¬TYPE THE NAMES
1070 PRINT"WITHOUT USING CAPITAL ¬
      ¬LETTERS.
1080 PRINT"↓(INSTRUCTIONS CONTINUED)↓
1090 PRINTL$:GOSUB210
1100 PRINT"ĥ↓"L$"↓FOR EXAMPLE, TO WRITE ¬
      ¬THE NAME FOR KCL  ↓YOU WOULD ¬
      ¬WRITE:↓"
1110 GOSUB180
1120 GOSUB180:PRINT"POTASSIUM ";:
      ¬GOSUB180:PRINT"CHLORIDE↓":GOSUB180
1130 PRINT"FOR CU↓2↑O:↓":GOSUB180
1140 PRINT"↓COPPER";:GOSUB180:PRINT"(I) ¬
      ¬";:GOSUB180:PRINT"OXIDE↓":GOSUB180
1150 PRINT"FOR NA↓2↑SO↓4:":GOSUB180
1160 PRINT"↓SODIUM ";:GOSUB180:PRINT"SUL
      ¬FATE↓":GOSUB180
1170 PRINT"(INSTRUCTIONS CONTINUED)↓
1180 PRINTL$:GOSUB210
1190 PRINT"ĥ↓"L$"↓BE SURE TO INDICATE ¬
      ¬MULTIVALENT ELEMENTS
1200 PRINT"WITH THE ROMAN NUMERAL IN ¬
      ¬PARENTHESIS.
1210 PRINT"↓THE ROMAN NUMERAL MUST BE ¬
      ¬IN PARENTHESIS
1220 PRINT"NEXT TO THE METAL IT GOES ¬
      ¬WITH.  USE A
1230 PRINT"↓(I) FOR ONE, (II) FOR TWO,
      ¬ (III) FOR
1240 PRINT"↓THREE, (IV) FOR FOUR AND ¬
      ¬(V) FOR FIVE.
1250 PRINT"↓NOTE THAT THE ROMAN ¬
      ¬NUMERALS ARE CAPITAL
1260 PRINT"LETTERS.
1270 PRINTL$:GOSUB210
1280 Q=0:GOSUB1510
1290 REM SELECT NAME WRITE FORMULA
1300 GOSUB870
1310 N1$=E1$+" "+E$
1320 PRINT"ĥ↓"L$
1330 Q=Q+1:PRINTQ". "F1$:T=0
1340 PRINT"↓"L$"↓"
1350 INPUT"NAME  _←←←";N$
1360 IFN$="_"THENPRINT"↑↑":GOTO1350
1370 T=T+1:IFN$<>N1$THEN1390
1380 PRINT"↓":GOSUB460:GOTO1410
1390 IFT=1THENPRINT"↓WRONG!    TRY ¬
      ¬AGAIN.":PRINT"↓"L$:GOTO1350
1400 IFT=2THENPRINT"↓WRONG AGAIN!    ¬
      ¬NAME IS "N1$
1410 PRINT"↓"L$:GOSUB210
1420 IFQ<10THEN1300
1430 GOTO1440
1440 REM CALCULATE PERCENT & DISPLAY
1450 SC%=C/10*100
1460 PRINT"ĥ↓"L$"↓YOUR AVERAGE IS ¬
      ¬"SC%"%"
1470 PRINT"↓"L$"↓↓"
1480 GOSUB1590
1490 PRINT"↓"L$"↓"
1500 GOSUB210:GOTO40
1510 PRINT"ĥ↓"L$"↓YOU WILL BE GIVEN 10 ¬
      ¬PROBLEMS, ONE AT A
1520 PRINT"↓TIME.  YOU WILL HAVE TWO ¬
      ¬CHANCES TO
1530 PRINT"↓ANSWER CORRECTLY.  IF YOU ¬
      ¬ANSWER CORRECT
1540 PRINT"THE FIRST TIME, YOU GET 10 ¬
      ¬POINTS. IF
1550 PRINT"↓YOU ANSWER CORRECT ON THE ¬
      ¬SECOND TRY,
1560 PRINT"↓YOU GET 5 POINTS.
1570 PRINT"↓"L$
1580 GOSUB210:RETURN
1590 REM COMMENTS FOR SCORE
1600 IFSC%>=90THENPRINT"↓VERY GOOD! YOU ¬
      ¬MAY MAKE A CHEMIST!":RETURN
1610 IFSC%>=80THENPRINT"↓OK! ARE YOU IN ¬
      ¬ENRICHED CHEMISTRY??":RETURN
1620 IFSC%>=70THENPRINT"↓REALLY!! YOU ¬
      ¬CAN DO BETTER THAN THAT!":RETURN
1630 IFSC%>=60THENPRINT"↓COME ON!  DO ¬
      ¬HAVE A CHEMISTRY BOOK??":RETURN
1640 IFSC%>=50THENPRINT"↓YOU WERE ¬
      ¬READING THE QUESTIONS WEREN'T  ¬
      ¬YOU!!!":RETURN
```

```
1650 PRINT"↓DID YOU SIGN UP FOR THIS ¬
     ¬CLASS ALL BY   YOURSELF???":RETURN
1660 REM SUBRTN TO SUPPLEMENT INSTRUCTIO
     ¬NS ON WRITING FORMULAS
1670 PRINT"ĥ↓"L$"↓THE FIRST LETTER OF ¬
     ¬THE SYMBOL MUST BE
1680 PRINT"↓CAPITALIZED AND THE SECOND ¬
     ¬LETTER LOWER-
1690 PRINT"CASE AS THEY ARE USUALLY ¬
     ¬WRITTEN.
1700 PRINT"↓WHEN A POLYATOMIC ION WHICH ¬
     ¬ALREADY CON-
1710 PRINT"TAINS A SUBSCRIPT IS TO BE ¬
     ¬SUBSCRIPTED,
1720 PRINT"↓THE ION MUST BE IN PARENTHES
     ¬IS WITH THE
1730 PRINT"↓SUBSCRIPT OUTSIDE.
1740 PRINT"↓(INSTRUCTIONS CONTINUED)
1750 PRINT"↓"L$"↓":GOSUB210
1760 RETURN
1770 DATA HYDROGEN,"H",1
1780 DATALITHIUM,"LI",1
1790 DATASODIUM,"NA",1
1800 DATAPOTASSIUM,"K",1
1810 DATABERYLLIUM,"BE",2
1820 DATACALCIUM,"CA",2
1830 DATAMAGNESIUM,"MG",2
1840 DATABARIUM,"BA",2
1850 DATAZINC,"ZN",2
1860 DATAALUMINUM,"AL",3
1870 DATA"COPPER(I)","CU",1
1880 DATA"COPPER(II)","CU",2
1890 DATA"IRON(II)","FE",2
1900 DATA"IRON(III)","FE",3
1910 DATA"LEAD(II)","PB",2
1920 DATA"LEAD(IV)","PB",4
1930 DATAAMMONIUM,"NH4",1
1940 DATAFLUORIDE,"F",-1
1950 DATACHLORIDE,"CL",-1
1960 DATABROMIDE,"BR",-1
1970 DATAIODIDE,"I",-1
1980 DATAOXIDE,"O",-2
1990 DATASULFIDE,"S",-2
2000 DATASULFATE,"SO4",-2
2010 DATASULFITE,"SO3",-2
2020 DATANITRATE,"NO3",-1
2030 DATANITRITE,"NO2",-1
2040 DATAHYDROXIDE,"OH",-1
2050 DATACARBONATE,"CO3",-2
2060 DATAPHOSPHATE,"PO4",-3
```

©

THE apple® GAZETTE

Using Named GOSUB And GOTO Statements In Applesoft Basic

M. R. Smith

Using subroutines greatly improves the readability of a program and makes it easier to debug. However remembering what a particular GOSUB does is often difficult. Was it GOSUB 1000 or GOSUB 2000 that was wanted?

One of the nice features of Integer Apple Basic is its ability to let you give a name as well as a number in GOSUB statements. The following Integer program demonstrates this:

```
10   GOSUB 100
20   SUB1 =100
30   GOSUB SUB1
40   STOP
100  PRINT "HERE" : RETURN
```

Typing this program whilst using Applesoft will lead to the error message "UNDEFINED STATEMENT IN 30".

The purpose of this program is to show how to use names GOSUB and GOTO statements within Applesoft. By loading the short machine language program described in this article, you are able to run the Applesoft program.

```
10   GOSUB 100
20   SUB1 = 100
30   & GOSUB SUB1
40   STOP
100  PRINT "HERE" : RETURN
```

For the murky details of how it works read the section "PROGRAM DESCRIPTION". Otherwise, type in the demonstration BASIC program and type RUN. The program includes a routine to check that the DATA statements have been entered correctly. Once the demo program has run correctly, the machine language program can be saved using BSAVE NAMED.GOSUB,A$300,L$43. To have the program ready for future sessions, simply type BRUN NAMES.GOSUB as the first part of your programming session. This will load and fix the code. It will remain ready but out of your way until you power down.

WARNING: If you use a RENUMBER program to reorder your program statements, you must remember that variables are NOT changed. Therefore your subroutine pointers will not be renumbered; you'll have to do that by hand.

WARNING: The instructions GOSUB and ON. . .GOSUB are entirely different. The machine code given here will not allow the statement ON X & GOSUB FNAME, SNAME.

Machine Language Program Description

The first statement (at $D93E) of the Applesoft Interpreter GOTO subroutine is the reason that Applesoft does not handle GOSUB's and GOTO's in the same manner as Integer Basic. This statement goes and gets an integer number for use within the GOTO. This means that the BASIC statement GOSUB 1000 is okay but N = 1000 : GOSUB N is not allowed as N as a variable.

Now changing these memory locations to cause the next EXPRESSION to be evaluated, rather than the next NUMBER, allows us to use named GOSUB's. To change these actual locations is impossible. Instead the GOSUB and GOTO routines must be relocated lower in memory at $300 (768) where they can be changed. The Apple's ampersand instruction (&) can then be used to make the new commands operate.

Lines 19–25. Set the ampersand vector (&) at $3F5.

Lines 27–32. Check for GOSUB or GOTO tokens after the &.

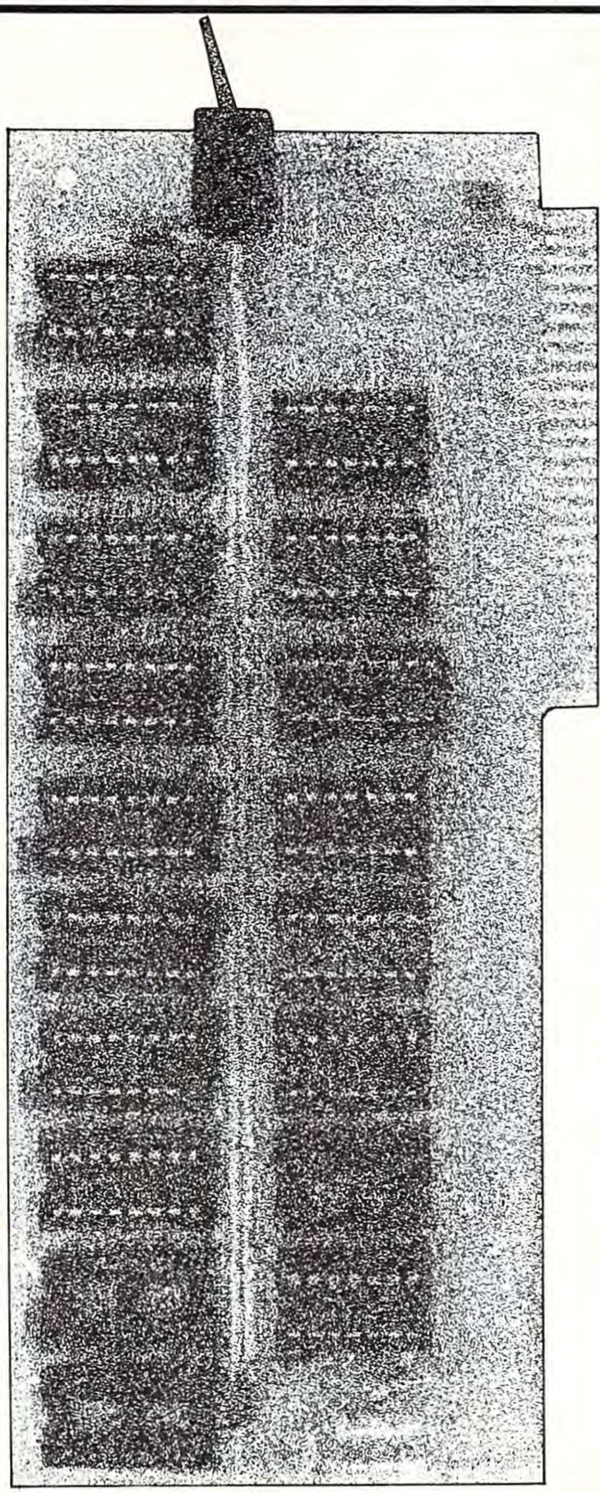
Lines 34–47. Relocated version of the monitor GOSUB routine. This now calls the new front end of the GOTO routine.

Lines 49–52. New front end to drive the monitor GOTO routine. It jumps into the middle of the old GOTO routine.

Lines 50 and **51** are the actual major changes.

BASIC Program Description.

Lines 20 and **5000–5200.** The program first checks that the DATA statements have been correctly entered. Each pair of DATA statements consists of 16 numbers and a checksum which is the previous 16 numbers added together. If this 17th number is not the actual sum of the previus 16 numbers, then an error is indicated. If all the statements are okay, then the code is loaded.



Line 40. Sets the ampersand vector. This is not necessary if the machine code is BRUN into memory but is necessary if the code is BLOADed.
Lines 60–80. These set the subroutine names.
Lines 100–140. Demonstrate the new instructions.
Lines 1000–3020. Demonstration Subroutines.

References

"AMPERSAND-INTERPRETER" by R. M. Mottala in **Nibble #6**, 1980, p27.
"APPLESOFT INTERNAL ENTRY POINTS" by Apple Computer Inc. in **Apple Orchard**, March/April 1980, p12.
"SOME ROUTINES IN APPLESOFT BASIC" by J. Butterfield in **COMPUTE!**, September/October 1980, p68.


```
]RUN
OKAY
OKAY
OKAY
OKAY
OKAY
BLOAD OKAY


THIS IS JOHN
HERE BY A NAMED GOSUB

THIS IS PETE
HERE BY A DIFFERENT NAMED GOSUB

THIS IS PHREDD
HERE BY A NAMED GOTO

BREAK IN 3020
]&FF
```

```
0300- A9 4C 8D F5 03 A9 10 8D
0308- F6 03 A9 03 8D F7 03 60
0310- C9 B0 F0 09 C9 AB F0 1F
0318- A2 10 4C 12 D4 A9 03 20
0320- D6 D3 A5 B9 48 A5 B8 48
0328- A5 76 48 A5 75 48 A9 B0
0330- 48 20 37 03 4C D2 D7 20
0338- B1 00 20 7B DD 20 52 E7
0340- 4C 41 D9
*
 00 00 00 00 00
*
```

Got A Question You'd Like Answered?

Write: Ask The Readers, c/o **COMPUTE!**, P.O. Box 5406, Greensboro, NC 27403

```
10   REM  LOAD THE ROUTINE -
     NORMAL GOSUB
20   GOSUB 5000
30   REM  ESTABLISH THE
     AMPERSAND VECTOR
40   CALL 768
50   REM ESTABLISH NAMES OF SUBROUTINES
60 JOHN = 1000
70 PETE = 2000
80 PHREDD = 3000
90   REM  CALL THE SUBROUTINES
100  &  GOSUB JOHN
110  &  GOSUB PETE
120  &  GOTO PHREDD
130  PRINT "DIDNOT WORK"
140  STOP
1000  PRINT "THIS IS JOHN"
1010  PRINT "HERE BY A NAMED GOSUB"
1020  PRINT : RETURN
2000  PRINT "THIS IS PETE"
2010  PRINT "HERE BY A DIFFERENT
      NAMED GOSUB"
2020  PRINT : RETURN
3000  PRINT "THIS IS PHREDD"
3010  PRINT "HERE BY A NAMED GOTO"
3020  STOP
4990  REM  LOAD IN ROUTINE
5000 LOW = 768:HIGH = 835
5010 OK = 1
5020  REM  LOAD IN GROUP OF SIXTEEN
5030  FOR J = LOW TO HIGH STEP 16
5040 CHECK = 0
5050  FOR K = J TO J + 15
5060  READ IT
5070 CHECK = CHECK + IT
5080  NEXT K
5090  REM  CHECK IF CHECK SUM OKAY
5100  READ SUM
5110 L$ = "OKAY": IF CHECK <  > SUM
     THEN L$ = "BAD":OK = 0
5120  PRINT L$
5130  NEXT J
5140  IF OK = 0 THEN  STOP
5150  REM  THINGS ARE OKAY - LOAD INTO MEMORY
5160  RESTORE : FOR J = LOW TO HIGH STEP 16
5170  FOR K = J TO J + 15: READ IT: POKE K,IT: NEXT K
5180  READ IT: NEXT J
5190  PRINT "BLOAD OKAY": PRINT : PRINT
5200  RETURN
6000  DATA  169,76,141,245,3,169,16,141,246
6010  DATA 3,169,3,141,247,3,96,1868
6020  DATA   201,176,240,9,201,171,240,31,162
6030  DATA 16,76,18,212,169,3,32,1957
6040  DATA 214,211,165,185,72,165,184,72,165
6050  DATA 118,72,165,117,72,169,176,2322
6060  DATA 72,32,55,3,76,210,215,32,177
6070  DATA 0,32,123,221,32,82,231,1593
6080  DATA  76,65,217,0,0,0,0,0,0
6090  DATA  0,0,0,0,0,0,0,358
```

```
SOURCE FILE: GOSUB1
------ NEXT OBJECT FILE NAME IS GOSUB1.OBJ0
0300:                     ORG  $300
0300:              ;
0300:              ; M.R.SMITH  APRIL 1981
0300:              ;
0075:              CLINL   EQU  $75        ;CURRENT LINE NUMBER
0076:              CLINH   EQU  $76
00B1:              GETCHR  EQU  $B1        ;GET NEXT CHAR
00B8:              TXTPTL  EQU  $B8        ;TEXT POINTER
00B9:              TXTPTH  EQU  $B9
0300:              ;
D3D6:              STACK   EQU  $D3D6      ;CHECK ON STACK POINTER
D412:              WRONG   EQU  $D412      ;PRINT SYNTAX ERROR MESSAGE
D7D2:              NGOSUB  EQU  $D7D2      ;JUMP INTO NORMAL MONITOR GOSUB
D941:              NGOTO   EQU  $D941      ;JUMP INTO NORMAL MONITOR GOTO
DD7B:              FRMEVL  EQU  $DD7B      ;PUSH VALUE IN FAC
E752:              FIXGOTO EQU  $E752      ;USE FAC AS GOTO POINTER
0300:              ;
0300:              ;FIX AMPERSAND VECTOR
0300:A9 4C         FIX     LDA  #$4C
0302:8D F5 03              STA  $3F5
0305:A9 10                 LDA  #$10
0307:8D F6 03              STA  $3F6
030A:A9 03                 LDA  #$3
030C:8D F7 03              STA  $3F7
030F:60                    RTS
0310:              ;
0310:C9 B0         ENTRY   CMP  #$B0       ;IS IT GOSUB?
0312:F0 09                 BEQ  GOSUB
0314:C9 AB                 CMP  #$AB       ;IS IT GOTO?
0316:F0 1F                 BEQ  GOTO
0318:A2 10                 LDX  #$10       ;FORCE SYNTAX ERROR MESSAGE
031A:4C 12 D4              JMP  WRONG
031D:              ;
031D:A9 03         GOSUB   LDA  #$3        ;NORMAL GOSUB PROCEDURE
031F:20 D6 D3              JSR  STACK      ;RELOCATED FROM $D921
0322:A5 B9                 LDA  TXTPTH     ;STORE CURRENT TEXT POINTERS
0324:48                    PHA
0325:A5 B8                 LDA  TXTPTL
0327:48                    PHA
0328:A5 76                 LDA  CLINH      ;STORE CURRENT LINE NUMBER
032A:48                    PHA
032B:A5 75                 LDA  CLINL
032D:48                    PHA
032E:A9 B0                 LDA  #$B0       ;IT NEEDS THIS
0330:48                    PHA
0331:20 37 03              JSR  GOTO       ;DO A GOTO
0334:4C D2 D7              JMP  NGOSUB     ;CONTINUE NORMAL GOSUB
0337:              ;
0337:20 B1 00      GOTO    JSR  GETCHR     ;GET NEXT CHAR
033A:20 7B DD              JSR  FRMEVL     ;EVALUATE NEXT EXPRESSION
033D:20 52 E7              JSR  FIXGOTO    ;FIX GOTO LOCATION
0340:4C 41 D9              JMP  NGOTO      ;CONTINUE NORMAL GOTO ROUTINE
```

©

Commas, Colons And Quote Marks Too

Craig Peterson
Santa Monica, CA

Have you ever wanted to be able to input commas, colons or quotation marks as part of an input statement to one of your Applesoft programs? But, hard as you may try, Applesoft kept coming back with "EXTRA IGNORED." Contact 4 from Apple Computer, Inc., helped you by suggesting the use of the GET statement, but all that B$ = B$ + A$ stuff meant that you often had to endure string garbage cleanup delays. Then Contact 6 seemed to offer the ultimate solution, totally avoiding garbage collection. But was it? Besides requiring a small machine language program, there was a subtle problem you might not have been aware of. The input routine used to fill the input buffer made no allowance for the high bit of each character in the input line. The routine used to fill the input buffer left the high bit set, just as it comes from the keyboard. But Applesoft wants the high bit to be zero for its string characters. The line will print correctly and will look on the screen just like what you typed in, but if you ever try an IF IN$ = "Q", you'll never get a match. Or if you try to VAL (IN$),when IN$ was input as "1234", you'll get a value of 0.

The solution to this dilemma is in the program listed below. The subroutine shown in lines 1000 to 1020 (for Applesoft ROM Basic) will gather any input for you and place it into the variable IN$, even commas, colons and quote marks. The only exempt characters are the standard keyboard escape sequences. So, who is the little man at 54572? Well, he's the Applesoft equivalent of the monitor's keyboard input routine, with the difference being that he strips the high bit from all of the input characters. So line 1000 fills the input buffer with normal Applesoft string characters gathered from the keyboard. Line 1010 finds the length of the string, and line 1020 finds the IN$ variable and stuffs its pointers with the right info to point to the keyboard buffer. Then IN$ is relocated into RAM, away from the keyboard buffer. It is not necessary for IN$ to be the first variable used in the program. Lines 1000-1020 can be placed anywhere in your program. The pointers for IN$ are found through the magic of locations 131 and 132, which hold the address of the pointers for the last used variable. It's fast, it totally avoids string garbage build-up, and it's done in Basic. None of that nasty machine language stuff.

One additional note. Not only does this routine work slick for keyboard input, but it also performs the same super feat for disk input, which can be real handy. Commas, etc., in the middle of a name file cause no difficulty when read from the disk. Please note, however, that this routine limits the size of an input string to 239 characters just like the Applesoft INPUT statement does.

So if you need it, try it. It's an easy solution to a common problem.

```
10   HOME : VTAB 4: PRINT "INPUT A
      NYTHING THAT YOU WANT..": PRINT
      : GOSUB 1000: PRINT : PRINT
      "VOILA..": PRINT : PRINT IN$
      : END
20 :
30   REM   LINES 1000 TO 1020 ARE
      A SUBROUTINE THAT PUTS ANY
      INPUT INTO IN$
40 :
1000  CALL 54572
1010  FOR B = 512 TO 751: IF  PEEK
      (B) <  > 0 THEN  NEXT
1020 IN$ = "": POKE  PEEK (131) +
     256 *  PEEK (132) + 1,0: POKE
      PEEK (131) + 256 *  PEEK (1
     32) + 2,2: POKE  PEEK (131) +
     256 *  PEEK (132),B - 512:IN
     $ =  MID$ (IN$,1): RETURN
```

©

Generating Lower Case Text On The Apple II Plus Using The Paymar Chip

David Shapiro
Bloomington, IN

Introduction

The following program will allow lower case text to be displayed on an Apple II Plus which is equipped with a Paymar chip. The hardware requirements involve the "older" Apple with RAM configuration blocks (an "I.C. impersonator" which only contains jumper wires and is labeled with "16K"), and the PAYMAR lower case adapter, presently advertised as the "original LCA-1 (TM)". By appending this routine to a BASIC program, lower case characters can be embedded inside of quotation marks following a PRINT command by simply converting the corresponding upper case character in the given string. When the BASIC statement involving the PRINT command and the string are executed, the display of upper/lower case text is immediate. Lower case characters can also be converted back to upper case using this routine.

Sample Use Of The Lower Case Converter

Once this routine is appended to a BASIC program, it can then be used for converting between upper and lower case characters:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

A typical program statement may contain the string "POUR THE SOLUTIONS." and lower case conversion may be desired on all characters after the "P". The following brief example initially LISTs the statement containing this string, the lower case converter program (which starts at line number 63000) is then RUN, and finally the statement containing the now-converted text is reLISTed.

```
]LIST20
20 PRINT "POUR THE SOLUTIONS."
]RUN63000
WHAT LINE DO YOU WANT CONVERTED? 20
I HAVE FOUND THE LINE.
POUR THE SOLUTIONS.
DO YOU WANT TO CHANGE ANYTHING?
START WITH WHICH CHARACTER? 2
END WITH WHICH CHARACTER? 16
Pour the solutions.
DO YOU WANT TO CHANGE ANYTHING?
]LIST20
20 PRINT "Pour the solutions."
```

The program initially prompts the user for the line number of the BASIC statement to be converted. A search through the Apple's RAM continues until that line number is found, whereupon the characters within quotatin marks are then displayed (if no such line number exists, the program informs the user). A decision to change the string contents is then entered (Y in this case). Character limits for the conversion are individually entered, with only the characters from the upper/lower case sets (see above) sequentially counted (the spaces on either side of "THE" were ignored). The conversion will then start with "0" (the 2nd character) and terminate with the final "S" (the 16th character), with the resultant form displayed for more changes. No further changes were made (input of "N"), and the RESET key was pressed to terminate execution of this routine. This particular statement was then re-LISTed, displaying the quote-embedded lower case text.

More Lower Case Converter Details

The case conversion occurs between the user-defined limits in a continuous fashion. If there are two (or more) separated segments in the same string that are to be converted, then each segment conversion must be done individually. The string is re-displayed after each conversion for further changes if so desired. An individual character can also be converted if the lower and upper numerical limits are identical.

The first time "RUN 63000" is executed, the search for the input line number commences at the beginninng of the program. This search examines the appropriate locations in RAM which the program currently occupies, and with each new examination moves sequentially through the program (increasingly higher memory locations) in an attempt to find the line number. A variable (ML) contains the current RAM location when the line is eventually found. After making the necessary character changes in this statement as stipulated by the user, the search for the next line number will begin at this present memory location (ML). This optimizes the speed with which the program searches for the next line number. If the next line number is less than the last line number, or if it does not exist in the program, then the current RAM location variable ML is re-initialized to zero. The user is informed that the line can not be

found, and the next line number search must start at the beginning of the program. This unnecessarily increases the search time; therefore, for maximum speed-execution of the program, all entered line numbers must exist in the program, and they should be entered by increasing value.

The case conversion between upper/lower case in reciprocal; i.e., designated upper case characters will be converted to lower case, and lower case characters will be changed to upper case. Also, if the cursor is used to read a BASIC statement containing a string, any lower case characters will be converted back to upper case (an easy method for converting a mixed-case string to all upper case).

The line numbering of this routine begins at 63000 since lower line numbers should always be used when writing a BASIC program. It may be entered after the END command and accessed at the user's convenience. Typing "RUN 63000" from the keyboard RUNs the routine; pressing the RESET key will terminate its execution.

Program Listing And Explanation

63000 Line number to be converted input as LN.
63010 Initialization of ML to start of BASIC on first RUN of program or when line number is not found; ML is the memory location currently being examined.
63020 NL equated to RAM location of start of next BASIC statement. TL is equated to the line number of BASIC currently being examined.
63030 Jump from search loop if line number is found.
63040 Jump from search loop if line number is not found.
63050 Equate ML to RAM location of the next BASIC statement.
63070 Loop to examine each character/token in the current BASIC statement. Check for quotation mark (ASCII code =34). MODE is a "toggle"; set to 0 when first quote is found.
63080 Printing of characters after 1st quote and up to 2nd quote.
63090 Close PRINT loop.
63100 If no changes ("N") execution transferred to 63000. All other input (including "Y") defaults to 63110.
63110-63120 Limits to define character conversion.
63130 Loop examination of each character/token in BASIC statement. When 1st quote is found, MODE is set to 0.
63140 If the character is between quotes and alphabetic, then counter PO is incremented. When the counter is between the stipulated character limits, the character is converted to upper case (add 32) or lower case (subtract 32) depending on the original value of Q.
63150 Close conversion loop. Control transferred to 63070 for any further changes.

```
63000  INPUT "WHAT LINE DO YOU WA
     NT CONVERTED? ";LN
63010  IF ML = 0 THEN ML = 256 *
      PEEK (104) +  PEEK (103)
63020 NL =  PEEK (ML) + 256 *  PEEK
     (ML + 1):TL =  PEEK (ML + 2)
      + 256 *  PEEK (ML + 3)
63030  IF TL = LN THEN 63060
63040  IF NL < ML OR TL > LN THEN
      PRINT "LINE NOT FOUND.":ML =
     0: PRINT : GOTO 63000
63050 ML = NL: GOTO 63020
63060  PRINT "I HAVE FOUND THE LI
     NE."
63070  PRINT :MODE = 1: FOR A = M
     L + 4 TO NL:Q =  PEEK (A): IF
     Q = 34 THEN MODE = 1 - MODE
63080  IF MODE = 0 AND Q <  > 34 THEN
      PRINT  CHR$ (Q);
63090  NEXT : PRINT
63100  PRINT : PRINT "DO YOU WANT
      TO CHANGE ANYTHING? ";: GET
     A$: PRINT : IF A$ = "N" THEN
     63000
63110  INPUT "START WITH WHICH CH
     ARACTER? ";S
63120  INPUT "END WITH WHICH CHAR
     ACTER? ";E:PO = 0
63130 MODE = 1: FOR A = ML + 4 TO
     NL:Q =  PEEK (A): IF Q = 34 THEN
     MODE = 1 - MODE
63140  IF MODE = 0 AND Q > 64 AND
     Q < 128 THEN PO = PO + 1: IF
     PO >  = S AND PO <  = E THEN
      POKE A,Q + 32: IF Q > 96 THEN
      POKE A,Q - 32
63150  NEXT : GOTO 63070
```

©

Apple Authors

COMPUTE! is looking for applications articles aimed at beginners and intermediate programmers. We're specifically interested in programming hints, tutorials, articles written to help users get more out of their machine.

THE ATARI® GAZETTE

A cure for Atari BASIC or Make Your Atari A Bit Wiser

Charles Brannon

As pointed out by Glen Fisher and Ron Jeffries in "The Ouch in Atari BASIC" (**COMPUTE!**, January/February 1980), the keywords AND and OR in Atari BASIC do not let you "get at" the individual bits of a number, as Microsoft BASIC does. Where PRINT 127 and 64 would give 64 in Microsoft BASIC, the Atari interprets the command as PRINT (not zero) AND (not zero) and returns "1". Although this is fine for logical comparisons (e.g. IF A = 12 and B = 22 THEN PRINT A$), it makes bit hakcers a little angry.

If you do not appreciate why, let me explain. Besides the logical uses of AND and OR, it is often advantageous to use these operands for bit manipulation. This is most important in preparing a byte for a POKE command, or interpreting one that was read with PEEK. Being able to process a number on the binary level gives more "bite" to a computer's number crunching abilities. For example, a major use of the AND operator is to *mask* a number, that is, zeroing out some of the bits in a number. The ASCII value of "3" is 51, or $33 hexadecimal. This looks like %00110011 in binary. If the leftmost four bits (the left nibble) could be cleared, we would have the numerical value of the character "3". The action takes place on the binary level.

	51	**= 0011 0011**	**binary**
if we AND with	**15**	**0000 1111**	
we get		**0000 0011**	**= 3 in decimal**

The AND is performed bit by bit. Refer to the **truth table** for AND. Therefore, the Microsoft BASIC command to mask the left four bits would be:

PRINT 51 and 15

The computer would respond with "3".

The OR operator is commonly used to force bits into a byte. For example: a reverse field character is specified by a one in bit seven (the leftmost one). To force a character to print in reverse field, we just OR its ASCII value with 128.

ASC("A") = 65	**=**	**01000001**	**binary**
if we OR with		**1000000**	**(128)**
we get		**11000001**	**193 (reverse field "A")**

Once again, refer to the truth table for OR for details.

One other very useful operand is EOR (Exclusive OR). Unfortunately, virtually no BASIC provides this function. It is used commonly to "flip a bit", that is, if a bit is exclusive OR'd with a one, then the opposite bit results. If a number is exclusive OR'd with all ones (255), then the complement is formed.

10101011	**171**	**11000001**	**193 (reverse "A")**
11111111	**255**	**10000000**	**128**
01010100	**84**	**01000001**	**65 (normal "A")**

Perhaps now you can see why these operators are so useful. But why am I tormenting you? Didn't I say that Atari BASIC doesn't have this capability? Ah, too true, but once again – machine language comes to the rescue. Listing one is the assembly language program that will simulate the bitwise operators. (For 6502 programmers, notice the sequence CLC, BCC OUT. This will simulate an unconditional jump, yet the code remains relocatable.) Listing two is the BASIC program that will load the program into a protected area of memory. At least I think it is protected. **The Atari BASIC Reference Manual** claims that the area from $600 to $6FF is FREE RAM. If true, then this block of memory could be used like the "second cassette buffer" is used on the PET. When the machine language code is POKE'd here, it should remain there until the power is turned off. Listing three is an example program showing how to use the USR command to call the functions from your programs. It assumes that listing two has already been run. To use the operators in your program, first load the second program. If line 20 is changed to RETURN and the program is appropriately renumbered, then it could be called as a subroutine at the beginning of your program. The machine language program is called by the USR function. This is a truly remarkable command on the Atari, as it can have a variable length list of arguments for the machine language program to deal with. This machine language program uses three arguments. The format is:

A = USR(ML,avar1,key,avar2)

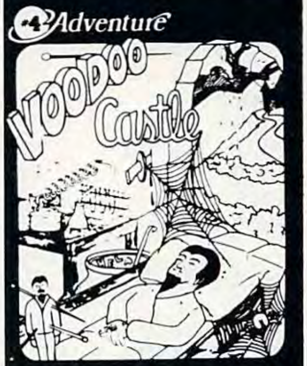
where **ML** is the starting location of the machine language program (1536), **avar1** is the first argument (value 0-255), **avar2** is the second argument

Adventure

BY Scott Adams

JUST RELEASED FOR ATARI

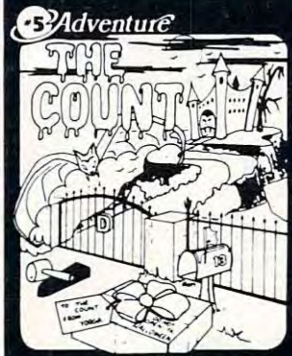
0.✓ **"SPECIAL SAMPLER"** - Never tried ADVENTURE? This special inexpensive sampler complete with 3 Treasures is a cut-down version of our large Adventureland. Guaranteed to supply hours of enjoyment: Try an ADVENTURE today!
ATARI - 24K TAPE to DISK* 051-0000 $ 6.95



1.✓ **ADVENTURELAND** - You wander through an enchanted world trying to recover the 13 lost treasures. You'll encounter wild animals, magical beings, and many other perils and puzzles. Can you rescue the Blue Ox from the quicksand? Or find your way out of the maze of pits? Happy Adventuring....
ATARI - 24K TAPE to DISK* 051-0001 $14.95

2.✓ **PIRATE'S ADVENTURE** - "Yo ho ho and a bottle of rum..." You'll meet up with the pirate and his daffy bird along with many strange sights as you attempt to go from your London flat to Treasure Island. Can you recover Long John Silver's lost treasures? Happy sailing, matey. . . .
ATARI - 24K TAPE to DISK* 051-0002 $14.95

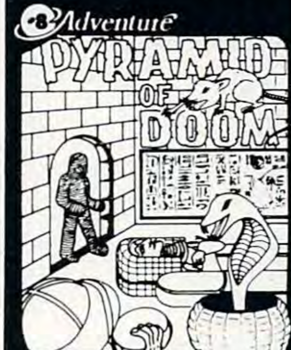
3. **MISSION IMPOSSIBLE ADVENTURE** - Good morning, your mission is to... and so it starts. Will you be able to complete your mission on time? Or is the world's first automated nuclear reactor doomed? This one's well named. It's hard, there is no magic, but plenty of suspense. Good luck. . . .
ATARI - 24K TAPE to DISK* 051-0003 $14.95



4. **VOODOO CASTLE** - Count Cristo has had a fiendish curse put on him by his enemies. There he lies, with you his only hope. Will you be able to rescue him or is he forever doomed? Beware the Voodoo man. . . .
ATARI - 24K TAPE to DISK* 051-0004 $14.95

5. **THE COUNT** - You wake up in a large brass bed in a castle somewhere in Transylvania. Who are you, what are you doing here, and WHY did the postman deliver a bottle of blood? You'll love this ADVENTURE, in fact, you might say it's Love at First Byte. . . .
ATARI - 24K TAPE to DISK* 051-0005 $14.95

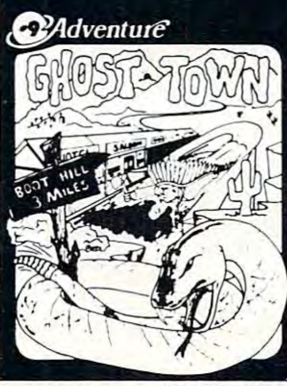
6. **STRANGE ODYSSEY** - Marooned at the edge of the galaxy, you've stumbled on the ruins of an ancient alien civilization complete with fabulous treasures and unearthly technologies. Can you collect the treasures and return or will you end up marooned forever?. . . .
ATARI - 24K TAPE to DISK* 051-0006 $14.95



7. **MYSTERY FUN HOUSE** - Can you find your way completely through the strangest Fun House in existance, or will you always be kicked out when the park closes?. . . .
ATARI - 24K TAPE to DISK* 051-0007 $14.95

8. **PYRAMID OF DOOM** - An Egyptian Treasue Hunt leads you into the dark recesses of a recently uncovered Pyramid. Will you recover all the treasures or more likely will you join its denizens for that long eternal sleep?. . . .
ATARI - 24K TAPE to DISK* 051-0008 $14.95

9. **GHOST TOWN** - Explore a deserted western mining town in search of 13 treasures. From rattlesnakes to runaway horses, this ADVENTURE's got them all! Just remember, Pardner, they don't call them Ghost Towns for nothin'. (Also includes new bonus scoring system!)
ATARI - 24K TAPE to DISK* 051-0009 $14.95



10. **SAVAGE ISLAND — PART 1** - WARNING FOR EXPERIENCED ADVENTURERS ONLY! A small island in a remote ocean holds and awesome secret. Will you be the first to uncover it? NOTE: This is the first of a larger multi-part Adventure;, it will be necessary to purchase additional packages to complete the entire Adventure.
ATARI - 24K TAPE to DISK* 051-0098 $14.95

✓ **RECOMMENDED FOR THE NOVICE ADVENTURER, with many built-in helps**
***TAPE to DISK - a standard cassette tape that runs on a tape machine or can easily be transferred to disk.**

ai Adventure INTERNATIONAL
A DIVISION OF SCOTT ADAMS, INC.
CATALOG & ORDERING TOLL FREE HOT LINE (800) 327-7172
BOX 3435, LONGWOOD, FL 32750 • (305) 862-6917

ADVENTURE Series also available for TRS-80 Models 1, 2 & 3, APPLE, Pet & Scorcerer....Call Today!

Listing 1

```
                 ;******  BOOLEAN FUNCTIONS
                 ;******     FOR THE ATARI
                 ;
                 WHICH      .DE $D0
                 ARG1       .DE $CB
                 ARG2       .DE $CF
                 RTN        .DE $D4
                 ;
                            .BA $0000
                            .OC
                 ;
0000- 68         INIT       PLA
0001- C9 03                 CMP #$03
0003- D0 2C                 BNE OUT
0005- 68                    PLA
0006- 68                    PLA
0007- 85 CB                 STA *ARG1
0009- 68                    PLA
000A- 68                    PLA
000B- 85 D0                 STA *WHICH
000D- 68                    PLA
000E- 68                    PLA
000F- 85 CF                 STA *ARG2
                 ;
0011- A5 D0                 LDA *WHICH
0013- C9 01      AND        CMP #$01
0015- D0 07                 BNE OR
0017- A5 CB                 LDA *ARG1
0019- 25 CF                 AND *ARG2
001B- 18                    CLC
001C- 90 13                 BCC OUT
                 ;
001E- C9 02      OR         CMP #$02
0020- D0 07                 BNE EOR
0022- A5 CB                 LDA *ARG1
0024- 05 CF                 ORA *ARG2
0026- 18                    CLC
0027- 90 08                 BCC OUT
                 ;
0029- C9 03      EOR        CMP #$03
002B- D0 04                 BNE OUT
002D- A5 CB                 LDA *ARG1
002F- 45 CF                 EOR *ARG2
                 ;
0031- 85 D4      OUT        STA *RTN
0033- A9 00                 LDA #$00
0035- 85 D5                 STA *RTN+1
0037- 60                    RTS
                            .EN

LABEL FILE:  [ / = EXTERNAL ]

/WHICH=00D0      /ARG1=00CB      /ARG2=00CF
/RTN=00D4        INIT=0000       AND=0013
OR=001E          EOR=0029        OUT=0031

//0000,0038,0038
```

Listing 3

```
100 REM SAMPLE PROGRAM
110 GRAPHICS 0:ML=1536
120 SCR=PEEK(560)+256*PEEK(561)+4
130 SCR=PEEK(SCR)+256*PEEK(SCR+1)
140 REM
150 REM DEMONSTRATE "EOR"
160 REM
170 FOR I=0 TO 199
180 A=USR(ML,PEEK(SCR+I),3,128)
190 POKE SCR+I,A
200 NEXT I
210 REM
220 REM DEMONSTRATE "AND" & "OR"
230 REM
240 OPEN #1,4,0,"K:"
250 GET#1,KEY
260 PRINT "NORMAL CHARACTER:";
270 A=USR(ML,KEY,1,127))
280 PRINT CHR$(A)
290 PRINT "REVERSED CHARACTER:";
300 PRINT CHR$(USR(ML,A,2,128)
310 REM
320 REM TEST EACH FUNCTION
330 REM
340 GRAPHICS 0
350 PRINT"(ENTER -1 TO STOP)"
360 PRINT "FIRST VALUE";
370 INPUT ARG1
380 IF ARG1=-1 THEN END
390 PRINT "ENTER FUNCTION:"
400 PRINT "1=AND, 2=OR, 3=EOR"
410 INPUT KEY
420 IF KEY<1 OR KEY>3 THEN 390
430 PRINT "SECOND VALUE";
440 INPUT ARG2
450 PRINT USR(ML,ARG1,KEY,ARG2)
460 PRINT:GOTO 350
READY.
```

Listing 2

```
10 ML=1536:FOR I=0 TO 55:READ X:POKE ML+I,X:NEXT I
20 NEW
30 DATA 104,201,3,208,44,104,104,133,203,104,104,133,208,104
40 DATA 104,133,207,165,208,201,1,208,7,165,203,37,207,24
50 DATA 144,19,201,2,208,7,165,203,5,207,24,144,8,201
60 DATA 3,208,4,165,203,69,207,133,212,169,0,133,213,96
READY.
```

of the function to be performed, and **key** is the code for which operator is being used.

1 = AND
2 = OR
3 = EOR

The USR function MUST supply all four variables (ML,avar1,key,avar2) and in proper order or the Atari will "lock-up". It will not respond to the keyboard, necessitating a power off/on reset to regain control.

I have provided here a machine language program that extends Atari BASIC. It would be very useful if others could submit similar programming aids, particularly a graphics extension to use the player/missile graphics. Let's make the most of the USR function to extend Atari BASIC as far as possible.

Truth Tables

0 AND 0 = 0	**0 OR 0 = 0**	**0 EOR 0 = 0**
0 AND 1 = 0	**0 OR 1 = 1**	**0 EOR 1 = 1**
1 AND 0 = 1	**1 OR 1 = 1**	**1 EOR 1 = 0**

©

Odds And Ends

John Girard
Berkeley, CA

Here is an early routine I figured out for the ATARI that encourages people to play with the many sound possibilities.

HYPER DRIVE SIMULATOR

```
100 PRINT"TONE NUMBER";
    INPUT T                       SEE BELOW
110 OPEN#1,4,0,"K:"
120 GET#1,K                       PRESS A KEY TO START
130 FOR I=200 TO 1 STEP-1
140 SOUND 0,I,T,8                 SPACESHIP ACCELERATES
150 FOR J=1 TO 5: NEXT J
160 NEXT I
170 SOUND 0,0,0,0                 KILL SOUND IN HYPERSPACE
180 GET#1,K                       PRESS A KEY TO FINISH
190 FOR I=1 TO 200
200 SOUND 0,I,T,8                 SPACESHIP DECELERATES
210 NEXT I
220 SOUND 0,0,0,0                 ENGINES OFF
230 GO TO 120
```

For even more realistic sounds, the volume can be made to rise and fall with the pitch of the engines:

```
140 SOUND 0,I,T,15-INT(I*.05)
200 SOUND 0,I,T,15-INT(I*.075)
delete line 220
```

Each run of the program requests T, a tone number. Giving T a value of 8 produces a satisfactory rushing noise for the engines. Other interesting values are:

10 — a pure tone
4 — damaged engine
12 — bizarre sounding engines

©

Editor's Note: Here are two variations of screen printers for your Atari. Enjoy them. RCL

Copy Your Atari Screen To Your Printer

Harry A. Straw
Wilmington, DE

Here's a handy routine for copying text from your ATARI screen (GRAPHICS 0 mode) to your printer. It is set up to use two GOSUB commands in your main program:

GOSUB 32010 to initialize.

GOSUB 32040 each time you want to line-print a page displayed on your screen.

The program is straightforward, but a few comments may help you to run it smoothly.

The main business of this program is the double FOR-NEXT loop in lines 32050–32110. With the POSITION command, these loops move the cursor over the entire screen, one position at a time. At each cursor position, line 32080 GETs the ASCII number for the character under the cursor, and line 32090 puts the corresponding character on the printer. Since I have an 80-column printer and the ATARI screen is only 40 characters wide, I need line 32105 to get printer carriage return at the proper place. You may be able to delete this line if you have a 40-column printer (or one that can be set to 40 columns).

Line 32040 (printer carriage return) makes sure that the printer head starts copying at its left-hand margin. Line 32120 "homes" the cursor at the end of the subroutine. This is not always necessary but, depending on the next line in your main program, it may prevent an ERROR - 141, "cursor out of range."

You must OPEN a port to GET from the screen. I use port no. 5, leaving ports 1-4 free for use in main programs. The initializing subroutine in lines 32010–32030 does this. It also expands the ATARI display to its full 40-character width and 24-line height to match the cursor movement controlled by lines 32050 and 32060. The OPEN command clears the screen, so you must OPEN before displaying the text you want to copy. Just be sure your main program says GOSUB 32010 *ahead* of the screen display to be printed.

If you have only a few lines to copy, no problem. Merely adjust line 32050 to cover the rows you want to scan. Otherwise, the printer will run for all 24 rows, printing a lot of blank spaces wherever nothing shows on the screen.

There is no CLOSE no. 5 statement in the listing. This leaves port no. 5 open so it is not necessary to repeat GOSUB 32010 for each page to be line-printed.

Take advantage of ATARI's ability to merge cassette-recorded programs with RAM-resident programs by recording this routine with the LIST"C command and reading the cassette with ENTER"C. CSAVE and CLOAD won't work this way. In fact, CLOAD erases programs in RAM! This routine starts with a high line number, 32000, so its line numbers won't conflict with those of a program already in RAM.

In a future note, we'll discuss copying graphics to a printer.

```
32000 REM - COPY SCREEN TO PRINTER.
32001 REM
32002 REM - "OPEN" CLEARS SCREEN.
32003 REM - DO THIS EARLY IN PROGRAM.
32004 REM - USE "GOSUB 32010" FOR THIS.
32005 REM
32010 POKE 82,0:POKE 83,39
32020 OPEN #5,4,0,"S:"
32030 RETURN
32031 REM
32032 REM - USE GOSUB 32040 TO LPRINT
32033 REM - TEXT FROM SCREEN.
32034 REM
32040 LPRINT CHR$(10)
32050 FOR Y=0 TO 23
32060 FOR X=0 TO 39
32070 POSITION X,Y
32080 GET #5,G
32090 LPRINT CHR$(G);
32100 NEXT X
32105 LPRINT CHR$(13)
32110 NEXT Y
32120 POSITION 0,0
32130 RETURN
```

©

Screen To Printer

Len Lindsay

Here is a simple program, completely in BASIC that will print what is on your screen to your printer. It is designed for the 40 column printer. Thus it can only print 39 characters per line, since printing the 40th character creates an extra line feed. To change to 40 characters per line you can change the 39 in line 32130 to 40.

The program is meant to be used as a subroutine. It depends on these two lines occuring at the beginning of the program first:

20 DIM XC$(39)

40 OPEN #3,4,0 "S:"

Note that the program is reading characters right off the screen. Screen input of this type can be used within other types of programs.

Finally, note that the ATARI printer will not print all the characters as on your screen. Often it will just print a blank space for a character it can't print.

Listing

```
0 REM PRINT SCREEN TO PRINTER
1 REM (C) 1980 LINDSAY
20 DIM XC$(39)
40 OPEN #3,4,0,"S:"
32100 XC$="
          ":REM PRINT SCREEN
32101 REM XC=CHARACTER READ FROM SCREEN
AS ASCII VALUE
32102 REM XLOOP=COL LOOP VARIABLE
32103 REM YLOOP=ROW LOOP VARIABLE
32104 REM XC$=LINE OF CHARACTERS FROM SC
REEN
32105 REM ** INCLUDE A DIM XC$(39)
32106 REM ** INCLUDE THESE AT START
32110 FOR YLOOP=0 TO 23
32120 POSITION 1,YLOOP
32130 FOR XLOOP=1 TO 39
32140 GET #3,XC
32150 XC$(XLOOP,XLOOP)=CHR$(XC)
32160 NEXT XLOOP
32170 LPRINT XC$
32180 NEXT YLOOP
32199 RETURN
```

Sample Output

```
FILENAME IS: DIRPRINT.1
     003 SECTORS
FILENAME IS: DIRPRINT.2
     005 SECTORS
FILENAME IS: DRFACTOR.
     068 SECTORS
FILENAME IS: PRINT.DRF
     009 SECTORS
FILENAME IS: TEST.HST
     001 SECTORS
FILENAME IS: DRFACTOR.HST
     001 SECTORS
FILENAME IS: MENU.
     023 SECTORS
FILENAME IS: PREVHIGH.
     001 SECTORS
FILENAME IS: LEN.HST
     001 SECTORS
FILENAME IS: SCREEN.PRT
     005 SECTORS
FILENAME IS: ROBERT.HST
     001 SECTORS
 SECTORS FREE =527
```

©

Hardware Information

At Last!

Richard Bills
Lisle, IL

For those Atari owners who have been tormented by the inadequacy of information concerning the hardware and other technical aspects, relief is finally here. Three manuals are now available:

Atari 400/800 Technical User Notes (C016555)
Cost: $27.00 plus $3.00 shipping
Includes information on the OS and other hardware as well as schematic diagrams

Atari 400/800 Operating System (OS) (CA016557)
Cost: $17.00 plus $3.00 shipping
Includes machine language routines that reside in the OS

Atari 400/800 Disk Operating System (DOS) (C016558)
Cost: $4.00 plus $1.50 shipping
Includes machine language routines that are used in DOS

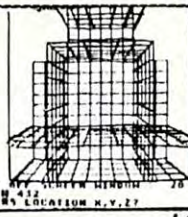
Both OS and DOS can be ordered for $24.00 (price includes shipping). To order a manual enclose a check and letter stating which manual you want to:

Atari Inc.
Customer Support
1346 Bordeaux Drive
Sunnyvale, CA 94086
Attn: Tom Harris

©

ATARI 800 SOFTWARE!
FROM: SEBREE'S COMPUTING
456 Granite Av., Monrovia,CA 91016

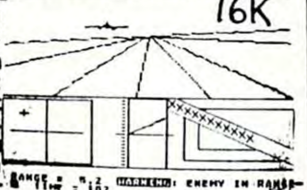
3-DIMENSIONAL GRAPHICS PACKAGE:Allows you to view objects from ANY location in space.12 Programs! 6 on cassette,& the rest in the 32 page manual! Package is designed to be tutorial on the handling of graphics in ATARI BASIC!! Includes instructions on getting five colors up in the HI-RES mode! All the listings are given!ONLY $29.95+.95p&h



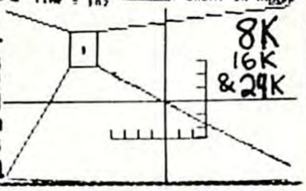
8K TO 48K

3-D RED BARON DOGFIGHT/FLT.SIMULATOR! One of our best selling games! All of the graphics are done in HI-RES with MULTI-COLORS!! Sound Effects! Graphic explosions, RADAR,ALTIMETER,a graphic scoreboard! Frames vary from 6 cycles per second to 2.25 per sec.UPDATED on 3/26/81.Req.Jystik.ONLY $16.95+.95p&h

16K



DOWN THE TRENCH! Another best seller! 2 programs,3 versions.HI-RES w/multi-colors,& low RES.Fast action Graphics explosions, sound effects, animation! Is NOT the same as version published in SOFTSIDE(tm).Much expanded program set. Many options and playing scenes included.REQ.Jystk.ONLY $16.95+.95p&h



8K 16K & 29K

TRS-80 POCKET COMPUTER SOFTWARE
WUMPUS ADVENTURE! Play our very popular program on your P.C. With A 4 page manual with listing. Supplied on cassette. ONLY- $7.95 +.95 p&h

Also,software available for TRS-80 COLOR COMPUTER with EXTENDED BASIC

>>>>>>>>>>>>Write for a free catalog of software today!!<<<<<<<<<<<<

ATARI® Software for Personal Computers

A collection of 10 challenging programs created to provide a unique entertainment value — and 2 personal/business programs with broad functional value. In disk and/or cassette as indicated.

GAMES	• HELICOPTER BATTLE	Req. — 16K RAM/Cassette	$ 9.95
		— 16K RAM/Disk	14.95
	• HORSE RACING	Req. — 16K RAM/Cassette	9.95
		— 16K RAM/Disk	14.95
	• KENO	Req. — 8K RAM/Cassette	9.95
		— 16K RAM/Disk	14.95
	• LIGHTNING BOLTS and REACTION	Req. — 16K RAM/Cassette	9.95
		— 24K RAM/Disk	14.95
	• THE MAD MARBLE	Req. — 8K RAM/Cassette	9.95
		— 16K RAM/Disk	14.95
	• MUSIGAME (2 Games)	Req. — 16K RAM/Cassette	9.95
		— 24K RAM/Disk	14.95
	• SUPERMASTER	Req. — 8K RAM/Cassette	9.95
		— 16K RAM/Disk	14.95
	• TAG	Req. — 16K RAM/Cassette	9.95
		— 16K RAM/Disk	14.95
	• TRACTOR BEAM	Req. — 8K RAM/Cassette	9.95
		— 16K RAM/Disk	14.95
	• WAR AT SEA	Req. — 16K RAM/Cassette	14.95
		— 24K RAM/Disk	19.95
BUSINESS	• CCA Data Management System	Req. — 48K RAM/Disk	99.95
	• LETTER WRITER	Req. — 24K RAM/Disk	19.95

Mastercard & VISA Accepted

CE DIVISION OF CUSTOM ELECTRONICS, INC.
SOFTWARE

238 Exchange St., Chicopee, Massachusetts 01013
(413) 592-4761

• Dealer And Distributor Inquiries Invited
• Closed Mondays — Open Daily 'Til 5:30 — Fridays 'Til 8

ATARI* 400*& 800*
Computer Software

HORSEWORD DERBY™
†Player against the computer
†Action graphics
†Attractive colors and sounds
†Four levels of play
$19.95

AVOID "THE BEST PLAYER WINS"
1 or 2 Player ACTION SPACE GAME
†Great color, sound, graphic combination
†5 play variations
†Keep ship in quadrant while trying to blast everyone. (including opponent)
$19.95

TRIBOMB ACTION GAME
†1 or 2 player game...just try to beat the computer. ·5 levels of play
$19.95

*ATARI REGISTERED TRADEMARK OF ATARI, INC.
ALL ATARI 2600, 800 & 400 CARTRIDGES & SOFTWARE, HARDWARE, JOYSTICKS, PADDLES ETC. SEND FOR CATALOG.
†Requires 16K RAM ADD $1.50 SHIPPING

COMPUTER MAGIC LTD.
176 Main St. Port Washington,N.Y.,11050

Dealer Inquiries
516 883-0094

DRAC IS BACK
(by Ted Clawges)

An adventure game to test your logic and your tendency towards greed. Battle monsters, werewolves, vampires, speeding demons, the crafty Igor and of course the Lord of Evil...Drac. How much gold can you escape with ? How much will you spend to stay alive ? Your instincts could be wrong. (non-scrolling). © 1981

24K Atari® 800 T.M. cassette only $14.95

SYNCRO, INC. SOFTWARE DIVISION
31332 VIA COLINAS
SUITE 107
WESTLAKE VILLAGE, CALIF. 91362

VISA AND MASTERCARD ACCEPTED
CALIF. RESIDENTS PLEASE ADD 6% SALES TAX. $1.00 SHIPPING.

Using Strings For Graphics Storage

Michael Boom
Spokene, WA

If you've ever been frustrated attempting to PLOT and DRAWTO your way through a complex pattern or design in Atari Graphics, you might appreciate a method of graphics generation using text strings to store pixel data. While this string method is not simpler to use in all cases, its ease of data entry and manipulation possibilities make it a strong graphics tool.

Simple line drawings over large areas of the screen are best done using PLOT and DRAWTO commands, since this method uses less memory and generates images faster than the string method will. However, if you have a very complex pattern in a small area of the screen, the string method works well. The heart of string graphics lies in the fact that if you run a PRINT #6 statement followed by ASCII characters while in Graphics Modes 3-7, colored pixels will appear on the screen. Different letters and symbols will plot different colors, but for our purpose we will deal only with the letters A, B, C, and D. Each of these letters plots a different colored pixel in Graphics modes 3, 5, and 7:

A plots color 1 (color register #0)
B plots color 2 (color register #1
C plots color 3 (color register #2)
D plots color 0 (color register #4)

In Graphics modes 4 and 6, only the letters A and B need be used, A for the plotting color, B for the background color.

For a demonstration, if you type the command
GRAPHICS 3: PRINT #6; "ABCDA"
moves the pixel string down and to the right.

Creating A Graphics String:

We can now use the above methods to plot a pattern. First graph out the area needed for the pattern, then fill in the pattern using "A", "B", "C", and "D" to represent the colors wanted:

String 1	**CDDDDAAAAA**
String 2	**DCDDDDDDAA**
String 3	**DDCDDDDADA**
String 4	**DDDCDDADDA**
String 5	**DDDDCADDDA**
String 6	**AAAAACDDDD**
String 7	**ABBBADCDDD**
String 8	**ABCBADDCDD**
String 9	**ABBBADDDCD**
String 10	**AAAAACCCCC**

Now break down the graph as a series of strings, in this case 10 strings of 10 characters each:

String 1 is "CDDDDAAAAA"
String 2 in "DCDDDDDDAA"
etc.

Concatenate the 10 strings for more efficient data storage:

"CDDDDAAAAADCDDDDDDAADDCDDDDADADD DCDDADDADDDDCADDDAAAAAACDDDDABB BADCDDDABCBADDCDDABBBADDDCDAAAA ACCCCC"

We have now generated all the data necessary to plot our figure (a square with an arrow) in the graphics mode, and have stored it in one long string

Display

To plot the string on the screen, determine where you would like the upper left hand corner of the figure to be located, and enter it during the run of the following program after prompt "X,Y?"

```
10  GRAPHICS 5
20  DIM A$(100)
30  $="CDDDDAAAAADCDDDDDDAADDCDDDD
      ADADDDCDDADDADDDDCADDDAAAAAA
      CDDCDDDABCBADDCDDABBBADDDCDAA
      AAACCCCC"
40  PRINT "X,Y";:INPUT X,Y
80  FOR K= 1 TO 10
90  POSITION X,Y+K -1
100 PRINT #6; A$(K*10-9,K*10)
110 NEXT K
```

In this program, lines 20 and 30 set up our main pixel data string and line 40 establishes the upper left corner coordinates of the figure. Lines 80 and 110 set up a loop of 10 steps, to divide our main data string into 7 rows. Line 90 positions the cursor for each row, and line 100 prints 10 consecutive 10 character strings on the screen.

Obviously there are figures which require strings too long for direct entry in Atari Basic. In that case, divide the figure into several rectangular sections, each small enough for inclusion into one string (usually under 100 characters in length.) Then concatenate the string as explained in the Basic Reference Manual, p. 39.

Figure Manipulation:

Plotting a figure using strinng graphics is fairly simple and straightforward. Its real strength lies in figure manipulation through string reading. Some easy manipulations are:

1. **Figure rotation (in 90° implements)**
2. **Figure inversion**
3. **Color changes**

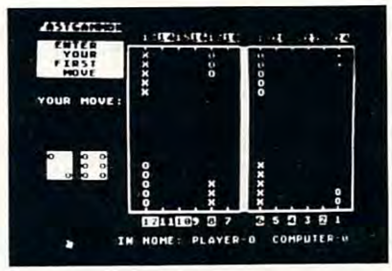
For figure rotation, using the same example figure and data string, let's substitute and add to the previous program. For a 90 degree turn clockwise, add and substitute:

SOFTWARE FOR THE ATARI 800* AND THE ATARI 400*

TARI TREK™
By Fabio Ehrengruber

Get ready for an exciting trek through space. Your mission is to rid the galaxy of Klingon warships, and to accomplish this you must use strategy to guide the starship Enterprise around stars, through space storms, and amidst enemy fire. Sound and color enliven this action-packed version of the traditional trek game. Nine levels of play allow the player to make the mission as easy or as challenging as he wishes. At the highest level you are also playing against time. Damage to your ship can be repaired in space at a cost of time and resources if you can't make it back to base. TARI TREK gives you a lot of trek at a low price. This program is written entirely in BASIC and requires at least 24K of user memory. For the Atari 800 only.

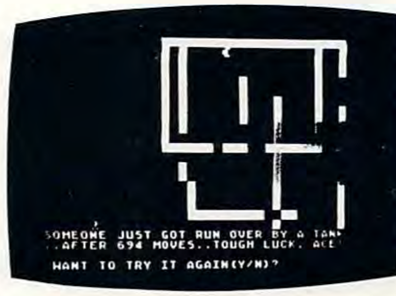
Cassette - $11.95 Diskette - $14.95



FASTGAMMON™
By Bob Christiansen

Play backgammon against a talented computer opponent. This is the latest and best version of the most popular backgammon-playing program for personal computers - FASTGAMMON. Roll your own dice or let the computer roll them for you. Adjust the display speed to be fast or slow. If you wish you can play a game using the same dice rolls as the previous game - a great aid in improving your skills at backgammon. Beginners find it easy to learn backgammon by playing against the computer, and even very good players find it a challenge to beat FASTGAMMON. The 12-page instruction booklet includes the rules of the game. Written in machine language. Requires only 8K of RAM and runs on both the Atari 400 and the Atari 800.

On cassette only - $19.95



TANK TRAP
By Don Ursem

A rampaging tank tries to run you down. You are a combat engineer, building concrete barriers in an effort to contain the tank. Use either the keyboard or an Atari joystick to move your man and build walls. If you trap the tank you will be awarded a rank based on the amount of time and concrete you used up. But they'll be playing taps for you if you get run over. There are four levels of play. Higher levels of play introduce slow curing concrete, citizens to protect, and the ability of the tank to shoot through any wall unless you stay close by. Music, color, and sound effects add to the excitement. Written in BASIC with machine language subroutines. Requires at least 16K of user memory. Runs on the Atari 800 and on an Atari 400 with 16K RAM.

Cassette - $11.95 Diskette - $14.95

QS FORTH™ By James Albanese. Step into the world of the remarkable FORTH programming language. Writing programs in FORTH is much easier than writing them in assembly language, yet FORTH programs run almost as fast as machine code and many times faster than BASIC programs. QS FORTH is based on fig-FORTH, the popular model from the FORTH Interest Group that has become a standard for microcomputers. QS FORTH is a disk-based system that can be used with up to four disk drives. There are five modules included:

1. The FORTH KERNEL (The standard fig-FORTH model customized to run on the Atari computer).
2. An EXTENSION to the basic vocabulary that contains some handy additional words.
3. An EDITOR that allows editing source programs (screens) using Atari type editing.
4. An IOCB module that makes I/O operations easy to set up.
5. An ASSEMBLER that allows defining FORTH words as a series of 6502 assembly language instructions.

Modules 2-5 may not have to be loaded with the user's application program, allowing for some efficiencies in program overhead. Full error statements (not just numerical codes) are printed out, including most disk error statements. QS FORTH requires at least 24K of RAM and at least one disk drive. For the Atari 800 only.

On diskette only - $79.95

★ ★ ★ ★ ★ ★ ★ ★ ★

ASSEMBLER by Gary Shannon. Write your own 6502 machine language programs with this inexpensive in-RAM editor/assembler. Use the editor to create and edit your assembler source code. Then use the assembler to translate the source code into machine language instructions and store the code in memory. Simple commands allow you to save and load the source code to and from cassette tape. You can also save any part of memory on tape and load it back into RAM at the same or at a different location. The assembler handles all 6502 mnemonics plus 12 pseudo-ops that include video and printer control. Commenting is allowed and error checking is performed. A very useful feature allows you to view and modify hexadecimal code anywhere in memory. Instructions on how to interface machine language subroutines to your BASIC programs are included. ASSEMBLER requires 16K of user memory and runs on both the Atari 800 and the Atari 400.

On cassette only - $24.95

★ ★ ★ ★ ★ ★ ★ ★ ★

6502 DISASSEMBLER by Bob Pierce. This neat 8K BASIC program allows you to disassemble machine code, translating it and listing it in assembly language format on the video and on the printer if you have one. 6502 DISASSEMBLER can be used to disassemble the operating system ROM, the BASIC cartridge, and machine language programs located anywhere in RAM except where the DISASSEMBLER itself resides. (Most Atari cartridges are protected and cannot be disassembled using this disassembler.) Also works as an ASCII interpreter, translating machine code into ASCII characters. 6502 DISASSEMBLER requires only 8K of user memory and runs on both the Atari 800 and the Atari 400.

Cassette - $11.95 Diskette - $14.95

QS QUALITY SOFTWARE
6660 Reseda Blvd., Suite 105, Reseda, CA 91335
(213) 344-6599

WHERE TO GET IT: Call us at (213) 344-6599 for the name of the Quality Software dealer nearest you. If necessary you may order directly from us. Mastercard and Visa cardholders may place orders by telephone. Or mail your check or bankcard number to Quality Software, 6660 Reseda Blvd., Suite 105, Reseda, CA 91335. California residents add 6% sales tax. SHIPPING CHARGES: Within North America orders must include $1.50 for first class shipping and handling. Outside North America the charge for airmail shipping and handling is $5.00. Pay in U.S. currency.

*Indicates trademarks of Atari, Inc.

```
 20 DIM A$(100),B$(100)
 50 FOR K = 1 TO 10: FOR L = 1 TO 10
 60 B$(K*10-10+L,K*10-10+L)=A$((10-L)*10+K,
    (10-L)*10+K)
 70 NEXT L, NEXT K
100 PRINT #6;B$(K*10-9,K*10)
```

For a 270 degree clockwise rotation, substitute:

```
60 B$(K*10-10+L,K*10-10+1)+A$(L*10+1-K,L*
  10+1-K
```

For a 180 degree clockwise rotation, substitute

```
50 FOR K = 1 TO 100
60 B$(K,K)=A$(101-K,101-K)
70 NEXT K
```

To change color assignments, add and substitute to the *original* program:

```
50 FOR K=1 to 100
60 IF A$(K,K)="C" THEN A$(K,K)="A"
70 NEXT K
```

To invert a figure, substitute to the original program:

```
100 PRINT #6; A$((11-K)*10-9,(11-K)*10)
```

To turn a figure left to right, substitute in the 180 degree rotation program:

```
100 PRINT #6; B$((11-K*10-9,(11-K)*10)
```

The string manipulations used to manipulate this 10x10 figure can easily be incorporated into subroutines for use in programs using repetitive figures in different positions. Further experimentation for more possibilities is definitely in order.

I hope that the method of string graphics is handy and useful for those of you interested in Atari graphics. Good luck with them. ©

Atari Machine I/O

Charles Brannon

There are three routines that will be of interest to ATARI machine language programmers.

Location $F6E2 waits for a key to be pressed, and will return its ASCII value in the accumulator. (Works like GET# in BASIC)

Location $F6A4 puts the character in the accumulator on the screen in the next print location. (Works like PUT#6) The X and Y registers are altered by this routine.

The INPUT routine at $F63E is a little trickier. It will input a line from the screen and keyboard, just like the INPUT statement does in BASIC. It does not store the line anywhere, however. To use it, do a JSR $F63E to get each character of the line. The character will be returned in the accumulator. Check for end of input by comparing the value to 155, the ATASCII value of the RETURN key. You must store the values in memory to save the input. Since the X and Y registers are altered by this routine, you have to save them if you are using them before you call the routine. The program at the end of this article demonstrates this.

Quick Reference

```
GETCHAR   $F6E2
OUTCHAR   $F6A4
INPUT     $F63E
```

Finally, I warn you that although these addresses work on my ATARI, they might be different on yours.

```
INPUT   LDX #0          ;initialize loop counter
NEXT    STX SAVEX       ;save it
        JSR  $F63E      ;get a character
        LDX SAVEX       ;restore index
        STA STRING,X    ;save character
        INX             ;increment counter
        CMP #9B         ;is accumulator = 155
                          (RETURN)?
        BNE NEXT        ;if not, continue
        RTS             ;Finished
```

```
OUTPUT  LDX #0          ;initialize loop counter
NXT     STX SAVEX       ;save it
        LDA STRING,X    ;fetch a character from memory
        JSR  $F6A4      ;print it
        LDX SAVEX       ;restore index
        INX             ;increment it
        CMP #$9B        ;accumulator = 155 (RETURN)?
        BNE NXT         ;if not, continue
        RTS             ;Finished
```

©

SOFTWARE FOR ATARI

SWIFTY SOFTWARE, INC.

HOME MANAGEMENT

FILEIT by Jerry White. A Database System. Use this no-nonsense electronic cataloguing system to create, add, delete, edit, print, store and sort different kinds of custom files. For every need around your home. Disk Package runs on 24K and up. **$34.95** disk only.

PROGRAMMING AIDS

BASIC RENUMBER by Fernando Herrera. Change lines, references, even variables!! Includes **BCD** converter and BASIC PROGRAM DECODER. **$14.95** cassette; **$19.95** disk.

CHALLENGE & FUN

SPACE CHASE by Fernando Herrera. Conquer the Galaxy ... if you can escape the deadly chase of the patrol fleet. 4 levels.
TIME BOMB by Fernando Herrera. Beat the clock in a Mine Field. 5 levels — any number of players — scoreboard lists top players.
MINDTRACE by Lee Jacknow. If you like Simon you will love this challenge ... a real test for your memory.
DOTS-N-BOXES by Lee Jacknow. You played it as a kid ... it's better on a computer.

Games on cassette; **$14.95** each
Add $5 to total if requesting disk

EDUCATIONAL

MY FIRST ALPHABET by Fernando Herrera. Before you send your kid to school he or she will know all the letters, numbers, musical tunes, animals ... even typing and how to run a program!! Program allows parents to use built-in drawings or to design their own. **$29.95** disk only.

TO ORDER SEND CHECK OR MONEY ORDER TO:

SWIFTY SOFTWARE, INC.
P.O. BOX 641
MELVILLE, N.Y. 11747
(516) 549-9141

Atari is a trademark of Atari, Inc. · N.Y. RESIDENTS ADD 7% SALES TAX

ADVENTURE
for your
ATARI 400/800

Vol. #1 THE QUEST

... and now, after the final battle, the remnants of humankind manage to survive in small bands scattered around the globe. Chaos and savagery reign supreme on the devastated planet. Can you discover the awesome secret that will save Earths dwindling populace from a fate worse than death? As you progress thru the more than 60 locations in this adventure, you encounter obstacles and aids, loathsome beasts and helpful strangers. But beware, one false move could mean doom both for you and all mankind. THE QUEST is the first volume in a larger, multi-part adventure. Completion of the entire adventure will require purchase of added volumes. Graphics and sound enhance this classic game.

16K/Tape-$14.95 24K/Disk-$19.95

SURVIVAL SOFTWARE
3033 LA SELVA, #B306
SAN MATEO, CA. 94403

ATARI SOFTWARE by COMPUGRAPHICS

LASER BALL-Action game of zap-or-be-zapped!
PILLBOX-Destroy the enemy with mortar missles!
SPACE BASE-Protect your space station from aliens!

All require 8K, 2 joysticks
On Cassette Only $5.95 Each !!!
(AZ Residents add 5% tax)
SEND CASH, CHECK OR M.O. TO:
COMPUGRAPHICS
13850 N. 19th Ave Suite #220
Phoenix, AZ 85023 (602)-863-1096
Atari is a trademark of ATARI, Inc.

Illusions II

P.O. BOX 16489
IRVINE, CA 92713

WE ARE DIFFERENT!

80% of the games available for Atari systems are pure garbage! The other 20%, the good ones, charge a fortune.

We create superior, total graphic games at the right price. We'll also save you money on hardware.

Try us one time with this guarantee-

Send $15 for our Games-1 disk (16 K). It has two *thinking* games, CHEENG'S RISERS and CLOSE 'N COUNTERS. Both are one player games. CHEENG has 10 levels of play, CLOSE has 6 levels. Your children will love them, you'll become addicted to them.

IF YOU DON'T THINK THE DISK IS WORTH EVERY PENNY, SEND IT BACK AND WE'LL REFUND EVERY PENNY - IMMEDIATELY!

****

EPSON MX-80 PRINTER........$497

****

Calif residents - add 6% sales tax

8K BASIC ATARI™ CASSETTE PROGRAMS

THREE-BASE CALCULATOR - $12.95
Hexadecimal, Decimal, Octal. Reverse Polish Notation.

AUDIBLE DISASSEMBLER - $9.95
Op-code generates a tone. Decimal Hexadecimal.

DECISION MAKER - $8.95
Aids in evaluation of complex decisions.

Color Computer Concepts

ATARI is a trademark of ATARI, INC.

P.O. BOX 1206
KENT, WA 98031

Disk Directory Printer

Len Lindsay

If you have an Atari disk, you know that you can see its directory by entering DOS and choosing option A. Well, here is a program I wrote completely in ATARI BASIC that will give you the same directory listing. Then a second program is listed that will give you an "expanded" directory.

The key to this program is being able to open the directory as a file for a READ. This is easily accomplished with the following statement.

100 OPEN #1,6,0,"D1:*.*"

Next you must know how the file name info is stored in the directory. The file info is stored as a string 17 characters long.

The first character tells if the file is locked or not. If it is "*" then it is locked. If it is " " (space) then it is not locked.

The file name comes next. Characters 3-10 are the file name. Characters 11-13 are the extension for the name. Any unused characters are stored as spaces. Note, however, that you can't imbed the spaces in your name when you access the file.

Characters 15-17 are the number of sectors used by the program.

With that info you can see how the second, expanded directory list, works. You now can read the directory within your programs by following the new simple methods shown.

Listing 1

```
0 REM PRINT DIRECTORY
1 REM *** (C) 1981
2 REM *** LEN LINDSAY
3 REM ***
4 REM *** SAME AS VIA DOS
10 GRAPHICS 0
20 DIM FILENAME$(20)
100 OPEN #1,6,0,"D1:*.*":REM OPEN DIRECT
ORY FOR A READ
110 TRAP 900:REM NO MORE FILES
200 INPUT #1;FILENAME$
300 PRINT FILENAME$
400 PRINT LEN(FILENAME$)
800 GOTO 200
900 END
```

Listing 2

```
0 REM PRINT DIRECTORY
1 REM *** (C) 1981
2 REM *** LEN LINDSAY
3 REM ***
4 REM *** EXPANDED DIRECTORY PRINT
10 GRAPHICS 0
20 DIM FILENAME$(20)
100 OPEN #1,6,0,"D1:*.*":REM OPEN DIRECT
ORY FOR A READ
110 TRAP 900:REM NO MORE FILES
200 INPUT #1;FILENAME$
300 IF LEN(FILENAME$)<5 THEN 900
400 PRINT "FILENAME IS: ";
410 FOR LOOP=3 TO 13
420 IF LOOP=11 THEN PRINT ".";
430 IF FILENAME$(LOOP,LOOP)<>" " THEN PR
INT FILENAME$(LOOP,LOOP);
440 NEXT LOOP
450 IF FILENAME$(1,1)="*" THEN PRINT "
 LOCKED ";
460 PRINT
500 PRINT "      ";FILENAME$(15,17);" SEC
TORS"
800 GOTO 200
900 PRINT " SECTORS FREE =";FILENAME$
```

©

Condensing Data Statements On The Atari

Craig Patchett

This article was originally written as an appendum to my article "Designing Your Own Atari Character Sets" (see the March 1981 issue of **COMPUTE!**). It then occured to me, however, that there are most likely many other applications where this simple technique might be useful, especially in the loading of machine language subroutines from BASIC DATA statements. In general, any program where a significant amount of numbers between 0 and 255 must be stored as data can be reduced in size using the technique.

An Atari memory location, as is true with most microcomputers, can only hold numbers in the range of 0 to 255. Not by coincidence, 0 to 255 is also the range of ATASCII values, each of which can be translated to an Atari character using the CHR$ function. On the same note, each Atari character can be translated to its ATASCII value using the ASC function. This means that one character can be used in place of from one to three digits. Since characters can be combined in character strings, one character can replace up to three digits *and* a comma when used in place of its corresponding value in DATA statements. Therefore, in programs that use a lot of numerical data in the 0 to 255 range, character strings can be utilized in the following way to cut down the program's memory requirements:

```
30000 REM /*Make sure we're not at the e
nd of the current string*/
30010 IF ME=LEN(DAT$) THEN ME=0:READ DAT
$
30020 REM /*Increment ME (pointer into D
AT$)*/
30030 ME=ME+1
30040 REM /*Convert next character to it
's ATASCII value*/
30050 VALUE=ASC(DAT$(ME,ME))
30060 REM /*All done*/
30070 RETURN
```

To use this subroutine, first DIMension DAT$ to the length of the longest data string you plan to use, and initialize ME to 0. Then, each time you would normally use a READ command, use a GOSUB 30000 instead and the data value will be returned in VALUE. Of course, you must first convert your data to the appropriate Atari characters. Appendix C: ATASCII Character Set, in the BASIC Reference Manual, can be used to aid in this task. Keep in mind that, for the most part, ATASCII values 128-255 are just the reverse of values 0-127 (in other words, use the reverse character key). The ‹ESC› key, in combination with other keys, can often be used to get the more evasive characters. To make life a little easier for you, I've included this short program that will print out the ATASCII values of any characters typed while it is running. Good luck!

```
10 OPEN #1,4,0,"K:"
20 GET #1,VALUE
30 PRINT VALUE
40 GOTO 20
```

(Note: to get the ATASCII value of a character such as ‹ECS›‹CTRL›+ using this program, just type ‹CTRL›+. Pressing the ‹ESC› key will give you the value of the ‹ESC›‹ESC› character.) ©

Real-Time Clock On The Atari

Richard Bills
Lisle, IL

As the popularity of the Atari Computer grows, more people are realizing that it offers more capabilities than other computers in the same price range. Many of its capabilities, however, are not advertised. For instance, I would not have known that it had real-time clock hardware if my dealer had not told me about it. I have since developed this flexible 3K program to utilize this hardware.

The program will first ask you if you want to set the alarm time. If you do, it will ask you to give the time in twenty-four hour format (for example, 15,30,20). Otherwise it will disable the alarm. Next, it will ask you if you would like to set the time. If you do, it will ask for the hours, minutes, and seconds, and will enter this time into the hardware registers. You may use twenty-four hour time if you wish. If you don't want to set the time, the time presently in the hardware registers will not be changed. After fulfilling these preliminaries, the clock time is then displayed in the center of the screen. The time is stored and kept in the hardware and should not be disturbed unless you hit SYSTEM RESET. You may have noticed that this program uses large line numbers (near the 32,767 limit). This enables you to attach this program to the end of another program (or several programs) as a subroutine. I suggest using LIST "C" to save the program and ENTER "C" to load the program. These commands allow you to enter the program without erasing the program that resides in memory. LIST "C", X,Y will list lines X through Y to the cassette, enabling you to save a certain routine without including the clock program. A line by line description of this program follows.

30000 Let's clear the screen and shut off the cursor.

30001–30002 OFF is a flag. When it equals 1 then the alarm will not go off.

30003 TOTAT is the total alarm time in sixtieths of a second

30010–30016 These lines input the time which is to be placed in the hardware registers.

30020–30021 Register 53279 is the register which indicates which console button(s) are pressed. It equals 6 when START is pressed.

30025 This line POKE's the clock hardware down to 0. The largest number a register can have is 255. Register 20 increments by 1 every sixtieth of a second and increments register 19 by 1 when it counts beyond 255 (back to 0 again). Register 18 increments by 1 when register 19 counts beyond 255.

30030–30049 Now we break the current time down into sixtieths of a second and store them in the hardware registers.

30100 This collects the time from the hardware registers in sixtieths of a second.

30150 If the time in the registers is greater than 24 hours, lines 31000–

31070 will be executed. They bring the time in the registers down to an equivalent time below the 24 hour level. This allows the time to continue to be kept in the hardware for an indefinite period of time by preventing all the registers from counting beyond the 255 level and going to 0 at the same time; this would cause the time to be lost.

30522 This line can be eliminated if 24 hour time is preferred.

30523 The time is obtained from the registers in order to compare it at line 30526 to the time the alarm was set to go off at. Since the program is too slow to be able to check the alarm time continuously, a tolerance (100) may be changed.

30524 This line may also be eliminated if 24 hour time is preferred.

30530 This is the line which produces the alarm sound. Use your imagination here!

30539–30700 The printing of the time is performed by these lines. They insure that the zeros will be correctly placed and that the length of the line will always be the same.

```
30000 PRINT "}":DIM X$(10):POKE 752,1
30001 ? "Do you want to initialize the a
larm";:INPUT X$:IF X$="YES" OR X$="yes"
THEN OFF=0:GOTO 30003
30002 OFF=1:GOTO 30004
30003 PRINT "Set alarm time [use 24 hour
 time in 0,0,0 format]":INPUT AH,AM,AS:T
OTAT=AH*60*60*60+AM*60*60+AS*60
30004 ? "Do you want to set the time";:I
NPUT X$
30005 IF X$="YES" OR X$="yes" THEN 30007

30006 GOTO 30099
30007 ? "}"
30010 PRINT "Hours";:INPUT H
30015 PRINT "Minutes";:INPUT M
30016 PRINT "Seconds";:INPUT S
30020 PRINT "Hit START to begin the time
."
30021 IF PEEK(53279)<>6 THEN 30021
30022 PRINT "}"
30023 REM ******* PUT CURRENT TIME IN HA
RDWARE REGISTERS*******
30025 POKE 18,0:POKE 19,0:POKE 20,0
30030 T=H*60^3+M*60^2+S*60
30040 POKE 18,INT(T/(256*256))
30043 T=T-(256*256)*(INT(T/(256*256)))
30045 POKE 19,INT(T/256)
30047 T=T-256*(INT(T/256))
30049 POKE 20,INT(T)
30099 ? "}"
30100 TIME=PEEK(20)+PEEK(19)*256+PEEK(18
)*256*256
30150 IF TIME>=5184000 THEN 31000
30200 TIME=INT(TIME/60+0.5)
30300 SEC=TIME-60*(INT(TIME/60))
```

```
30350 TIME=INT((TIME-SEC)/60)
30400 MIN=TIME-60*(INT(TIME/60))
30500 HOURS=INT((TIME-MIN)/60)
30505 IF SEC>=60 THEN 30510
30508 GOTO 30515
30510 MIN=INT(SEC/60)+MIN
30511 SEC=SEC-60*(INT(SEC/60))
30515 IF MIN>=60 THEN 30520
30518 GOTO 30522
30520 HOURS=INT(MIN/60)+HOURS
30521 MIN=MIN-60*(INT(MIN/60))
30522 IF HOURS=0 THEN HOURS=12
30523 ATCHECK=PEEK(18)*256*256+PEEK(19)*
256+PEEK(20)
30524 IF HOURS>12 THEN HOURS=HOURS-12
30525 SOUND 0,0,0,0
30526 IF ABS(ATCHECK-TOTAT)<100 AND OFF=
0 THEN 30530
30527 GOTO 30539
30530 ? " ":SOUND 0,50,10,10:FOR X=0 TO
1000:NEXT X:? "}}}}}}}"
30539 POSITION 15,10
30540 IF HOURS<10 THEN 30550
30542 IF MIN<10 THEN 30630
30544 IF SEC<10 THEN 30700
30545 PRINT INT(HOURS+0.5);":";INT(MIN+0
.5);":";INT(SEC+0.5):GOTO 30100
30550 IF MIN<10 THEN 30560
30551 GOTO 30600
30560 IF SEC<10 THEN PRINT "0",INT(HOURS
+0.5);":0";INT(MIN+0.5);":0";INT(SEC+0.5
):GOTO 30100
30561 PRINT "0";INT(HOURS+0.5);":0";INT(
MIN+0.5);":";INT(SEC+0.5):GOTO 30100
30600 IF SEC<10 THEN PRINT "0";INT(HOURS
+0.5);":";INT(MIN+0.5),":0";INT(SEC+0.5)
:GOTO 30100
30601 PRINT "0",INT(HOURS+0.5);":";INT(M
IN+0.5);":";INT(SEC+0.5):GOTO 30100
30630 IF SEC<10 THEN PRINT INT(HOURS+0.5
);":0";INT(MIN+0.5),":0";INT(SEC+0.5):GO
TO 30100
30631 PRINT INT(HOURS+0.5);":0",INT(MIN+
0.5);":";INT(SEC+0.5):GOTO 30100
30700 PRINT INT(HOURS+0.5);":";INT(MIN+0
.5);":0";INT(SEC+0.5):GOTO 30100
30900 REM The next lines will poke the h
ardware clock registers down 24 hours
31000 TIME=PEEK(18)*256*256+PEEK(19)*256
+PEEK(20)
31005 TIME=TIME-5184000*(INT(TIME/518400
0))
31020 POKE 18,INT(TIME/(256*256))
31030 TIME=TIME-(256*256)*INT(TIME/(256*
256))
31040 POKE 19,INT(TIME/256)
31050 TIME=TIME-256*(INT(TIME/256))
31060 POKE 20,INT(TIME)
31070 GOTO 30100
```

©

Review Stud Poker

Robert W. Baker
Atco, NJ

STUD POKER is an interesting card game program for the 16K Atari from Dynacomp, Inc., 6 Rippingale Road, Pittsford, NY 14534. ($11.95, cassette; $15.95, diskette) The program includes two separate menu selectable versions of familiar stud poker, each with simple graphics and some sound effects. The card displays are simply the card outline with the face value and suit, no fancy card displays are used. For sound, you get to hear the cards shuffled and dealt along with other appropriate "bells and whistles" at important times.

One of the games deals two cards to you and the Atari, with one card down for the Atari. You each bet on your hands, and bet again after each of the remaining three cards are dealt. At each betting interval you can call, bet/raise from $1 to $3, or fold. The current pot value and your current winnings or loses are always displayed. When the hand is over, the Atari's down card is turned over and the winner is declared.

The other game is even simpler, both you and the Atari are each dealt five cards. Two of the Atari's cards are face down and not displayed. You must bet on your hand ($10 to $100) and cannot fold. After betting, the Atari's down cards are turned over and the winner is declared. Again, your total winnings or loses are displayed.

The games are rather interesting and it would appear that the Atari's card playing skills are pretty good. However, the documentation supplied was rather confusing and did not match the program operation. The names of the two games as well as the betting limits were different in the manual from that used in the program. Also, a different method of indicating whether to continue or quit was used by each part of the program after each hand. One section wanted a "C" or "Q" while the other wanted a RETURN with a null or "Q" input. Totally confusing! With a little more consistency and clearer documentation this could be a very nice package. ©

THE OSI® GAZETTE

Through The Fill-The-Buffer Routine

With Gun And Camera

Kerry Lourash
Decatur, Illinois

This is an effort to shed some light on the Fill-the-Buffer routine (FTB) of OSI BASIC-in-ROM. Subroutines with FFXX addresses are for the C1P, but should be about the same for the C2P. Let me warn you - all numbers in this article are hexidecimal, unless stated otherwise! I will appreciate any corrections or additions readers may have.

What Is It?

The buffer mentioned is a section of zero-page memory (locations 13-5A). When you type in a line of BASIC or the tape recorder loads your favorite program the computer stores one BASIC line at a time in the buffer. Since the buffer is only 72 (decimal) bytes long, no BASIC line can be longer than 72 (dec.) characters. By the way, when you type a 4-digit line number, you have only 68 (dec.) characters left in the line. The FTB takes input from the keyboard or ACIA (Asynchronous Communication Interface Adapter), depending on the status of the SAVE and LOAD flags. After the line is stored in the buffer, other routines tokenize the line and store it in the BASIC workspace.

What Does It Do?

This is what the FTB does:

1. Filters input so no graphics or control characters except "BEL" (end of line) and NULL (zero) gets into the buffer.
2. Checks the "CTRL 0" (output) flag (loc. 64) to see if characters should be output to TV and ACIA.
3. Counts the number of characters input and gives an automatic carriage return/line feed (CR/LF) if the line length stored in loc. 0F is exceeded.
4. Outputs ten NULLS after a CR, and an additional number of NULLS equal to that stored in loc. 0D after a LF.
5. Implements control characters such as carriage return (0D), line feed (0A), "BEL" (07), backspace (5F), and line delete (@,40).
6. Puts a NULL in the buffer at the end of a line to mark the end of line for following routines. Sets the X and Y registers to the start of the buffer(-1).

Preparing For Our Journey

Machine language routines are murder to decipher, and the FTB is no exception. The code is compact in order to stuff BASIC into 8K of ROM, and uses nested subroutines extensively. In my chart, I've put the subs immediately after the point where they are called, instead of in numerical order. Also, subs are indented and bracketed, so the addresses at the far left are the main routine and the subs are at the right, in brackets. The format is somewhat like the outlines we did in school. I've tried to make the routine understandable to both machine language and BASIC oriented readers. The ML addresses have been kept so any part of the routine can be pinpointed and disassembled for additional info; BASIC readers can consider the addresses as line numbers. Most assembly language has been replaced by explanations of what is happening. I have used only a few mnemonics and have given their BASIC equivalents in the heading of the chart.

Into The Jungle

Now we're thru the preliminaries, on with the safari! Look for line A357 on the chart; this is our starting point. First, the X register is zeroed. The x-reg. counts characters as they are input into the buffer. Through a series of JSR's(JSR = GOSUB) and JMP(GOTO) thru RAM, we come to the input sub at FFBA. For those who have the Aardvark cursor program, this is where it steps in and does its stuff. Locations 218 and 219 are changed so that BASIC jumps to the Aardvark program instead of FFBA.

The Input Trek

The input sub looks at loc. 203, the LOAD flag. If the MSB (Most Significant Bit) of 203 is zero, the sub goes to FD00, the keyboard scan sub, which waits for an input from the keyboard, decodes it, puts it in the A register, and returns (RTS) to A389. On the other hand, if the MSB of loc. 203 is 1, the sub checks the LSB (Least Significant Bit) of F000, the ACIA's status register, and waits 'til it is zero, which means the ACIA has a byte ready in F001. This byte is stored in the A-reg. and the routine returns to A389, just like the keyboard routine does. Oh yes, one thing I forgot to mention: before F000 is

OSI **SOFTWARE FOR OSI** OSI

A JOURNAL FOR OSI USERS!!

The Aardvark Journal is a bimonthly tutorial for OSI users. It features programs customized for OSI and has run articles like these:

1) Using String Variables.

2) High Speed Basic On An OSI.

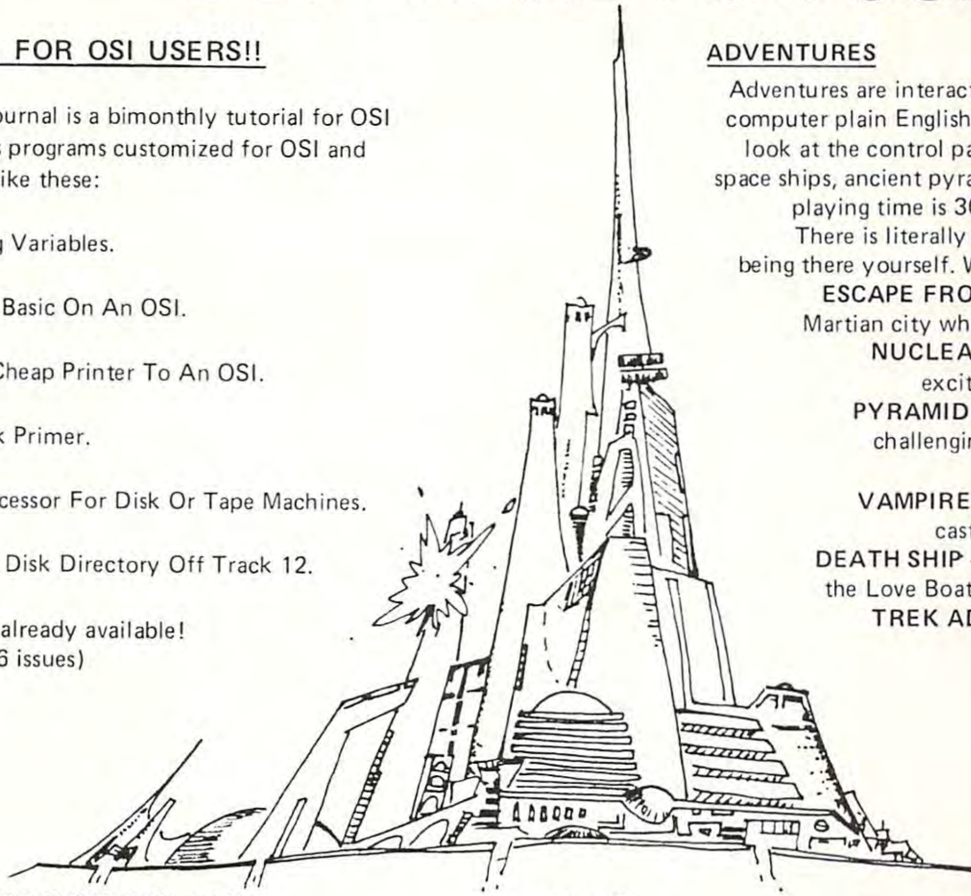
3) Hooking a Cheap Printer To An OSI.

4) An OSI Disk Primer.

5) A Word Processor For Disk Or Tape Machines.

6) Moving The Disk Directory Off Track 12.

First year issues already available!
$9.00 per year (6 issues)



ADVENTURES

Adventures are interactive fantasies where you give the computer plain English commands (i.e. take the sword, look at the control panel.) as you explore alien cities, space ships, ancient pyramids and sunken subs. Average playing time is 30 to 40 hours in several sessions. There is literally nothing else like them – except being there yourself. We have six adventures available.

ESCAPE FROM MARS – Explore an ancient Martian city while you prepare for your escape.

NUCLEAR SUBMARINE – Fast moving excitement at the bottom of the sea.

PYRAMID – Our most advanced and most challenging adventure. Takes place in our own special ancient pyramid.

VAMPIRE CASTLE – A day in old Drac's castle. But it's getting dark outside.

DEATH SHIP – It's a cruise ship – but it ain't the Love Boat and survival is far from certain.

TREK ADVENTURE – Takes place on a familiar starship. Almost as good as being there.

NEW SUPPORT ROMS FOR BASIC IN ROM MACHINES

C1S – for the C1P only, this ROM adds full screen edit functions (insert, delete, change characters in a basic line). Softwave selectable scroll windows, two instant screen clears (scroll window only and full screen), software chose of OSI or standard keyboard format, Bell support, 600 Baud cassette support, and a few other features. It plugs in in place of the OSI ROM. NOTE: this ROM also supports video conversions for 24, 32, 48 or 64 characters per line. Replaces video swap tape on C1P model 2. All that and it sells for a measly $39.95.

C1E/C2E for C1/C2/C4/C8 Basic in ROM machines. This ROM adds full screen editing, software selectable scroll windows, keyboard correction (software selectable), and contains an extended machine code monitor. It has breakpoint utilities, machine code load and save, block memory move and hex dump utilities. A must for the machine code programmer replaces OSI support ROM. Requires installation of additional chip when installed in a C2 or C4. C1 installation requires only a jumper move. Specify system $59.95.

DISK UTILITIES

SUPER COPY – Single Disk Copier
This copy program makes multiple copies, copies track zero, and copies all the tracks that your memory can hold at one time – up to 12 tracks at a pass. It's almost as fast as dual disk copying. – $15.95

MAXIPROSS (WORD PROCESSOR) – 65D polled keyboard only - has global and line edit, right and left margin justification, imbedded margin commands, choice of single, double or triple spacing, file access capabilities and all the features of a major word processor – and it's only $39.95.

P.C. BOARDS

MEMORY BOARDS!! – for the C1P. – and they contain parallel ports!
Aardvarks new memory board supports 8K of 2114's and has provision for a PIA to give a parallel ports! It sells as a bare board for $29.95. When assembled, the board plugs into the expansion connector on the 600 board. Available now!

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

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

ARCADE AND VIDEO GAMES

GALAXIA one of the fastest and finest arcade games ever written for the OSI, this one features rows of evasive, hardhitting, dogfighting aliens thirsty for your blood. For those who loved (and tired of) Alien Invaders. – P.S. The price is a giveaway. SPECIFY SYSTEM!
Cassette $9.95 – Disk $12.95

TIME TREK (8K) – real time Startrek action. See your torpedoes move across the screen! Real graphics – no more scrolling displays. $9.95

INTERCEPTOR C1P ONLY! An all machine code program as fast and smooth as the arcades. You use your interceptor to protect your cities from hordes of enemy invaders. A pair of automatic cannons help out, but the action speeds up with each wave of incoming ships. The fastest and most exciting C1P game yet.
C1P Cassette $19.95

MINOS – A game with amazing 3D graphics. You see a maze from the top, the screen blanks, and then you are in the maze at ground level, finding your way through on foot. Realistic enough to cause claustrophobia. – $12.95

SINGLE STEPPER / MONITOR

This is probably the finest debugging tool for machine code ever offered for OSI systems. Its' trace function allows you to single step through a machine code program while it continuously displays the A, X, Y and status registers and the program and stack pointers. You can change any of the registers or pointers or any memory location at any time under program control. It takes well under 1k and can be relocated anywhere in free memory. It is a fine tool for all systems – and the best news of all is the extremely low price we put on it. – Tape $19.95 – Disk $24.95

FOR DISK SYSTEMS – (65D, polled keyboard and standard video only.)
SUPERDISK. Contains a basic text editor with functions similar to the above programs and also contains a renumberer, variable table maker, search and new BEXEC* programs. The BEXEC* provides a directory, create, delete, and change utilities on one track and is worth having by itself. – $24.95 on 5" disk - $26.95 on 8".

AARDVARK IS NOW AN OSI DEALER!

Now you can buy from people who can support your machine.

– THIS MONTH'S SPECIALS –

Superboard II	$279
C1P Model II	429
C4P	749
8K 610 board for C1P	269
Epson MX-80 printer with RS232 installed	595

. . . and we'll include a free Text Editor Tape with each machine!

True 32X32 Video Mod Plans for C1P
(4 Chips $3.00 Crystal Required)
$7.95

This is only a partial listing of what we have to offer. We now offer over 100 programs, data sheets, ROMS, and boards for OSI systems. Our $1.00 catalog lists it all and contains free program listings and programming hints to boot.

OSI **Aardvark Technical Services • 1690 Bolton • Walled Lake, MI 48088**
(313) 669-3110 OSI

checked, the keyboard is checked to see if the space bar has been hit. If so, the LOAD flag is turned off and we JMP to FD00 and then RTS to A389.

Now we have a byte, but we're not done processing it yet. At A389-A396 there is a section of code that tells the CPU to do nothing for a few microseconds. I'm not sure whether this is a time delay or just a spot where some code was deleted and the gap not closed up. Anyone know? After this lull, the MSB of the input byte is set to zero so we don't get any graphics characters and if the char. is a CTRL 0(0F) the output flag (loc. 64) is toggled. That means the output flag is changed to FF (all 1's) if it is zero, and vice versa. Finally, the input processing is completed and we RTS to the main routine at A35C.

Character Runs The Gauntlet

At A35C the character is tested to see if it is a "BEL". If it is, the X-register is checked to see if the buffer is full (more than 71 dec.). If there is room in the buffer, "BEL" is stored in the buffer and sent to the output sub A8E5 (more on this sub later). At A381 we are sent back to A359 to get another character. If the buffer is full, the "BEL" is output to the TV (or ACIA, if doing a SAVE) by A8E5, but "BEL" is not stored in the buffer. Now we are back at A359.

Let's temporarily bypass the test for carriage return (A360) and look at A364. This test blocks out control and graphics characters and sends us back to A359. That's why there's no way to stick a graphics char. directly into a line, even in a PRINT statement, without a CHR$ command. Look in your graphics manual and see what characters are legal (20-7D).

At A36C we test for @, the line erase character. We branch to A351 and JSR to A8E5 (outputs the @ character). Then a JSR to A86C, which sends a CR and a LF to A8E5, sending the cursor to "home". Now an RTS to A357 to zero the buffer counter, and we are back at A359, ready to start filling the buffer again. A370 tests for "SHIFT 0". Oddly enough, the ASCII of "SHIFT 0" happens to be 5F, which is also the cursor character. This time we branch back to A34B. A JSR to A8E5 outputs a cursor character. A34E decrements the buffer counter (X), and if we haven't erased backward beyond the start of the buffer, A34F sends us to ol' A359. If we have erased too far, a JSR to A86C homes the cursor, A357 zeroes the buffer counter, and we start filling the buffer at A359.

At A376 the buffer counter is checked. If the buffer is full, the input char. is changed to "BEL" (A37C) and output (A8E5) to tell you you're wasting your time. Nothing is stored in the buffer and we branch to A359 for another journey thru the FTB. Finally at A378, the character, if it has passed all the tests, is stored in the buffer. The contents of the buffer counter (X) are added to the number 13 (start of the buffer) and the character is stored at the resulting address. A37A increments the buffer counter, counter and A37E JSR's to A8E5, which prints the character.

The A8E5 Routine

Now for an explanation of the A8E5 sub. If the MSB of the output flag (loc. 64) is a 1, we RTS with no output to TV or ACIA.

If the MSB is zero, we check to see if the ASCII of the char. is less than 20 (BEL, CR, LF). If so, we skip the line length check and branch to A8FA. At A8FA we JSR to FFEE, which JMPs to the address found in 021A and 021B. This address is normally FF69, but you could cook up your own routine and put its starting address in 021A and 021B. From FF69, we JSR to BF2D, the video output sub, which I will explain in another article. To make a long story short, a "BEL" will be displayed as a graphics character, a CR will cause the cursor to be moved to the start of the line, and a LF will scroll the screen and "home" the cursor.

Now we RTS from the video sub and check the status of the SAVE flag (205). If 205 contains a zero, we RTS to A901. If the SAVE flag is non-zero the ACIA status register is monitored until its second bit is zero and then the character is sent to the ACIA (loc. F001). If the character is a CR then 10 (dec.) NULLs are also sent to the ACIA (this gives the computer time to process the line and scroll the screen when the program is LOADed from tape) and then we RTS to A901. A901 RTS's to A381 which brings us back to A359.

Back at A8EA, we assumed the input character would be less than 20. Let's see what happens if it's greater than 20. At A8EE addresses 0E and 0F are compared. 0E is the counter for the number of characters since the last CR. 0F contains the user-selectable line length (remember the "terminal width?" message at cold start?).

Don't confuse this line length with the maximum line length for the video stored at FFE1 or the cursor position counter at loc. 0200. If 0E and 0F are equal then there is a JSR to A86C. At A86C a CR and anLF are fed to the A8E5 sub for an automatic LF/CR. At A87A an additional number of NULLs equal to the number stored in loc. 0D are output. If 0E and 0F aren't equal there is a branch to A8F7 and 0E is incremented before the JSR to FFEE. The character is output to the TV and, if the SAVE flag is on, to the ACIA. Finally, we return to A359.

Last Leg of Our Journey!

Have patience, our journey is almost at an end. We skipped over the CR test at A360, now let's go through that one. If the input is a CR, a branch is made to A86C which puts a NULL at the end of the line in the buffer, marking the end of the line. This done, we are at A86C, which starts the auto CR/LF and the extra NULLs from loc. 0D. When we reach the end of the sub at A88A we RTS not to the FTB but to the Tokenize-the-Buffer routine, which is another story.

I highly recommend both Carlson's *OSI Basic In ROM* and William's and Dorner's *First Book of OSI*. The information in their books was invaluable in writing this article. I would like to hear from other people interested in Basic-in-ROM.

Fill-The-Buffer Routine (A357)

```
            JSR – GOSUB
            RTS – RETURN
     BRANCH,JMP – GOTO
            INC – ADD 1 (TO)
            DEC – SUBTRACT 1 (FROM)
        /02180/ – CONTENTS OF (LOC. 0218)
           CHAR – ASCII CHARACTER
            MSB – MOST SIGNIFICANT BIT
            LSB LEAST SIGNIFICANT BIT
 ALL NUMBERS IN HEX:
A34B JSR   A8E5
     A8E5  (SEE A8E5 BELOW)
     .....
     A901  RTS
A34E DEC   X-REG. (BUFFER COUNTER)
A34F IF X  GREATER THAN ZERO THEN A359
A351 JSR   A8E5
     A8E5  (SEE BELOW)
     .....
     A901  RTS
A354 JSR   A86C
     A86C  (SEE BELOW)
     .....
     A88A  RTS
A357 ZERO  X-REGISTER (BUFFER COUNTER = 0)
A359 JSR   A386
     A386  JSR FFEB
           FFEB JMP /218,219/ (NORMALLY FFBA)
           FFBA IF LOAD FLAG OFF, BRANCH TO FFD8
           FFBF IF SPACE BAR HIT, BRANCH TO FFD5
           FFCB IF ACIA NOT READY, BRANCH TO FFBF
           FFDI LOAD CHAR FROM ACIA AND RTS
           FFD5 TURN OFF LOAD FLAG
           FFD8 JMP TO FDOO (KEYBOARD SCAN SUB)
           FDOO (RETURNS WITH CHAR. IN A-REGISTER)
           FDCE RTS
     A389  TIME DELAY?
     A396
     A397  MASK MSB OF CHAR. TO ZERO
     A399  IF CHAR, IS "CNTRL 0" TOGGLE OUTPUT
           FLAG (0064)
     A3A5  RTS
A35C  IF CHAR. IS "BEL" (END OF LINE), BRANCH TO A376
A360  IF CHAR. IS CARRIAGE RETURN,   BRANCH TO A866
A866  PUT A NULL AT END OF LINE IN THE BUFFER (THIS SUB ALSO SETS X REGISTER & Y-REGISTER TO POINT
       AT BUFFER
      FOR GET-CHAR. SUB)
A86C  (SEE BELOW)
A88A RTS   GO TO TOKENIZE BUFFER ROUTINE-THE END.
A364  IF CHAR. IS LESS THAN 20 OR GREATER THAN 7D THEN A359
A36C  IF CHAR. IS @ (ERASE LINE) THEN A351
A370  IF CHAR. IS 5F (BACKSPACE, SHIFT 0) THEN A34B
A376  IF LINE LENGTH IS GREATER THAN 71(DEC.) THEN A37C
A378  STORE CHAR. IN BUFFER
A37A  INC X-REG. (BUFFER COUNTER) AND GOTO A37E
A37C  CHANGE A-REG. (CHAR. INPUT) TO "BEL"
A37E JSR   A8E5
     A8E5  IF OUTPUT FLAG(0064) IS ON, RTS (NO OUTPUT)
     A8EA  IF CHAR. IS LESS THAN 20(BEL, CR, LF)
           BRANCH TO A8F9
           CHARS ALLOWED PER LINE, JSR A86C
           A86C PUT CR IN A-REG. (TO BE OUTPUT)
           A86E PUT CR IN ACCRESS 0E
           A870 JSR A8E5
                A8E5
                .....
                A901 RTS
           A873 PUT LF IN A-REG.
           A875 JSR A8E5
                A8E5
                .....
                A901 RTS
           A87A OUTPUT NO. OF NULLS IN ADDRESS OD
           A886 ZERO ADDRESS OE (NO. OF CHARS. SINCE CR)
           A88A RTS
      A8F7 INC  0E
      A8FA JSR  FFEE
           FF69 JSR BF2D
                BF2D VIDEO OUTPUT ROUTINE
                .... (THIS WILL BE EXPLAINED
                NEXT INSTALLMENT.)
                BF72 RTS
           FF6D IF SAVE FLAG /0205/ IS OFF, RTS
           FF73 JSR FCB1
                FCB1 IF STATUS REG.(f000) OF ACIA
                NOT READY, THEN FCB1
                FCBA WRITE CHAR. TO ACIA (F001)
                FCBD RTS
           FF76 IF CHAR WAS NOT A CR, RTS
           FF7D WRITE 10(DEC.) NULLS TO ACIA
           FF8A RTS
      A901 RTS
A381  BRANCH TO A359
```

©

FOOTU: FOO Revisited

A Game For The OSI C1P, or how we learned the true meaning of the oft used phrase "This program is easily adapted to. . ."

Charles M. and Michael J. De Santis

On p. 26:45 of the July 1980 issue of **MICRO,** the "small systems jornal" from Ohio Scientific listed a little race program called "FOO". It was stated that the program would run on disk based OSI machines but that "the program is easily adapted to" OSI basic-in-ROM machines. Well, maybe its easy if you're one of OSI's computer designers or software whizzes and know where all the goodies are tucked away inside all the OSI computers, but my son Mike and I had one devil of a weekend getting "FOO" to run on our diskless — C1P. However, I can't say it was a bad experience because we learned a lot about our little machine and have come up with a couple of things that should be of interest to other C1P owners as well.

A Carriage Control

For instance, did you know that SPC (0) when used in a PRINT statement causes about 15 line feeds to occur. We discovered this one while trying to figure out why the roadway on OSI's version of "FOO" would space out and break up occasionally (see their line 550).

Keyboard Control Routine

After that was corrected, our next problem was to get the vehicle in the game moving under keyboard control. We found that, for some reason that we didn't want to take a lot of time to discover, the subroutine starting at line 600 of the OSI version of the game wouldn't work on the C1P as the program was originally written.

To correct this problem, we just re-wrote the subroutine using the "more standard" format from the OSI graphics manual, i.e. POKE 57088, row #: IF PEEK (57088) = col. # THEN ...etc. However, our keyboard control software evolved into a form that we think is really useful for many other programs.

In the typical game program as in "FOO", numbers, i.e. number keys, are used to control the direction of an object on the video screen, e.g. "1" for movements to the left and "2" for movements to the right. A problem with this approach usually crops up at the end of a game if, for instance, an INPUT statement is used to query the user about continuing. If the player isn't fast enough (he's just been controlling a space ship and has crashed into a star at 30,000 mi/hr.) he enters a "1" or "2" where a "Y" or an "N" was expected, and he has to fuss around to correct the entry or restart the program if he's already hit the RETURN. The more insidious version of this problem arises when the "keyboard-control-during-program-execution" feature is turned off while you're still holding down the "1" or "2" key. This situation usually arises abrubtly because of a game rule violation of some sort. The game stops and control returns to BASIC. This happens so fast that you're still holding down one of the number control keys, and BASIC interprets this to be the entry of a program line number. If you type anything else and then hit the RETURN you've just added a new line to your program; and you won't know it until the next time you try to run it. My favorite error in this regard ends up with line 1 reading: 1 LIST. When the program is run, I get a listing.

Well here's how to fix things so that the problem never happens again. First of all, don't use numbers for control functions (obvious, right?); we've used the left and right shift keys for control for several reasons: (a) they're spaced a nice distance apart for hand control; and (b) they're both accessible using the same row number in the keyboard polling routine.

Secondly, and this is where the serendipity comes in, the SHIFT LOCK key must be released in order for the SHIFT keys to be activated since it is also accessed through the same row number. In our version of "FOO", after all of the game options are selected, we use instructions such as:

```
270 PRINT "TO START, RELEASE SHIFT LOCK"
271 POKE 57088, 254: IF PEEK (57088) = 254 THEN
    270
```

The "254" is the column number of the SHIFT LOCK key on the polled keyboard so that line 271 keeps getting repeated until the SHIFT LOCK is released. As soon as it is released, the game starts and the shift keys are active. If the game should end abruptly or unexpectedly, and keys that may have been pressed are not entered because the RETURN key is inactive while the SHIFT LOCK key is not depressed.

The SHIFT LOCK must be pressed in order for BASIC to respond. At the end of the game or at any intermediate INPUT statement, we print a reminder to "PUSH THE SHIFT LOCK" for the proper data entry or to restore normal operation. It's a great way to do it! Try it, you'll like it.

FOOTU — C1P Version

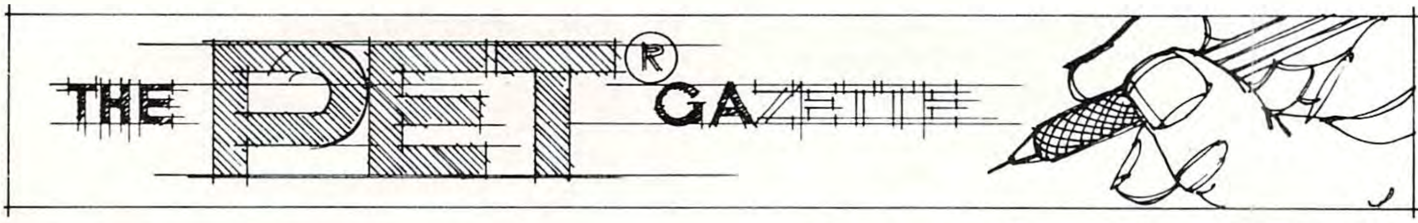
Listing 1 is our version of FOO modified to run on our C1P which has 4k of memory. Some of the scaling factors of the original program have been eliminated and the SHIFT and SHIFT LOCK features discussed in this article are employed. The display has been scaled to fit the C1P's capabilities. For other machines, lines 110, 230, 240, 290 and

520 may have to be modified. Also lines 600 – 660 will have to be modified for C2 and C4 computers. Just remember . . . . "This program is easily adapted to . . ." Good Luck!

FOOTU

```
100 POKE 530, 1
110 KY=57088:SM=2:MS=1:RN=0
115 BS=54051
117 ML=0:SN=255
120 LP=5
130 PL=2/LP
155 KP=0
160 IF A$="Y" THEN ME=EM:WI=WF:GU=UG:
    GOTO 270
170 FOR I=1 TO 30:PRINT:NEXT I
180 PRINT "FOOTU"
190 PRINT:PRINT"RACEWAY"
200 PRINT:PRINT"YOU RUN AT YOUR OWN
    RISK!"
210 PRINT:PRINT"LEFT=LEFT SHIFT RIGHT=
    RT SHIFT"
215 PRINT:PRINT"OVERDRIVE=CTRL"
230 PRINT: INPUT"INITIAL WIDTH (1-20)";WI
240 PRINT:INPUT"DELAY(1-15)";ME
241 EM=ME
245 PRINT
250 GU=0:INPUT"PEDESTRIANS (Y/N)";X$:
    IF LEFT$(X$,1)="Y"THEN GU=.3
260 KP=0: INPUT "KILLER FOO (Y/N)"; X$:
    IF LEFT$(X$,1)="Y" THEN PK=1
270 PRINT: PRINT "TO START PRESS SHIFT
    LOCK"
271 POKE KY, 254: IF PEEK (KY)=254 THEN 271
280 FOR I=1 TO 30: PRINT:NEXT I
290 WD=WI:WF=WI: WI=(12-WI) /2
291 ME=54060-ME*32
300 FOR M=1 TO LP; GOSUB 600: GOSUB 500:
    ML=ML+1: NEXT M
350 WI=WI-1
370 IF WI 4 THEN 300
400 FOR M=1 to LP: GOSUB 600: GOSUB 500:
    ML=ML+1: NEXT M
450 WI=WI+1
470 IF WI<WD THEN 400
490 GOTO 300
500 RN=RN+SM*RND (1)–MS
510 WT=WT+SGN(RN)
520 IF WI+WT>20 THEN WT=WT-1: RN=0
530 IF WT<0 OR WT=0 THEN WT=WT-1: RN=0
540 IF WI<0 THEN WI=2
545 IF WI>8 AND RND (1)<GU THEN POKE
      BS+WT+1+INT (WI*RND(1) ), 240
550 PRINT SPC (WT); "XX"; SPC (WI); "XX"
560 RETURN
600 POKE Y, 254
610 IF PEEK (KY) =251 THEN ME=ME–1KK=–1
620 IF PEEK (KY) =253 THEN ME=ME+1:KK=1
630 IF PEEK (KY) =191 THEN ME=ME+KK
640 IF PEEK (ME)<>32 THEN 700
650 POKE ME, C
660 RETURN
700 IF PEEK (ME) =240 THEN GY=240
705 IF GY=240 AND PK THEN KP=KP+1:
    GY=0: GOTO 650
720 PRINT "YOU BLEW IT!"
725 PRINT
730 MI=ML*PL
750 PRINT "AFTER"; MI;"MILES"
755 IF PK THEN PRINT "AND"; KP; "KILLS"
757 PRINT: PRINT "TOTAL POINTS:";
    INT(MI+4*(1-PK) *MI + 100*KP)
760 GOSUB 1000
810 GOTO 5000
1000 IF PK THEN WD=KP: GOTO 1030
1010 WD=MI/WF
1030 PRINT: PRINT "CONGRATULATIONS"
1040 PRINT "YOU MAY NOW CALL"
1045 PRINT "YOURSELF"
1050 PRINT: PRINT "       "
1060 IF WD<3 THEN PRINT "LITTLE"; GOTO 1200
1070 IF WD<5 THEN PRINT "TENDER";:
     GOTO 1200
1080 IF WD<12.5 THEN PRINT "MEDIOCRE";:
     GOTO 1200
1099 IF WD<25 THEN PRINT "BIG";: GOTO 1200
1100 IF WD<38 THEN PRINT "MASTER";:
     GOTO 1200
1110 IF WD<50 THEN PRINT "GRAND";:
     GOTO 1200
1120 PRINT "CHEATER"
1200 PRINT "FOO"
1210 IF GY=240 THEN PRINT "KILLER!"
1220 PRINT "!"
1230 RETURN
5000 PRINT: PRINT: PRINT "PRESS SHIFT LOCK"
5001 PRINT: INPUT "AGAIN"; A$: A$= LEFT$
     (A$,1)
5010 IF A$<>"Y" THEN 6000
5020 INPUT "SAME"; A$:A$= LEFT$ (A$,1)
5025 IF A$<>"Y" THEN CLEAR
5030 GOTO 100
6000 END
```

©



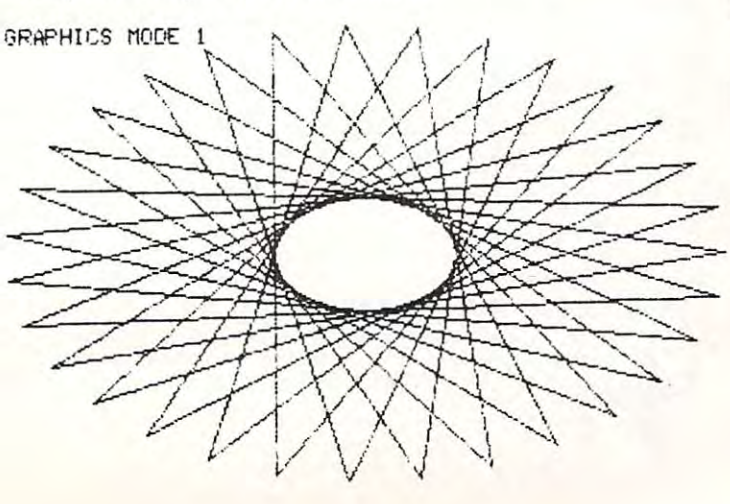
A Fast Visible Memory Dump

Martin J. Cohen, Ph.D.
Los Angeles, CA

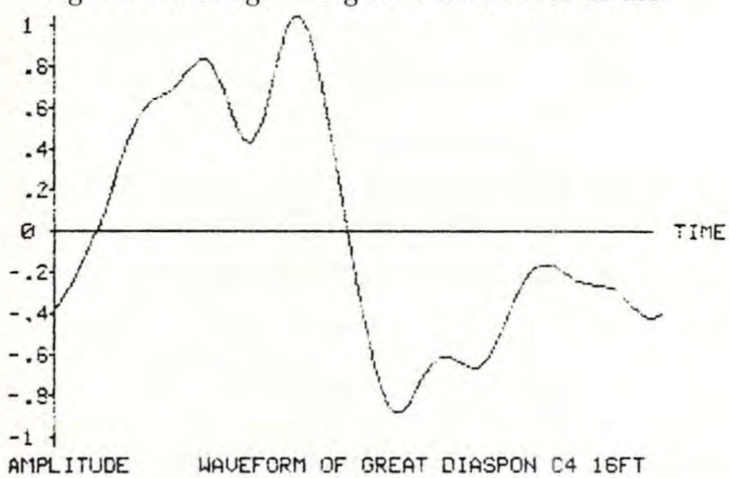
"The MTU Visible Memory is 8K bytes of dynamic RAM which, during refresh (transparent to the 6502), generates a video image of itself. The 320 (horizontal) by 200 (vertical) pixel display allows you to generate moderately high resolution graphics. (64,000 individual pixels can be set on or off — obviously a job for 6502 machine language or routines callable by BASIC.)"

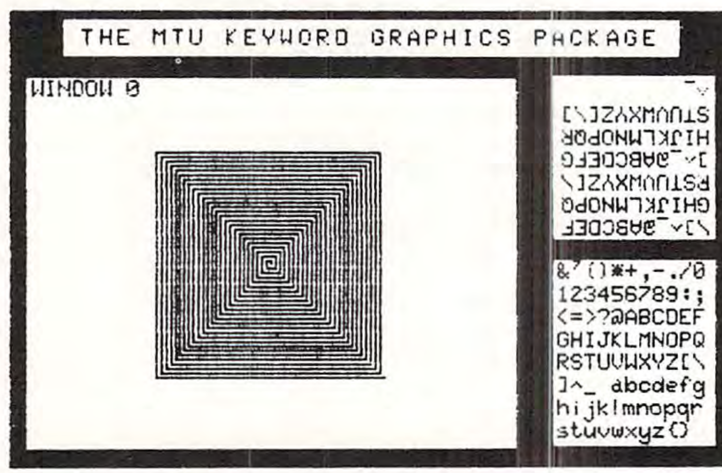
This description (on page 104 of **COMPUTE!**, issue 7, Nov./Dec., 1980) begins Dr. Frank Covitz's article in which he gives a truly ingenious method of using Commodore's 2022 tractor-feed printer to produce a hard copy of the MTU Visible memory. The primary disadvantage of this method is that, because the 2022 was not designed for graphics output, the process can take 10 to 30 minutes.

The 6502 machine language program described here, called SDUMP, produces a hard copy of the Visible Memory on Integral Data Systems' "Paper Tiger" printers with DotPlot graphics. Because these printers have graphics built in, the Visible Memory can be dumped in 90 seconds on any Paper Tiger and in only 45 seconds on the Paper Tiger 460 run at 9600 baud. These times apply to any contents of the Visible Memory, no matter how complicated or dense. The routine SDUMP does not even take advantage of clear areas of the Visible Memory, and could presumably be speeded up if this were done.



To see some of the capabilities of the Visible Memory/Paper Tiger combination, examine figures 1 through 3. Figure 1 shows four of the





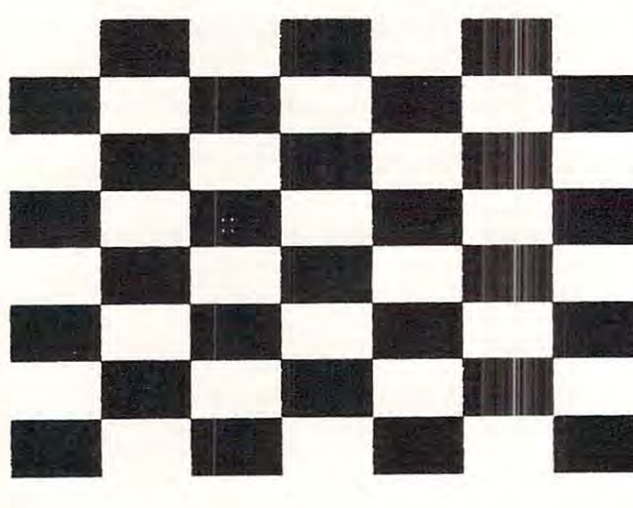


Figure 1

P1 (DENSITY) FOR K = 4, N = 2

Q	ALPHA = 0. RHO = 0.	.125 .5	.25 1	.5 2	1 4	2 8	4 16
.05	.285	.181686	.115778	.046960	.007693	.000203	0.
.1	.540000	.361678	.241868	.107708	.021050	.000768	0.
.15	.765000	.538112	.377244	.183775	.042380	.002078	.000004
.2	.960000	.708930	.520502	.276517	.074611	.004827	.000015
.25	1.125	.871864	.669801	.386991	.121355	.010224	.000052
.3	1.26	1.024422	.822792	.515782	.186954	.020314	.000166
.35	1.365	1.163877	.976548	.662790	.276438	.038453	.000491
.4	1.44	1.287245	1.127478	.826958	.395385	.069998	.001387
.45	1.485	1.391273	1.271233	1.005928	.549572	.123247	.003769
.5	1.5	1.472420	1.402602	1.195608	.744344	.210629	.009896
.55	1.485	1.526836	1.515399	1.389644	.983511	.350033	.025174
.6	1.44	1.550348	1.602333	1.578762	1.267587	.565844	.062113
.65	1.365	1.538434	1.654867	1.749954	1.591018	.888714	.148533
.7	1.26	1.486205	1.663062	1.885497	1.938004	1.351842	.343298
.75	1.125	1.388381	1.615403	1.961747	2.276254	1.979278	.762388
.8	.960000	1.239265	1.498604	1.947681	2.547874	2.757421	1.608574
.85	.765	1.032722	1.297404	1.803133	2.656193	3.572946	3.152575
.9	.54	.762148	.994325	1.476662	2.446936	4.086017	5.446845
.95	.285	.420448	.569423	.903005	1.681591	3.482044	7.005667

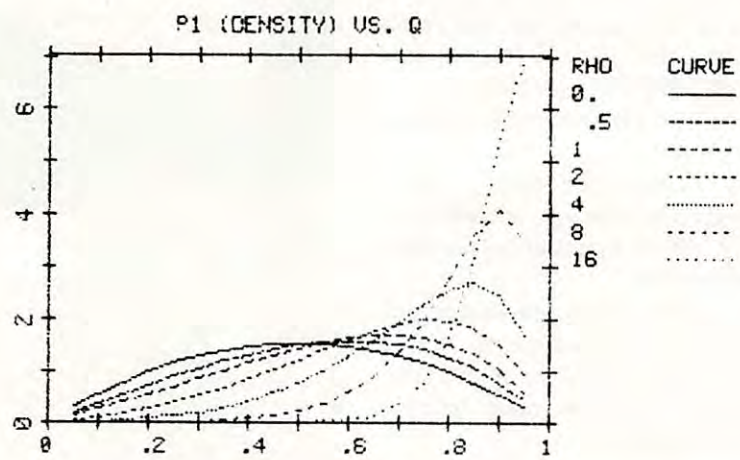


Figure 2

screens produced by the demonstration program supplied with the visible Memory. Figures 2 and 3 show some intermixed text and graphics produced using the MTU Keyword Graphics Package, of which I am the principal author. This package interfaces with BASIC to allow graphics commands to be entered as part of your BASIC program. Listing 1 shows the code used to produce the plots in figures 2 and 3.

The principal problem in dumping the Visible Memory to the Paper Tiger is that a byte of the Visible Memory is displayed as 8 pixels lined up horizontally, while a byte output to the Paper Tiger in graphics mode produces, depending on the model, 6 or 7 dots lined up vertically. The main task of SDUMP is therefore to take 6 or 7 bytes in the Visible Memory which are lined up vertically and convert them to 8 bytes of 6 or 7 bits which will then be output to the Paper Tiger.

My first attempt at this was done in BASIC, and is in listing 2. I knew it would execute extremely slowly, but it would be much easier to debug. Once the code was working, it was a fairly straightforward matter to translate the BASIC into assembly language — since I knew the logic was correct, I only had to make sure the translation was correct. Another advantage of this method is that if I want to program the routine in some other language, such as PASCAL, FORTH, or FORTRAN, it will be much easier to do it with BASIC as the basis instead of assembly language.

The current version of SDUMP is in listing 3. It is a modularized form of the BASIC code in listing 2, and is designed to be easily modifiable. It is assembled starting at $6000 (hex), so that it can reside in memory with the MTU Keyword Graphics Package, and be called with a SYS (96*256).

The initial part of SDUMP contains a transfer vector and a data area. The transfer vector has jumps to the three main routines in SDUMP: OUTVM, which dumps the whole Visible Memory; OUTROW, which dumps 6 or 7 rows of the Visible Memory starting at the location set in VM (at $6013); and OUTCOL, which outputs a column of