Introducing The OSI Gazette

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

Hard Disks For The Apple

COMPUTE!

$2.00
November/December, 1980
Issue 7
Vol. 2, No.6

The Journal for Progressive Computing™

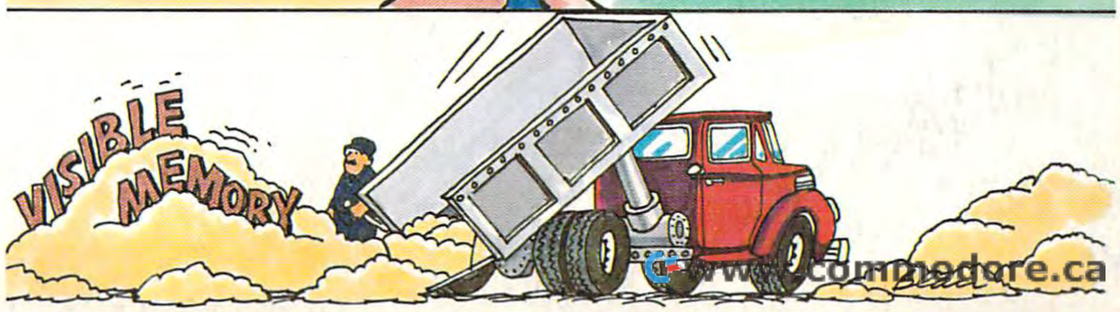
COMPUTE! Looks At The New TRS-80 Color Computer?

Times Square On Your Atari

Interfacing KIM/SYM/AIM/OSI With BASIC

Combining BASIC And Machine Language, II

Visible Memory PET Printer Dump



www.commodore.ca

TRENDCOM 200

High-Speed Intelligent Printer

40 characters-per-second

80 characters per line

Upper and lower case

Continuous graphics at 60 dots per inch

Microprocessor controlled

Bidirectional look-ahead printing

Automatic "wrap-around"

Quiet operation

The Trendcom 200 is a high speed thermal printer offering the combination of text printing at 80 characters per line and continuous graphics at 60 dots per inch. In the text mode, upper and lower case data are printed at 40 characters per second. The 5 x 7 characters provide clear readable copy on white paper; no hard to find, hard to read aluminized paper.

In the graphics mode, seven bits of each byte correspond to the seven dots in each of the 480 print positions per line. Since the computer driving the printer has full control over every print position, it can print graphs, bar charts, line drawings, even special and foreign language symbols. Despite its low cost, the Trendcom 200 is a true intelligent printer with full line buffering and bidirectional look-ahead printing. After one line has been printed left to right, the internal microprocessor examines the next line to choose the shortest print direction. The microprocessor also provides a built-in self-test mode for easy verification of proper operation.

High reliability is designed in: The thick film thermal print head has a life expectancy of 100,000,000 characters. Two DC stepping motors provide positive control of the print head and the paper drive, the printer's only driven parts. The absence of gears and solenoids also makes the printer extremely quiet; the only noise is the rustling of the paper advancing.

——— **Skyles PAL-80 printer(s) complete** with 2½ foot interface cable to attach to my PET at **$675.00 each.*** (Plus $10.00 shipping and handling.) I also will receive a test and graphics demonstration tape at no additional charge and over 150 feet of 8½ inch wide black on white thermal paper.

——— rolls of **8½ inch wide** by 85 ft. long thermal paper (black ink) at $5.00 each; or cartons at 10@ $45.00.

Visa, Mastercharge orders call (800) 538-3083

California orders please call (408) 257-9140

Skyles Electric Works

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

www.commodore.ca

METAMORPHOSIS!

Now you can add high resolution graphics to your Commodore PET computer. The MTU K-1008-6 GRAPHIC INTERFACE can be used with either old, new, or business PET computers. It is simple to use, and fits inside the PET for protection.

The GRAPHIC INTERFACE gives you easy control over each dot in a matrix which is 320 wide by 200 high for a total of 64,000 dots. Because each dot can be controlled, either graphic images, text lines, or any mixture of the two can be displayed. Since each dot is controlled from software you can even design your own special character font or graphic image set (logic, chemical, architectural).

INTERFACE TO ALL PETS - With separate connector boards for each style PET (K-1007-2 for OLD PETS, K-1007-3 for NEW). The K-1008-6 can be used with either.

THREE TYPES OF VIDEO - You can select either normal PET video, graphic video, or the COMBINED image of both video signals simultaneously!

8K RAM MEMORY EXPANSION - The graphic matrix requires 8K RAM which is supplied onboard. This memory can be used for program or data storage when not being used for graphics (or see your program in binary on the display!).

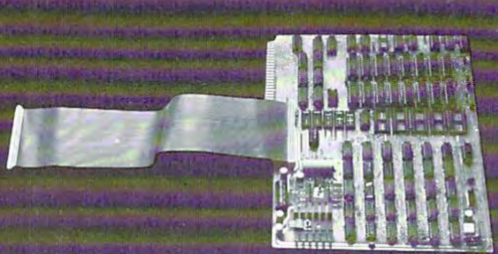
OLD OR NEW PETS CAN NOW HAVE HIGH RESOLUTION GRAPHICS

FLEXIBLY ADDRESSED ROM SOCKETS - Five ROM sockets are included on the board. They can be set *at the same* or different addresses, with you controlling which sockets are enabled at any time through software control. You also choose the sockets to be enabled when the PET is turned on.

EXTERNAL EXPANSION - This board also creates the KIM memory expansion bus supported by all MTU products. This allows insertion into our K-1005-P card file for expansion up to 4 other boards outside the PET case.

LIGHT PEN - The board has been designed to work with an optional light pen which MTU will be announcing soon.

SOFTWARE INTERFACED TO BASIC - MTU also has available machine language software to allow you to plot points, draw lines, and display characters at high speed.



Call or write for our full line catalog of products.

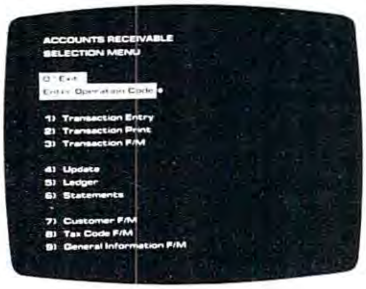
MICRO TECHNOLOGY UNLIMITED
P.O. Box 12106
2806 Hillsborough Street
Raleigh, N.C. 27605
(603) 627-1464

MTU
Micro Technology Unlimited
P.O. Box 12106
2806 Hillsborough Street
Raleigh, N.C. 27605
As of June 1, 1980

www.commodore.ca

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. **$295.00**

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 **$195.00**

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 **$195.00**

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 $350.00

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

www.commodore.ca

Table of Contents

November/December 1980. Issue 7, Vol. 1. No. 6

Page 56

Page 70

Page 104

Page 128

COMPUTE. The Journal for Progressive Computing (USPS: 537250) is published six 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, $16.00. 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. Entire contents copyright © 1980 by Small System Services, Inc. All rights reserved. ISSN 0194-357X.

www.commodore.ca

Robert C. Lock, Publisher/Editor
Joretta Klepfer, Manager, Dealer Marketing
Carol Holmquist Lock, Circulation Manager
Kathleen Martinek, Publication Assistant
J. Gary Dean, Art Direction/Production Assistance

COMPUTE receives continuing editorial assistance from the following persons:
Harvey Herman, University of North Carolina at Greensboro
Jim Butterfield, Toronto, Canada
Larry Isaacs, Raleigh, NC

The following writers contribute on a regular basis as Contributing Editors:
Al Baker, 2327 S. Westminster, Wheaton, IL 60187
Gene Beals, 115 E. Stump Road, Montgomeryville, PA 18936
Len Lindsay, 5501 Groveland Terrace, Madison, WI 53716
Roy O'Brien, P.O. Box 426, Beaumont, CA 92223

Subscription Information (12 Issue Year):
COMPUTE. Circulation Dept.
P.O. Box 5406
Greensboro, NC 27403 USA
U.S. $16.00
Canada $18.00 (U.S. funds)
Europe: Surface Subscription, $20.00 (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

Authors of manuscripts warrant that all materials submitted to COMPUTE. are original materials with full ownership rights resident in said authors. By submitting articles to COMPUTE., authors acknowledge that such materials, upon acceptance for publication, become the exclusive property of Small System Services, Inc. Programs developed and submitted by authors remain their sole property, with the exception that COMPUTE. reserves the right to reprint the material, as originally published in COMPUTE., in future publications. Unsolicited materials not accepted for publication in COMPUTE. will be returned if author provides a self-addressed, stamped envelope. Program listings should be provided in printed form (new ribbon) as well as machine readable form. Articles should be furnished as typed copy (upper and lower case, please) with double spacing. Each page of your article should bear the title of the article, date and name of the author.

COMPUTE. assumes no liability for errors in articles or advertisements. Opinions expressed by authors are not necessarily those of COMPUTE.

PET is a trademark of Commodore Business Machines, Inc.
Apple is a trademark of Apple Computer Company.
Atari is a trademark of Atari, Inc.

The Editor's Notes

Robert Lock, Publisher/Editor

The TRS-80 Color Computer?

Yes, it's reviewed in this issue, even though its heart is a 6809 rather than a 6502. We felt it would be of interest, given the increasing number of small, inexpensive color machines. And it will provide a bit of background information for our full scale review of VIC in the January issue.

Recommended Reading

One of the problems with our current "Gazette" method of organization is that articles sometimes become compartmentalized when they really shouldn't. Three good examples in this issue appear in the **Apple Gazette** (Anatomy of A Word-Research Processing Program) and the **SBC Gazette** (Mixing BASIC and Machine Language, by George Wells; Interfacing KIM,SYM,AIM,OSI to BASIC, by Jim Butterfield). These articles have some relevance regardless of your machine, and if you're interested in refining your programming techniques, or learning more about the inner workings of your Microsoft machine, take a look at these.

One possibility is to divide the articles into content areas rather than machine ones. Using this method, we'd have a programming techniques section, a machine language programming section, etc. On the other hand, we'd have problems with machine specific material. Please use the Editor's Feedback Card this time to provide your input on the current organization of the magazine.

More On Recommended Reading

"Computing Correlation Coefficients" (see Table of Contents) was written by a TRS-80 owner. You'll notice that some of the code looks a bit strange. We're using the article because we feel it's an excellent introduction to its title, and if you're statistically inclined you should find the article useful. Keep me posted on getting up and running on the 6502!

In The Review Queue

We currently are using Personal Software's **Visicalc for the PET.** Much to my delight it not only works as promised, but has one of the most professional pieces of documentation I've seen in a long time.

Word Pro III + and **Word Pro IV +** have replaced the recently introduced Word Pro III and Word Pro IV from Professional Software, Inc. Several new features have been added that we haven't seen yet. Rather than review the current ver-

www.commodore.ca

sions of WPIII and WPIV, we'll wait for the new ones. While prices have risen accordingly, the older versions we have in house work as advertised.

Regency's **Electric Crayon** (for the PET/CBM) is here waiting for us to hook it up and get it rolling (scrolling?). Other software in the queue includes PET terminal packages from Madison Computer and Micro Computer Industries; ATARI educational software from T.H.E.S.I.S.; and some other goodies we'll preview in the months ahead! Remember that COMPUTE! will be showing up every month beginning in January.

What is it?

(This information compiled by Dr. Chip and I through various sources...)

It's beige, has an RS-232 standard data bus, accepts external plug-in memory and cartridges, has a graphics character set, and comes with interface for joystick/light pen/game paddle. Its memory is expandible to 32K RAM. It has a full-sized standard keyboard, and four special function keys with 8 special functions (some or all of these will be programmable). It's **VIC from Commodore.** Currently being introduced in Japan, it's supposed to be available here in the next few months. Check with your dealer for more information.

The bottom-line configuration has a 22 character by 23 line display with a graphics resolution of 176 by 184. Balanced against a $299.00 price tag, VIC ought to help keep things interesting around this marketplace. Dr. Chip and I are anxious to get our hands on one.

The Missing Advertisers

We're not sure what happened, but GPA Electronics, a recent advertiser in COMPUTE! and several other publications, seems to have gone away. We've had one complaint from a British reader of a cashed prepayment check and no further response. As far as we can tell, the phone has been disconnected. On the other hand, we've had no other complaints. Does anyone have any other information?

RAYGAM appears to have closed as well, albeit in a more orderly fashion. We received notice that they were closing shop. All of which brings us to that perpetual problem of how you determine the stability of your suppliers? I don't have any ready answers, but would certainly welcome some input. COMPUTE! will be happy to provide some space to stimulate the discussion. ©

First Monthly Issue is January, 1981

Good News for COMPUTE! Authors

Our reorganization is complete, with the addition of Kathy Martinek to our staff. She will be coordinating all unsolicited manuscripts so you'll be getting faster, more reliable responses.

Our New System

You'll receive an acknowledgement by return mail. Within one to two weeks we'll let you know whether we plan to use the material, etc.

If You Haven't Heard

With Kathy's help, we've caught up on all outstanding material. If you submitted something before October 1, and haven't heard from us by now, give me a call.

Keep Those Articles Coming

COMPUTE! Needs You...

You May Be Expiring...

or
Renewing Your Subscription To COMPUTE!

If you're an early COMPUTE! subscriber, your subscription may be running out. If your mailing label bears the code "11/80" or "12/80" then Issue #7, the November/December issue, is your last one.

Don't Miss An Issue

Renew now by sending us your check or money order for your 1980 subscription. Please follow these simple guidelines. Mail your renewal check to COMPUTE! P.O. Box 5406, Greensboro, NC 27403 USA. Mark the envelope "Attention: Subscription Renewal". Include your current mailing label. If you've thrown your envelope away, please make sure that you include your name and address (especially your address) the same way you've been receiving the magazine. Check the new price schedule and include a check, money order or Master Charge/Visa number with your renewal.

www.commodore.ca

NEECO

"Your complete source for all CBM Hardware and Software Products"

PROUDLY ANNOUNCES OUR NEW ONE YEAR WARRANTY ON ALL CBM COMPUTERS!

The 8032 CBM Computer is now available!

commodore

NEW! operating system and new disk operating system with direct access!

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.

CBM™ 8050 DUAL DRIVE FLOPPY DISK

The CBM 8050 Dual Drive Floppy Disk is 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 (single sided)
- 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 2.1
- 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	PRODUCT DESCRIPTION	PRICE
4008N	8K RAM-Graphics Keyboard-40 col.	$ 795.00
4016N	16KN RAM-Graphics Keyboard-40 col.	$ 995.00
4016B	16K RAM-Business Keyboard-40 col.	$ 995.00
4032N	32K RAM-Graphics Keyboard-40 col.	$1295.00
4032B	32K RAM-Business Keyboard-40 col.	$1295.00
8016	16K RAM-80 Col.-4.1 O/S	$1495.00 *
8032	32K RAM-80 Col.-4.1 O/S	$1795.00
2023	Friction Feed Printer	$ 695.00
2022	Tractor Feed Printer	$ 795.00

NOTE: All current CBM production computers/disks now contain operating system 4.1/DOS 2.1

CBM	PRODUCT DESCRIPTION	PRICE
2040	Dual Floppy-343K-DOS 1.0	$1295.00
4040	Dual Floppy-343K-DOS 2.0	$1295.00
8050	Dual Floppy-974K-DOS 2.0	$1695.00 *
C2N Cassette	External Cassette Drive	$ 95.00
CBM to IEEE	CBM to 1st IEEE Peripheral	$ 39.95
IEEE to IEEE	CBM to 2nd IEEE Peripheral	$ 49.95
8010	IEEE 300 Baud Modem	$ 395.00 *
2.0 DOS	DOS Upgrade for 2040	$ 50.00
4.0 O/S	O/S Upgrade for 40 Column	$ 100.00

*Asterisks indicate fall delivery—all others are immediately available

SPECIAL OFFER ON CBM COMPATIBLE BUSINESS SOFTWARE!

Purchasing software has always been difficult due to the "you buy it - you own it" attitude of most vendors. We at NEECO, recognize this problem and can now, on all of the Software Packages listed, offer a full 30 day refund policy to NEECO's customers. Now you can purchase with confidence. Buy it - try it; if the program package is not suitable for any reason, send it back to us within 30 days and we will refund the full purchase price—less shipping charges!

NEW!

SOFTWARE	APPLICATION	REQUIRES	AUTHOR	AVAILABILITY	PRICE
Word Pro I	Word Processing	8K + cassette	Professional Software	Immediate	$ 29.95
Word Pro II	"	10K + 2040	"	"	99.95
Word Pro III Plus	"	32K + 2040	"	"	395.00
Word Pro IV Plus	"	8032 + 2040/8050	"	"	595.00
BPI Integrated G/L	Business	32K/8032 + 2040	BPI	"	360.00
BPI Inventory	"	"	"	"	T.B.A.
BPI Payroll	"	"	"	"	
BPI Enhanced A/R	"	"	"	"	
CMS G/L	"	"	CMS Software	"	295.00
CMS A/R	"	"	"	"	195.00
CMS A/P	"	"	"	"	195.00
CMS Customer Mail List	"	"	"	"	195.00
CMS Payroll	"	"	"	"	350.00
Datasource 1	All Business	32K/8032 + 2050/8050	BMB	August/Sept.	295.00

*Wordprocessing Software requires output printer. We recommend the NEC Spinwriter ($2995) for letter quality.
*PET is a registered trademark of Commodore Business Machines. Small Keyboard PETS require a ROM Retrofit Kit.
Multi-Cluster is available in Canada from BMB Compu Science, P.O. BOX 121, Milton, Ontario, L9T2Y3
All prices and specifications are subject to change without notice.

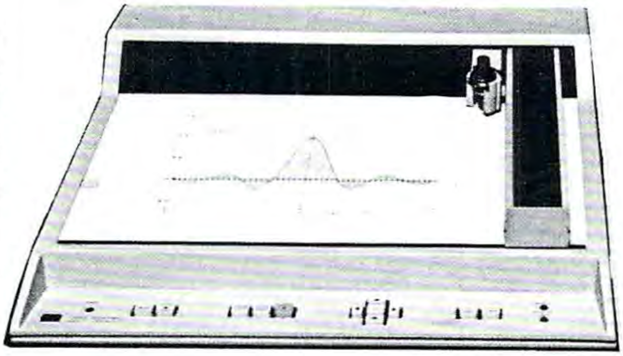
NEECO
679 HIGHLAND AVE.
NEEDHAM, MA 02194

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

www.commodore.ca

HEWLETT-PACKARD PLOTTER for CBM SYSTEMS



The Hewlett-Packard 7225A is a compact and efficient graphics plotter that provides a cost effective solution to the need for professional hard-copy graphics. With the 7225A, publication quality graphics can be drawn with clean and visually continuous ink lines.

The HP 7225A can be user-adapted to a wide range of systems. By changing a "plug-in" unit, called a Personality Module, the 7225A will provide the appropriate interface, language, and graphics capabilities for a variety of desktop computers, computer systems, personal computers, terminals and intelligent instrument systems. This interface flexibility makes the 7225A the ideal plotter for many present and future configurations.

THE 7225A GRAPHICS PLOTTER WITH PERSONALITY MODULE

$2800.00

Interconnect Cables

2 meters (6.6 ft)	$75.00
4 meters (13.1 ft)	$85.00

OPTIONS:

Opt. 007 (15 pads of 8½ × 11 in. paper, assortment of pens, pen holder	$ 75.00
Opt. 010 (vinyl carrying case)	$125.00
4-Color Pen Pack (Red, Blue, Green, Black)	$ 6.00
Plotter Paper (100 sheets 8½ × 11 in. grid)	$ 7.00
Plotter/Printer ROM Handbook	$ 10.00

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

$349.00

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
679 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.

www.commodore.ca

The Reader's Feedback

Robert Lock, Publisher/Editor and Readers

Well group, **COMPUTE!** is happy to present our biggest issue ever. Welcome back to you OSI, KIM, SYM and AIM users. Next issue (January) will be the first of the new monthly **COMPUTE!'s.**

Best Article last issue... I'm calling it a three way tie, with Apple honors going to "Screendump", PET honors going to "Feed Your PET Some APPLESOFT", and ATARI honors going to "Designing Your Own Atari Graphics Modes". In the general category, Marvin DeJong's article "Solving Equations With A Computer" is leading at press time.

Now, on with the feedback:

Hurrah!!!!! Can you make the new monthly issue bigger too?
Probably not. I expect our monthly issue to be 96-112 pages. But then again I orginally planned this one to be 120 pages. We'll have to wait and see.

Give more space to Atari and Apple and cut the PET Gazette in half.

Great that you have decided to again have one magazine and have it monthly... I hope quantity of material for PET will not be reduced.
Well, it's not all relative. We are growing in page size, especially given our new monthly frequency. Several factors help determine the extent of coverage and overall size of any given issue. One of these is advertising, but an even more critical one is material. Keep those Apple and Atari articles coming in. The same comment applies to you OSI and single-boarders as well. We rely on our readers in large part to help us keep providing the best and the latest.

I would be happy to be a business applications reviewer. I own an 8032, 2040, etc...
Oops! I assume your card arrived in an envelope with a subscription. When it reached me it was in a stack of cards. Since I'm the only one who gets the Editor's feedback cards, please make sure you add your name and address if you're including a comment such as the above. Thanks.

Regarding Columns

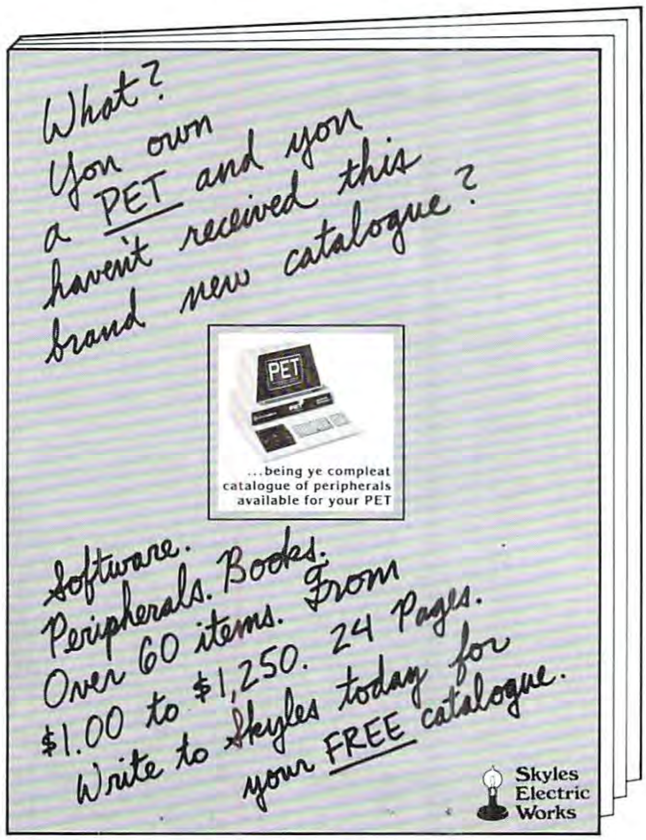
Beginning this issue for some columnists, and next for others, we'll be rotating some of our columns. This is because all do other things besides write and our new monthly schedule will be a bit hectic for them. Here's how we've paired them off; in each pair, we'll alternate months:

Nuts and Volts, Gene Zumchak
The Single-Board 6502, Eric Rehnke
Computers and Society, David Thornburg and Betty Burr
Micros With the Handicapped, Susan Semancik
Programming Hints for Apple and Atari, Al Baker
Fun With The 6502, Len Lindsay

Keep Those Cards and Letters Coming

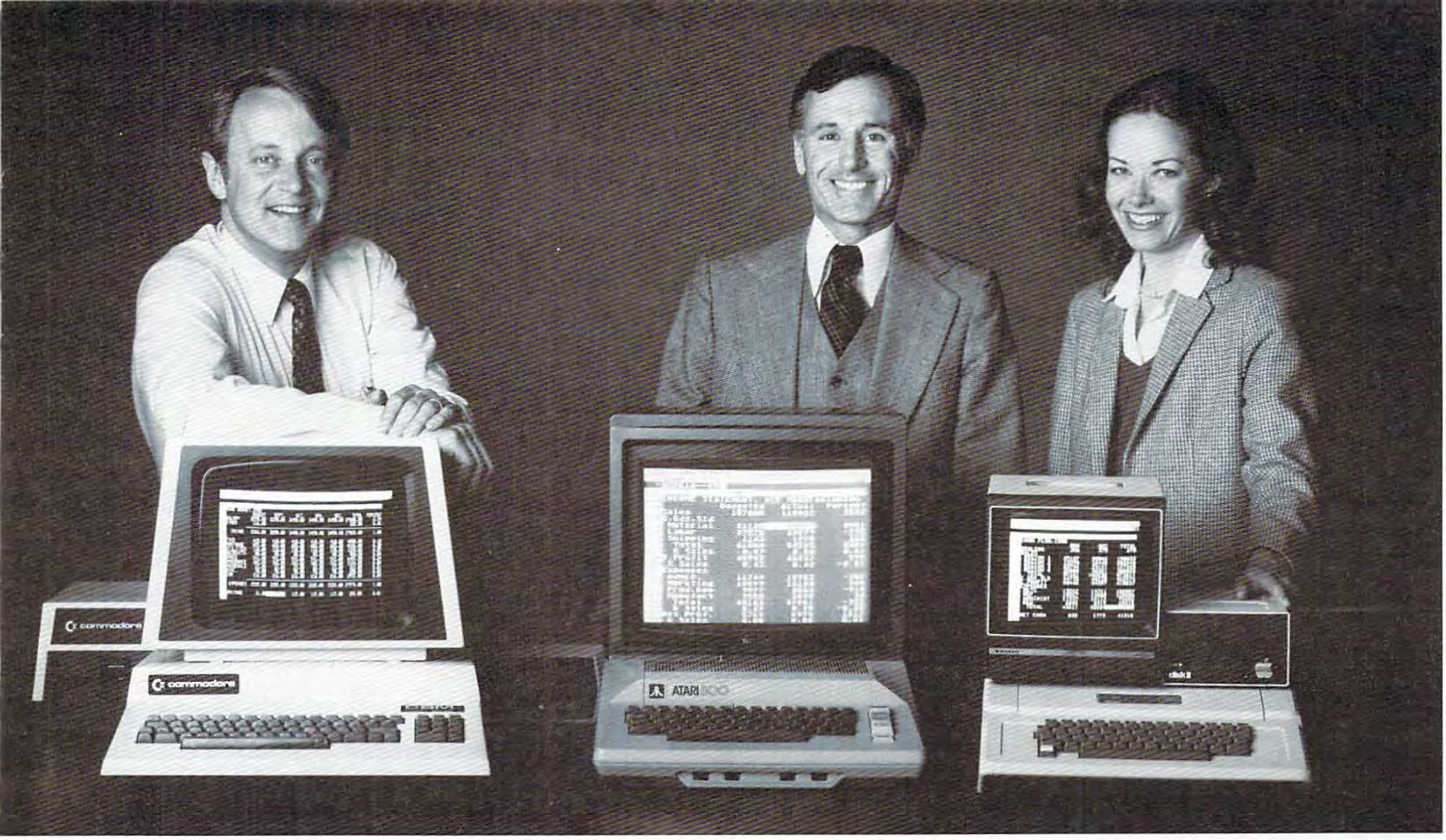
COMPUTE! Needs You!

Address articles, programming notes and comments to:
The Editor
COMPUTE!
P.O. Box 5406
Greensboro, NC 27403



Skyles Electric Works
231 E South Whisman Road
Mountain View, CA 94041

www.commodore.ca



Time & Money. Commodore®, Atari® & Apple® users get more with VisiCalc™ software.

A financial VP in Massachusetts is cutting the time it takes to prepare month-end reports from three days to three hours.

A California company is replacing most of its time-share computer service with a personal computer and VisiCalc, saving at least $30,000 the first year.

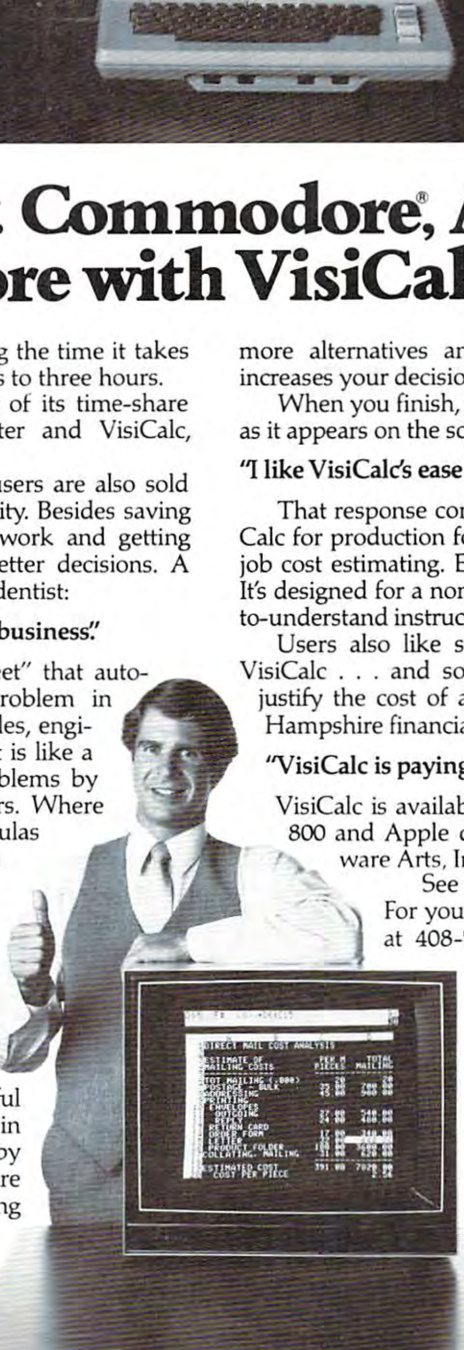
Thousands of other personal computer users are also sold on how VisiCalc is increasing their productivity. Besides saving time and money, they're simplifying their work and getting more information that helps them make better decisions. A typical user reaction comes from a New York dentist:

"VisiCalc has become an integral part of my business."

VisiCalc displays an "electronic worksheet" that automatically calculates nearly any number problem in finance, business management, marketing, sales, engineering and other areas. The huge worksheet is like a blank ledger sheet or matrix. You input problems by typing in titles, headings and your numbers. Where you need calculations, type in simple formulas (+,–,×,÷) or insert built-in functions such as net present value and averaging. As quickly as you type it in, VisiCalc calculates and displays the results.

"I am extremely impressed with VisiCalc's capability, flexibility and orderly presentation of instructions."

So writes the director of a New York corporation. He appreciates VisiCalc's powerful recalculation feature. Change any number in your model and instantly all numbers affected by that change are recalculated and new results are displayed. You can ask "What if . . .?," analyzing more alternatives and forecasting more outcomes. It really increases your decision-making batting average!



When you finish, you can print a copy of the worksheet just as it appears on the screen and/or save it on diskette.

"I like VisiCalc's ease of use."

That response comes from a Utah businessman using VisiCalc for production forecasts, financial report ratio analysis and job cost estimating. Ease of use is VisiCalc's best-liked feature. It's designed for a non-programmer, and has an extensive, easy-to-understand instruction manual.

Users also like solving a wide variety of problems with VisiCalc . . . and solving them their way. VisiCalc can even justify the cost of a personal computer, according to a New Hampshire financial analyst:

"VisiCalc is paying for itself over and over."

VisiCalc is available for 32k Commodore PET/CBM, Atari 800 and Apple disk systems. VisiCalc is written by Software Arts, Inc.

See VisiCalc at your Personal Software dealer. For your dealer's name, call Personal Software Inc. at 408-745-7841, or write 1330 Bordeaux Drive, Sunnyvale, CA 94086.

While there, see our other Productivity Series software: Desktop Plan and CCA Data Management System. They're like time on your hands and money in the bank.

PERSONAL SOFTWARE

Commodore is a registered trademark of Commodore Business Machines Inc., Atari is a registered trademark of Atari Inc., Apple is a registered trademark of Apple Computer Inc.

www.commodore.ca

Computers and Society

David D. Thornburg and
Betty J. Burr
Innovision
P.O. Box 1317
Los Altos, CA 94022

In August one of us (DT) had the pleasure of presenting an address on the challenges of personal computer design at the Centenary meeting of the ASME. While one might not normally think of the annual meeting of Mechanical Engineers as being an appropriate place to discuss the design challenges coming from the personal computer "revolution", the ASME had divided their meeting into a series of "Emerging Technology" conferences, one of which was devoted to computers.

Because this talk provided an opportunity to deal with some broad issues, we thought it appropriate to provide a condensation of it in our column this month.

The advent of the low cost microprocessor several years ago made possible the development of a revolutionary new product - the personal computer. This product has brought the price of computational power down to the point where one can honestly forsee the presence of a computer in nearly every home.

As revolutionary as the enabling technology has been, the mere existence of the personal computer is insufficient to give everyone access to all the things computers can be used for. In order for this technology to move into people's homes, the interface between computers and people has to be improved to the point where the average person can operate the computer as easily as he or she can operate a color television. In order for the full potential of the revolution to be realized, we thus need to enter a period of coevolution - a period where advances in the computer technology become connected to advances in applications and to improved interfaces between people and machines. The task is so large and so important that there will be myriad opportunities for each of us to take part in one of the most exciting opportunities of our lifetime - the implementation of the post-industrial revolution.

The challenge faced by designers arises from the fact that an increasing fraction of computer users will have no special background or training in this technology. It is the satisfaction of these naive users that will prove to be the true test of our creative abilities. As we will see, this was not always the case.

Back in the days when computer systems were not available for under $100,000, computer designers knew that the users of their products were technical people much like themselves. The task of designing a computer system revolved around issues of speed, memory size, and cost, with not too much attention paid to human factors. Traditionally, large computer systems were controlled by highly skilled operators who received special education in the use of their particular computer. One can perceive several benefits which came from making this interface very specialized:

1. It tended to intimidate non-professionals who might otherwise tamper with the system, thus interrupting the normal operation of the machine.

2. By having the operators program in machine language, it made the computer operate more efficiently.

3. The presence of a physically impressive machine with arrays of lights and switches, coupled with the specialized jargon interspersed in the speech of the programmers, served to provide added job security for these people. There was security in becoming a scribe and joining the cult of like-minded individuals.

Some time later, increased cost pressures, coupled with the need for distributed computing, led to the $10,000 minicomputer. Aside from their cost, word size, and memory capacity, these machines shared many traits with their larger counterparts. While still operated by highly trained professionals, it wasn't uncommon to see these computers being used by physicists, chemists, and others from outside the traditional "computer" community who needed to automate certain data acquisition and reduction tasks. While many minicomputers required some machine language programming to get them started, the widespread use of higher level languages (such as FOCAL and BASIC) through "teletype" terminals made the environment a little more tractable to the non-computer specialist.

It wasn't until the development of the microprocessor and the subsequent solution of the personal computer, however, that the background of the typical computer user underwent a radical change. Even as recently as last year, many of these $1,000 computers were being used by hobbyists who were content to live with cryptic keystroke combinations to make their computers perform.

As time passes, however, it becomes increasingly clear that computers will be used more and more by people who have no computer background at all -people who will expect the computer to be as easy to use as a television or a telephone.

To see why this is so, let us look at personal computer sales over the last few years. In 1978, the first year of the "revolution", 150,000 personal computers were sold. These products mostly came from Tandy, Commodore, and Apple - none of whom had

www.commodore.ca

WordPro™ Word Processing Software

Turn your Commodore CBM/PET computer into a highly sophisticated word processing system

Everyone expected it would happen sooner or later. . . with WordPro it already has! Now all the marvelous benefits of an advanced stand-alone wordprocessor are available with the WordPro series of software and the systems they create.

If you've already been shopping for software in the crowded word-processing marketplace, you've probably determined the features you really want. You'll find WordPro has them, and more.

And if you haven't begun to shop yet, we urge you to compare, because only by comparison will you fully appreciate how complete and sophisticated WordPro software really is.

WORDPRO SOFTWARE IS LOADED WITH THE LATEST INNOVATIONS

Sophisticated systems programmed with leading edge wordprocessing features, WordPro is a series of programs designed specifically for use with the Commodore CBM/PET computers, peripherals and compatible typewriter quality printers.

WHAT MAKES WORDPRO THE BEST?

Our research has shown that while many wordprocessing packages have comparable features to WordPro, none can surpass Wordpro's EASE OF USE AND FLEXIBILITY. Wordpro operators need not be familiar with computer commands or functions. WordPro is easy to learn for anyone with ordinary typing skills.

WORDPRO GIVES YOU THREE LEVELS TO CHOOSE FROM

WORDPRO 1 is ideal for hobbyists, students and organizations who can benefit from the advantages of a basic wordprocessor without the program refinements of a comercially oriented system. WordPro 1 is recommended for use with the CBM/PET 8/16K, C2N cassette and interfaced printer.

WORDPRO 3 converts the CBM/2001 32K computer into a highly sophisticated 40-column screen wordprocessor. This program incorporates the advanced features considered important to effective wordprocessing, including nearly every entering, editing, memory and printing feature available today. WordPro 3 is recommended for use with CBM/PET 32K (40-column) computer, CBM Dual Disk Drive, and a properly interfaced printer.

WORDPRO 4 has it all! With this program, you will have everything you could want from a wordprocessor. . . and then some. WordPro 4 includes every feature found on WordPro 3, but with the added advantage of an 80-column display screen. The 80-column display simplifies text editing and makes entering text in columnar formats effortless. And with a few simple keystrokes, you'll be able to visualize on the screen exactly how your document will look prior to printing it out. WordPro 4 is designed for use with the Commodore CBM 8032 computer, CBM Dual Disk Drives, and a properly interfaced printer.

Professional Software Inc.
166 Crescent Rd., Needham, MA 02194
(617) 444-5224

*WordPro International Distributor and Domestic Dealer Inquiries Invited
WordPro was developed by Steve Punter of Pro-Micro Software Ltd., and is marketed exclusively by Professional Software Inc.

WordPro is a registered trademark of Professional Software Inc. CBM is a registered trademark of Commodore Business Machines.

www.commodore.ca

been in the computer business before. By 1979, annual sales had climbed to 300,000 units, and for 1980 sales of 600,000 computers are projected. 1981 should be the first "consumer market" year, with sales of over 1,000,000 computers forcast for that year alone. No one knows the saturation level of this market, but by 1990 it is projected that more than 30 million computers will be installed in France alone. The U.S. market has been arbitraily pegged at 47 million units, but no one knows how vast this market really is.

Where will these computers be located?

Initially, personal computers appealed to hobbyists. Since the reliability, documentation, and software for these first computers was generally poor or almost non-existent, this initial core of users was appropriate to launch the industry. Some of these early users even saw their hobby as a path to riches, and became part of the burgeoning cottage industry which has grown to support these computer systems.

As time went on, some educators began to realize that low-cost personal computers made useful contributions in the classroom. The distributed (vs. time-shared) nature of this resource, coupled with the expressive capability of color and sound, made traditional computer aided instruction seem somewhat boring. The added value of teaching the children to use the computer as a tool for their own activities opened new avenues for exploration in many fields including the traditional areas of mathematics and the language arts. Here in the San Francisco Bay Area, for example, it is not uncommon to see schools introducing personal computers to children at the age of 8 or less. This type of user is a far cry from the trained professional seated at the helm of a $100,000 mainframe.

Owners of small businesses also began to see the personal computer as a tool to aid them. At a system price of $1,000 to $5,000, many small offices have found that the computer can help to handle simple accounting and filing chores more efficiently than these tasks can be done by hand. An increasing fractoin of personal computers began to be sold into this market by 1979. This area will probably continue to use the bulk of these systems for the next year or two.

There is another market which, while largely untapped today, represents the principal user of this equipment within ten years - the home. Although, except for recreation, there seems to be little use for computers in the home today, the advent and growth of remote information utilities will create tremendous demand for personal computers in the future. Today, all over the world, commercial services have been started which allow computer users to access vast data banks through a telephone line connection between the small computer in their homes and a large centrally located system. For under $3.00 per hour,

Intelligent, fast 11 x 17 Graphic Plotting for the Commodore PET®/CBM®

Com-plications™ C102 X-Y Plotter Adapter

- Increases plotting speed by a factor of 2 to 5
- Provides printed program or data listings
- Uses print statements — no assembly programming
- Compatible with all IEEE-488 (1978) controllers
- Industrial-quality construction, 72-hour burn-in
- Complete graphic software package available

Low cost, intelligent, high speed graphic, hardcopy for the Commodore PET®/CBM® is now a reality. The combination of the Com-plications™ C102 Interface Adapter and the Watanabe "MIPLOT" intelligent plotter gives the Commodore computer sophisticated plotting capabilities. Programming is accomplished with BASIC language PRINT statements, not assembly language. Axis generation, character rotation, vector generation and other functions are procedures built into the plotter. Call for a dealer package.

PET® and CBM® are registered trademarks of Commodore Business Machines, Inc.

F.I. ELECTRONICS
968 Piner Road
Santa Rosa, Ca 95401
Telephone: 707-527-0410
Telex: 33-7769

www.commodore.ca

one can peruse the UPI wire, searching for stories of interest by keyword. One can communicate with similarly equipped colleagues through the medium of electronic mail which is delivered instantaneously. As these services grow and prosper, consumers will have information and communication resources at their fingertips which were unheard of just a few short years ago.

As we look at the four classes of personal computer owners (hobby, education, business, home), we can see that only the hobby market is likely to contain a significant fraction of people who might enjoy learning some arcane instructions to make their computers do their bidding. The remaining three classes of users want to use the computer as a tool - not as an end in itself.

The introduction of computers into new markets, such as the home, provides challenges for the designers to ''humanize'' the man-machine interface. This humanization process does not mean having the machines look like humanoid robots who attempt to converse in English, but it does mean having the machines respond to people in a way that is sufficiently natural to users that the computer becomes a transparent facilitator between the user and the goal of the interaction - be that playing a game, watering the lawn, looking at a stock portfolio, etc.

As nice as todav's personal computers are, we are still far from this goal. It is our strong conviction that those of us outside (or on the fringes) of the ''computer world'' can be most effective in designing and implementing these improved interfaces.

There are a few otherwise astute computer scientists who feel that all user interface problems can be solved primarily through improvements in software - who believe that the challenges exist not in the physical environment of the man-machine interface, but rather in the languages and programs which mediate the interaction between the user and the electrical impulses running around in the computer itself.

While there is some truth to this concept, we feel strongly that there are many challenges still present on the hardware side of the computer system -especially at the mechanical interface between computers and people. It is improvements that can be made in these mechanical interfaces that will increase the value of this technology to the average consumer. The skills needed for designing these improved interfaces are those with which many non-computer designers are already familiar - human factors, package design, linkages, electromechanical actuators, etc. Of course, one must also know how the personal computers are likely to be used. This should not be a problem. Since we all are potential users of personal computers anyway, designers are much more likely to understand the improvements that are

Letter Quality Word Processing for the PET®/CBM®

Com-plications™ C101 Printer Adapter

- Designed specifically for NEC 5530P Spinwriter
- Fully compatible with all "Word-Pro" software
- Drives any Centronics-compatible (parallel) printer
- Compatible with all PET®/CBM® peripherals
- Uses print statements— no assembly programs
- Industrial quality construction, 72-hour burn-in

Word-processing is the fastest growing segment of the micro-computer market. Small businesses, sales organizations, lawyers, physicians, publishers and a myriad of other potential users are prospects for Commodore PET®/CBM® computers and NEC "Spinwriters" with the new Com-plications™ C101 printer adapter. The C101 is manufactured by an industrial electronics company to industrial quality standards. Can your customers afford anything less? Call or send for a dealer package today.

PET® and CBM® are registered trademarks of Commodore Business Machines, Inc.

F.I. ELECTRONICS
968 Piner Road
Santa Rosa, Ca 95401
Telephone: 707-527-0410
Telex: 33-7769

www.commodore.ca

needed than would be the case if the computers were solely resident in the hands of highly trained computer specialists.

As an example of one interesting area, let us look at a portion of the man-machine interface which is accepted almost without a second thought - the keyboard.

The arrangement of the letters on most personal computer keyboards was developed in 1872 by typewriter inventor C. Latham Sholes and his attorney, James Densmore, to overcome a major problem in the design on Sholes' original typewriter. Originally the keys were arranged in alphabetical order. Unfortunately, this arrangement made it very easy to type keys in such quick succession that adjacent type elements would jam together before hitting the ribbon. To overcome this problem, Sholes and Densmore placed the most commonly used letters as far apart in the type basket as possible, and the result was the QWERTY keyboard we have today. (The name QWERTY is derived from the first five keys in the top alphabet row of this keyboard, and is a folksy name for the Sholes' arrangement.)

Once the original mechanical limitations of early typewriters were removed, there were some attempts to improve the keyboard layout as well. Most of this effort was directed towards improvement in typing speed and in reduction of operator fatigue. In 1932, after many years of work, August Dvorak suggested an alternative keyboard arrangement. As with the Sholes' keyboard, the DSK (for Dvorak Simplified Keyboard) requires a lot of training to use effectively. Its principal advantage for touch typists is that skilled DSK users can type at up to twice their previous typing speed with less effort than required to use the Sholes keyboard.

If DSK is so much better than Sholes, it is logical to ask why the improved keyboard has not displaced the older inefficient model, especially since the original mechanical limitations leading to QWERTY have been overcome for many years. There seem to be two causes for this failure. The first is the inertia associated with displacing the many millions of Sholes keyboards in use today. Second, there is understandable inertia on the part of typists trained on the Sholes arrangement, each of whom would have to spend about a month making the transition to a new and hard to obtain keyboard.

My view is that both the Sholes and the DSK arrangement are confusing to a novice user. As the personal computer market expands into schools and homes, an increasing number of people are being asked to type on a keyboard which makes no sense whatsoever. Anyone who has watched a child use the computer has seen the intense concentration with which he or she scans the keyboard looking for the right key. These novice "hunt and peck" typists typically use the index finger of one hand to do their typing, with "advanced" novices using the index fingers of both hands. When one considers the myriad applications for the computers used by novice typists, it seems almost criminal that a powerful modern tool like a personal computer should use such a poorly designed interface to receive information from the user.

The personal computer revolution brings with it a unique chance to improve the user interface. The reason for this is that, for the first time since 1872, a major new keyboard market has opened for which the purchasers and users of these keyboards are not already skilled typists. This massive market, coupled with the fact that the overwhelming majority of personal computer users are not "touch typists" is what gives encouragement to the concept of a new keyboard arrangement.

In thinking about new keyboard arrangements useful to novices, it is fairly tempting to arrange the keys in alphabetical order. Alphabetical key arrangements are found on several high-volume specialized computer-based products on the market today, including the Texas Instruments "Speak & Spell", and the Nixdorf and Craig "pocket translators". Even skilled typists must know the alphabetical sequence in order to look up words in the dictionary, file reports, etc. It is very encouraging to see the spacious keyboards used in the children's computers at Sesame Place (the new park in Pennsylvania designed by Children's Television Workshop and Busch Gardens). Each of the custom keyboards on the myriad Apple computers in the park has an alphabetical key arrangement. As Joyce Hakansson of CTW Parks says, 'We want to give the children access to the computers, not teach them how to type.'

With an alphabetic key arrangement the novice user does not have to scan the whole keyboard to find a given key. This is important since the foveal regions of a user's eyes can only be focused on one or two keys at a time. Once a key is perceived (assuming it is the wrong one), the logic behind the alphabetic keyboard layout helps to reduce the time required to find the desired letter or symbol. One could even use color coded keytops (with vowels having a different color than consonants, for example) which may be of special benefit to younger users.

The development costs associated with this keyboard are no different from those associated with the Sholes arrangement, so there is no particular reason for this new keyboard to be more expensive than one using the traditional key layout.

The important point here is that one's skills as a designer of equipment to be used by people can be applied to a technology area which was previously the domain of a small contingent of highly specialized people. In fact, if you are sufficiently motivated, you can convert almost any of the under $1,000 personal computers to any keyboard arrangement you want in an afternoon, without first having to become an expert in computer design. I know because I have done it.

www.commodore.ca

There is another solution to the keyboard issue, of course, and that is to eliminate the keyboard altogether. This is the approach taken in the design of the PrestoDigitizer (TM) tablet - a low-cost peripheral which allows personal computer users to communicate with their computers through hand printed characters.

The point of this example is that just as this device was developed outside the traditional computer science community, non-computer designers, can create other devices which will improve the ability of ordinary people to effectively use computers. Personal computers are made from much more than a handful of chips and a collection of software. Computers have displays, keyboards, cursor control devices, graphics input tablets, disk drives, printers, mechanical control devices such as robot arms, and myriad other electromechanical mechanisms which mediate the interaction of people with machines.

Any of these areas is ripe for new developments. These advances in technology will come from motivated people whose training spans a variety of disciplines.

The personal computer revolution is in its infancy. We can each contribute to making this a friendly revolution in which the computers will come to be tools for creative expression - tools which have been designed to respond to the needs and desires of people on people terms.

We should not settle for less. ©

DIAL-A-ROM™

for the Commodore PET/CBM

Switch between any one of six ROMs such as:
- **WORDPRO** from Professional Software Inc
- **VISICALC** from Personal Software Inc
- **TOOLKIT** from Palo Alto IC's
- **SORT** from Matric Software Inc

or switch charactersets with the Math and Foreign Language ROMs from West River Electronics (from KOBETEK in Canada).

In attractive cabinet, with dipplug to fit into 24-pin socket.

$88.00 (US)
(+ $5.00 airmail)

kobetek systems limited
RR #1 WOLFVILLE NOVA SCOTIA
CANADA B0P 1X0 (902)542-9100

Let your Commodore PET®/CBM® Talk to the World with...

Com-plications™ C232 IEEE-488/RS-232C Adapter

- Baud rate selectable from 75 to 9600 baud
- Selectable RS-232 or 20ma current loop operation
- Full modem controls for systems capability
- 16 character internal buffer to assure no data is lost
- Compatible with all IEEE-488 (1978) controllers
- Industrial-quality construction; 72 hour burn in

Now your PET®/CBM® can communicate with the outside world via RS-232. Send and receive data from CRT terminals, printers, modems, acoustic couplers, and other computers. The C232 features full adherence to the IEEE-488 (1978) and RS-232C standards. Both the IEEE device address and baud rate are selectable. Communication is accomplished using BASIC language PRINT statements. The unit is built in a metal case to exacting industrial standards, fully assembled and burned in.

PET® and CBM® are registered trademarks of Commodore Business Machines, Inc.

F.I. ELECTRONICS
968 Piner Road
Santa Rosa, Ca 95401
Telephone: 707-527-0410
Telex: 33-7769

www.commodore.ca

SBCS

SOFT-WARE for MAXIMUM PERFORMANCE WITH YOUR Apple II

GENERAL LEDGER

offers you

- *Virtually complete flexibility in formatting balance sheets and income statements.*
  - 31-character account names.
  - 6 digit numbers.
  - 10 levels of subtotals for more detailed income statements and balance sheets.
  - up to 9 departments.
- A cash journal that automatically calculates the proper off-setting entry and allows a 33-character transaction description.
- A balance sheet and income statement for the current month, quarter, or any of the previous three quarters.

We are the authorized center for Osborne/McGraw-Hill providing you with business packages that will do everything the Osborne Software will do in addition to many features we have added.

ACCOUNTS RECEIVABLE allows you to:

- Enter invoices at any time.
- Keep track of invoice amounts, shipping charges, and sales tax (automatically computed).
- Accumulate total payments including progress billing information on each invoice.
- Print reports which list unbilled invoices, unpaid invoices, and paid invoices.
- Obtain an aging analysis of unpaid invoices.
- Assign your own alphanumeric customer code.
- Maintain the date of the last activity for each customer, as well as amounts billed this year and last year.
- Print Customer Statements. (Statements available through SBCS).

Accounts Receivable is available independently or can be integrated with the General Ledger program.

In the final analysis, making your bookkeeping easier is what our software is all about. There is virtually no limit on entries since you may process them as often as you like. These packages will support any printer/interface combination.

General Ledger requires one hundred ten columns.
Accounts Receivable requires one hundred thirty columns.

Suggested Retail:
Individually $180.00
Together $330.00
McGraw-Hill manuals (required for documentation)..... $20.00 ea.
Available from your local Apple Dealer or contact SBCS.

YOU NEED EXPERIENCE WORKING FOR YOU!
Contact or write:

SMALL BUSINESS COMPUTER SYSTEMS
4140 Greenwood — Lincoln, Nebraska 68504 — (402) 467-1878

APPLE & PET

MAE

The Most Powerful Disk-Based Macro Assembler/Text Editor Available at ANY Price

Now includes the Simplified Text Processor (STP)

For 32K PET, disk 3.0 or 4.0 ROMS or 8032 (specify) — OR — 48K APPLE II or APPLE II + and DISK II

MAE FEATURES
— Control Files for Assembling Multiple named source files from disk
— Sorted Symbol table — Up to 31 chars./label
— 27 Commands, 26 Pseudo-ops, 39 Error Codes
— Macros, Conditional Assembly, and a new feature we developed called Interactive Assembly
— Relocatable Object Code
— String search and replace, move, copy, automatic line numbering, etc.

STP FEATURES
— 17 text processing macros
— Right and left justification
— Variable page lengths and widths
— Document size limited only by disk capacity
— Software lower case provision for APPLE II without lower case modification

ALSO INCLUDED
— Relocating Loader
— Sweet 16 macro library for APPLE and PET
— Machine Language macro library
— Sample files for Assembly and text processing
— Separate manuals for both APPLE and PET

PRICE
— MAE, STP, Relocating Loader, Library files, 50 page manual, diskette — $169.95

SEND FOR FREE DETAILED SPEC SHEET

EASTERN HOUSE SOFTWARE
3239 LINDA DRIVE
WINSTON-SALEM, N. C. 27106

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

Microphys is pleased to announce the release of a series of twelve programs designed for use in *introductory calculus courses* on both the high school and college levels.

The programs are intended for use with a Commodore PET/CBM microcomputer having at least 8K of storage. Each program is recorded on a C-10 cassette and is accompanied by simple descriptive instructions. The programs retail for $20 each. A diskette containing all 12 programs may be obtained at a cost of $180.

Each program generates a unique set of problems for each student. Answers may be generated so that the student may check his own work or these answers may be suppressed. The student then solves his set of problems away from the computer. When his work is completed, the student enters his code number and answers and the computer will then grade his work, displaying the answers to those questions which were incorrectly solved; a percent score and a brief comment reflecting an overall evaluation are also given.

Individual Program Content

PC726- Differentiation of Algebraic Functions
PC727- Maxima/Minima Problems: Part I
PC728- Maxima/Minima Problems: Part II
PC729- Relative Rate Problems: Part I
PC730- Relative Rate Problems: Part II
PC731- Integration of Algebraic Functions
PC732- Differentiation of Trigonometric Functions
PC733- Integration of Trigonometric Functions
PC734- Integration: Areas of Plane Figures
PC735- Integration: Volumes of Solids
PC736- Integration: Arc Lengths
PC737- Integration: Surface Areas of Solids

Note: All programs are available from your local computer dealer. They may also be obtained directly from Microphys.

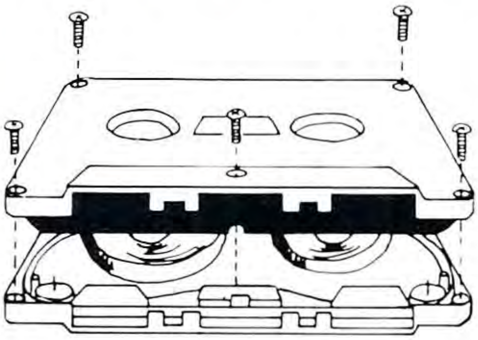
Educators: Be sure to write for our free educational software catalogue which describes over 140 programs for use in Chemistry, Physics, Math, English Vocabulary and Spelling. These programs are designed for use on the PET with a minimum of 8K of storage.

Dealer Inquires Invited

MICROPHYS PROGRAMS
2048 Ford St. Bklyn, N.Y. 11229
(212) 646-0140

www.commodore.ca

Don't lose your message because of the medium...



RAMWARE™

CASSETTES

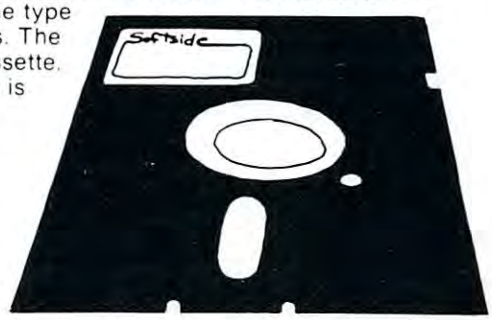
The cassette tapes used for recording data are composed of two parts: the cassette shell and the tape loaded into the shell. The shell can be either a 5-screw or sonic welded type with a non-magnetic leader or a magnetic leader (so called leaderless cassettes). The shell used in our cassettes is of premium quality. 5-screw, with non-magnetic leader. The choice of non-magnetic leader may confuse some people, but there is a valid reason. There is a splice required to connect the magnetic tape to the leader at both ends of the tape. A person recording program material or data, using a leaderless tape, stands to drop a bit of data at the splice point. Not all leaderless tapes have the splice and you have to be very careful when buying this type of data tape. We use standard leader to avoid the confusion, and unhappy customers when the first recording on the tape is always bad.

The tape used in our cassettes is of studio quality. The same type of tape is used by some studios for making master recordings. The magnetic tape used in the cassette is the true heart of the cassette. You can have the best shell made, but with low quality tape it is still junk.

The cassettes offered here have been chosen for the highest quality components consistent with a practical cost level.

Cassettes come packaged in boxes of 10. They are offered in 10 and 20 minute lengths.

C-10 .... $6.95 + $1
C-20 .... $7.95 + $1



DISKETTES

We offer two levels of diskettes: certified and non-certified. The certified diskettes have been put through a test to check the entire working surface for bad spots. These diskettes are certified error-free by the manufacturer. If you require assurance of every diskette being perfect, then the Dysan certified diskette is for you.

The BASF company invented magnetic tape from which the very large and varied industry of today has grown. We offer the BASF premium quality (non-certified) Diskette. These diskettes enjoy one of the lowest reject rates of any manufacturer (all our disk-based software is duplicated on BASF).

We are also offering diskettes from 3-M SCOTCH. These come encased in a touch (PVC) jacket which resists handling damages. They are certified 100% error-free. Their low modulation provides better signal stability.

BASF:
Box of 5 .... $19.95 + $1
Box of 10 .... $34.95 + $2
Box of 100 .... $299.00 + $3

3-M SCOTCH:
Box of 10 .... $39.95 + $2
DYSAN:
Box of 5 .... $29.95 + $1

The Software Exchange

6 South St., Milford, NH 03055

TO ORDER TOLL-FREE: (in NH call 673-5144)

1-800-258-1790

www.commodore.ca

Music and The Personal Computer

Frank Winter, Dean,
Instructional Development,
Sheridan College

Most people have no idea of the tremendous potential for music that is found in the personal computer. So that more people might become aware of this potential a Music Symposium was held last June at Sheridan College in Oakville, Ontario, Canada.

Sheridan College is one of 22 Community Colleges in the Province of Ontario and has been heavily involved in the educational use of microcomputers for over 2 years. At the present time we have more than 100 microcomputers in use! We had done some programming for our Performing Arts department so that they could teach music sight reading by using the powerful drill and practice techniques available on the microcomputers; their interest had been whetted and it seemed to be a good time to bring to their attention some of the other music software that was available on various computers.

For the sound enthusiasts, the audio system used to produce the music at this symposium was a Phillips preamplifier control, feeding four Motional Feedback speakers, each of which was rated at 60 watts:

Low Frequency 40 watts 35-1000 Hz
High Frequency 20 watts 400-20,000 Hz
(.2% distortion)

Approximately 125 people attended, and the program looked like this:

Hal Chamberlin
"Techniques for the Computer Performance of Music"
AIM Computer & M.T.U. High Density Graphics Board

Jim Butterfield
"First Steps in Making Music"
PET Computer

Dr. Frank Covitz & Dr. Cliff Ashcraft
"Advanced Examples in Making Music"
KIM Computer & M.T.U. High Density Graphics Board

Michael Bonnycastle
"A Canadian Music System"
PET Computer

Professor Colette Wilkins
"Teaching Music with the TRS-80"
Radio Shack TRS-80 Computer

Trudy Van Buskirk
"Music with the Apple Alf"
Apple Computer

Dr. Sterling Beckwith
"The Atari Music ROM Pack"
Atari Computer

Terry Garbutt
"The Visible Music System"
PET Computer

Linda Borry
"M.E.C.C. Music"
Apple Computer

Dr. Sterling Beckwith
"The Human Interface to Computer Music"

Hal Chamberlin is Vice President of Research & Development for Micro Technology Unlimited, a firm which builds innumerable 6502 accessories. His first computer, built for music synthesis, was constructed from scrap IBM logic boards and discrete transistors as early as 1970 and, in my mind at least, Hal Chamberlin has to be the "FATHER" of music on microcomputers. His original article for Byte Magazine, "A Sampling of Techniques for Computer Performance of Music", was published in September, 1977. This article was followed by another, in the April 1980 issue of the same magazine, entitled "Advanced Real-Time Music Synthesis Techniques".

Hal's presentation was a masterpiece, as he was able to cover a complex topic in such a way as to make everyone feel they fully understood what he was talking about. He was clear and concise and his use of visuals on the overhead projector fully explained his concepts. When describing how a wave form was developed he used the High Resolution Graphics Board developed by Micro Technology Unlimited and, since it is addressable to 64,000 pixels, he was able to be very precise in his visual information.

I will not attempt to paraphrase his talk; interested persons would be far better off to obtain reprints of his articles in order to cover the topic in sufficient detail. However, his presentation established a high tone which highlighted his position as the keynote speaker.

Jim Butterfield was asked to explain just how easy it is to produce music on a microcomputer. Since Hal had earlier established how it was done technically, it was up to Jim to demonstrate exactly how it was done on the microcomputer. He explained simple note producing routines and illustratred how 2 part, 3 part, and even 4 part harmony is produced. He then showed that it was possible to speed up a piece to double tempo (without changing pitch) and also to change the various tones of each voice (once again, without changing the pitch). Jim's extensive knowledge of computers and computer music, together with his sense of humor, made his talk a delightful experience.

Dr. Frank Covitz and **Dr. Cliff Ashcraft** dealt with an advanced music system, and their presentation gave a magnificient overview of the "State of the Art" as it applies to computer music. Dr. Covitz

www.commodore.ca

L TO R
Philip Covitz, Jim Butterfield, Brian Walsh, Francis Ament

L TO R
Hal Chamberlin and
Karl J. Hildon, Technical Advisor, Commodore Business Machines

L TO R
Cliff Ashcraft and Dr. Frank Covitz
Getting Ready

Jim Butterfield
Holding Court Just At Lunch Break

mentioned that, although their music is based on Hal Chamberlin's system, it is different in that it has a more elaborate command system; it creates wave forms faster and employs a different technique in play routines called "Time Division Multiplexing". They have also utilized a volume control digital to analog converter and play many of their pieces in stereo.

Their demonstration employed many techniques and a great number of music pieces. At one point, in order to show that the music was not terribly hard to code, Dr. Covitz introduced his 12 year old son Phillip, who proceeded to play the Theme from Dr. Zhivago, "Somewhere My Love", which he had encoded, in its entirety, himself.

A great variety of instruments were reproduced on the computer, among them such standards as the cello, clarinet, mandolin and banjo. In addition, they

www.commodore.ca

were able to play instruments which haven't even been invented yet, such as the BLITHER, GIANT RUBBER BAND, 32 FT. PIANO, and GLOCKENFLUTE. Needless to say, it provided a real insight into the possibilities of computer music and the shape of things to come.

Not only did they provide a great deal of technical information, but they did it in such a way as to make it easy to understand, while at the same time providing great entertainment. All they need is an agent and they could take their show on the road!

Michael Bonnycastle is an engineer and computer hobbyist who had seen Hal Chamberlin's music system and had been intrigued by the wonderful sounds he could create; however, there was one major problem and that was that it took so much time to look up and load the various note tables. In order to overcome this problem he reasoned that you might as well use the computer to do the table look-up and loading; basically this is what his system does. In other words, he has constructed a human interface which makes Chamberlin's music system easier to use.

Notes are entered as simple dots, which may be played back, or changed. It is possible to play any four notes at one time and you can alter the duration and tempo as necessary. By using repeats he eliminates repetitious coding and achieves quite a remarkable flexibility in this simple system. Although this program does not have the graphics that are found on other programs it does a very efficient job and allows a person to create very innovative music.

Colette Wilkins is Professor of Music at Carnegie-Mellon University in Pittsburgh. She has a rather difficult task there in that, although most of her students have a good background in music, they differ markedly in the various elements of their knowledge.

Professor Wilkins teaches the "solfege" method of voice training and musical notation and this is a course which all music students are required to take. The solfege system is used extensively in Europe and uses a fixed "DO" and a pure tone related structure. As she says: "The students must learn in two years what took me ten years to learn in France"; and this is where the drill and practice capabilities of the computer come in handy.

Although Professor Wilkins had little or no knowledge of computers she worked with engineers to develop a music box for the Radio Shack computer and they, in turn, programmed her lessons and drills as she laid them out. The results have been very encouraging and she feels that without her computer programs she would be unlikely to advance as quickly as she does now. Further details of her system may be found in the article entitled "Computer Aided Sight Reading" which was published in the June 1980 issue of Creative Computing.

Trudy Van Buskirk is an elementary school teacher in Special Education and she feels that the Apple Alf is a good package for the person who is not necessarily a musician but is interested in music, and also for someone who is not necessarily a computer buff but is interested in learning how to make a computer play music.

The Alf requires a board which is plugged into slot 4 and may be either disc or cassette based. It will play 3 voices with one board, 6 voices with two boards and up to 9 voices with three boards. In addition it contains an excellent introduction which not only explains, but also demonstrates, music synthesis. The program has a simple entry system with all necessary controls for tempo, duration, correction, repeats, etc. In addition it visibly demonstrates the musical notation in full color on the screen, not only upon entry, but also while the music is playing. This system impressed Trudy a great deal because she was able to produce quite pleasant music with a minimum of effort and study. This package can be purchased right off the shelf and does not require any customization before use.

Dr. Sterling Beckwith has been working with computer aided music since 1965, and was responsible for the York University "Interactive Music Project". While he has worked mainly with mainframes, he was involved in helping Atari with their Music Composer. We thought it would be appropriate to have him demonstrate this unit since he had been involved from a very early date.

The Music Composer is geared to the non-technical computer user and also to the person who is not necessarily a competent musician. It uses a variety of menus which allow the user to play up to 4 voices, transpose pieces of music, edit music, provide a music display, arrange music, save or retrieve music, and also record file formats.

One of its main advantages is that the user can just plug in the cartridge, turn on the power, and start right to work!

Throughout his demonstration Dr. Beckwith reiterated his feeling that the general public had to demand from the software producers the type of program that the serious music student/teacher desires and needs.

Later in the program Dr. Beckwith was given the opportunity to discuss the human interface to computer music and also to discuss his concerns about the quality of software which is being produced. He chaired a discussion in which the attendees were given the opportunity to talk about their problems in the computer music field. Dr. Beckwith struck a responsive cord with the audience, who seemed, in many cases, to share his views. The lively discussion left the audience with some very interesting thoughts to take home with them.

Terry Garbutt attended both the University of Toronto and York University and is a former KIM user. He later moved on to the PET but, regardless of the hardware, he is an incurable music synthesist.

Terry demonstrated the Visible Music Monitor

www.commodore.ca

from A B Computers and he seemed to be quite taken with the flexibility and comprehensiveness of this system. It is written in 6502 machine language and allows you to enter notes directly from the PET keyboard. With this program you can insert, delete, or move notes up or down on the staff, use the "record changer" mode to load a number of songs, have a user defined keyboard, complete tempo flexibility, transpose capability, and waveform modification capability. Music can be played either with or without note display.

This package too may be used right off the shelf and, with its lively graphics, it makes a unit which will be attractive to many people.

Linda Borry is an Instructional Co-ordinator with the Metro Minnesota Schools and because she has some technical computer background, a musical background, and is also an educator, she was able to incorporate all these skills in designing and programming a series of educational units on music which can be used in the school system. The programs run on an Apple II with Applesoft Basic in ROM and use only components found in the original equipment (there are no add-ons). This is to make it easy for the ordinary classroom teacher to use the programs without having to go out and buy extra components.

Some of the design features of these programs are as follows: short student responses requiring accuracy, always displays the right answer, flags student weaknesses, complete student control as to difficulty and number of problems, menu driven and geared to students who have never used a computer before. These programs make use of three major strengths of microcomputers--to generate random questions, to inexhaustively perform drill and practice routines, and to keep track of the students.

There are approximately 18 programs in the series and some of the topics covered are Key Signatures, Rhythm, Intervals, Dictation Drill, and Scale Identification. It is extremely rare to see programs forming a series which demonstrates as much care and skill in its design as these do. They are educationally sound and really seem to care about the students' progress. Linda mentioned that Apple will probably be marketing the series in September or October, 1980.

Although Louis C. Cargile Jr. could not come to the symposium because of previous commitments, he sent along some of his music, which included Blues, Disco, Jazz, Ballads, Ragtime, Sousa, Bach, Duke Elligton, and some experimental music which includes volume control through software. Lou has unique arrangements which are set up in a jukebox style so that you can call up the piece you want to hear from a menu presented on the screen.

It was quite a revelation to see all the systems which were presented and to listen to the high quality of music played. It is obvious that this large group of interested computer musicians represent only the tip of a very large iceberg concerned with this topic.

If you wish to correspond with any of the above-mentioned speakers you will find their names and addresses listed below:

Hal Chamberlin
Micro Technology Unlimited
P.O. Box 12106
Raleigh, North Carolina 27605

Jim Butterfield
c/o CN Telecommunications
151 Front Street West
Toronto, Ontario M5J 1G1

Dr. Frank Covitz
Deer Hill Road
Lebanon, New Jersey 08833

Michael Bonnycastle
81 Wychwood Park
Toronto, Ontario M6G 2V5

Professor Colette Wilkins
Carnegie Mellon University
Music Department
Schenley Park
Pittsburg, Pennsylvania 15213

Trudy Van Buskirk
267 Bain Avenue
Toronto, Ontario M4K 1G2

Dr. Sterling Beckwith
P.O. Box 244
North Salem, New York 10560
or
605 Finch Avenue West
Willowdale, Ontario

Terry Garbutt
3557 Ash Row Crescent
Mississauga, Ontario L5L 1K3

Linda Borry
Instructional Coordinator
M.E.C.C.
2520 Broadway Drive
St. Paul, Minnesota 55113

©

PET • APPLE • TRS-80
DATA BASE SOFTWARE

Data Handler Package $25.00
Files: Create, Read, Write Print, Merge, View
Records: Add, Delete, Update
Specials: Sorting by fields, Mass Updating

Advanced Data Handler $38.50
Includes following compatible systems:
- Advanced version of Data Handler
- SEARCH - Data file scanning and subfile making system
- LABEL - Quality label making system
- ADD - Data file field summation system

Excellent for home, office, and professional uses! Both products include floppy disk, documentation, and example applications.
For more information write to:
Business Computer Services Co.
9020 Eby
Overland Park, KS 66212

www.commodore.ca

Micros With The Handicapped

Susan Semancik

The final testing was about to begin. This would ultimately determine the success of our months of work in preparing the package of programs using the Manual Alphabet Tutorial on the PET computer (See COMPUTE Issue #3). Who could better judge the usefulness and effectiveness of a computerized signing program than the coordinators and students involved in a summer session to teach handicapped honor high school students from all over the country about marine biology?

Members of the Delmarva Computer Club had been at the Marine Science Center the summer before when the signers there had helped to review the original drawings of the handshapes used to form the letters of the alphabet for the deaf on the PET computer's screen. But, this would be the signers' first opportunity to see the signs incorporated into computerized teaching programs and to evaluate the finished product. Dr. Ed Keller, director of the handicapped program, invited the Club to participate in their communication's workshop. One of the goals of the workshop was to help the students who were non-signers to communicate with those who were signers. This was no small feat when you consider the students' handicaps included blindness, deafness, orthopedic handicaps, and in some cases, multiple handicaps.

So, it was with a mixture of anticipation and reservation that we carried our equipment and programs into the workshop on that summer evening. As can be seen in the pictures, handicapped students, Club members and their families, all enjoyed the variety of computer programs demonstrated during the workshop. We were especially pleased with the students' responses to the programs in the Manual Alphabet Tutorial package, which are designed to teach and strengthen fingerspelling skills from several different approaches: learning all 26 signs together, learning them in groups of five with cumulative or separate tests, learning them in groups of user determined size and starting letter, learning them in groups of five with cumulative or separate tests, learning them in groups of five with cumulative or separate tests, learning them in groups of user determined letter combinations, and recognizing the fingerspelling of words at user determined rates.

Different students had preferences for different approaches and the variety available seemed to satisfy the needs of the students at the workshop. We knew our programs would not be able to be used by all students at the Center. For instance, the blind would not be able to see the signs on the PET's screen, and those with crippled hands would not be able to form the signs. None of the Club members participating in the demonstration could believe the enthusiasm, comraderie and concentration exhibited by the students in either learning to sign the alphabet or helping those who were just starting to learn. Even those students who have been signing for many years enjoyed seeing how fast they could make the computer fingerspell words while they still maintained comprehension. One young man attained an almost unbelievable speed where we could no longer distinguish the individual signs and yet he was able to identify the words perfectly and typed them confidently into the computer. It was a joy for all of us to see such positive reactions! Even those instructors with doubts as to how fingerspelling could flow naturally enough from a computer for rapid recognition soon had no reservations in the face of such apparent success.

This completed package of seven programs using the Manual Alphabet Tutorial on either an Old or New ROM 8K PET is also being tested by independent groups who will be reporting on the reactions of students and clients in their own applications of the programs.

We welcome any suggestions for improvements or expansion of the programs and will update free anyone's copy of the program who is first to suggest changes that we implement in future editions. All inquiries and suggestions should be made through Jean Trafford, Secretary, P.O. Box 36, Wallops Island, VA 23337.

The Delmarva Computer Club is marketing the complete package for $49.95, with proceeds being used to fund other Club projects. The package also includes documentation, a sample word file to be used with the fingerspelling program, and a DATA MAKER program that enables people without programming experience to easily create their own word files.

www.commodore.ca

C. Marshall Curtis, program chairman for the Delmarva Club, helped to coordinate these activities. He provides some personal impressions and observations:

As we drove in to the Marine Science Center which is located next door to NASA, Wallops Flight Center, I was more than a little apprehensive. I'd heard that each summer they hold classes for handicapped students. Of course I'd seen blind students, deaf students, and students in wheelchairs and on crutches, but all in one classroom, never! "I'll have to be careful", I thought. "In any case, Susan has been here before and she'll know what to do."

What to do first was to unload our CBM PET and some peripheral equipment and start setting it up. About this time, Bob Hinds, executive director of the Marine Science Center and treasurer of the Delmarva Computer Club, walked in with his PET. Susan loaded the Manual Alphabet Tutorial in both PETs and the "kids" gathered around full of curiosity and questions. David, who is blind, stuck his hand in a box and announced, "Here are some cassette tapes for your computer!"

Soon some of the deaf students were passjng the test at the end of the tutorial. A deaf boy who had never learned "finger spelling" and a student in a wheelchair were busy looking first at the signs on the PET screen and then forming the same letters with their own fingers.

What to do next was sit back in amazement as I began to see deaf students forming letters on the hands of blind students and the blind kids signing back to them. What I had not expected was the way they would be helping each other, their obvious sense of humor, and their eagerness to learn about computer hardware and programming as well as the manual alphabet and marine science.

By the time we were setting up for our third visit with the physically handicapped students at the Marine Science Center I felt quite at home. I also had learned that the official name is The Marine Science Consortium Inc. (TMSC). It's a non-profit corporation which was incorporated in 1971. The member institutions are twelve state colleges in Pennsylvania, American University and Catholic University in Washington, D.C., and West Virginia University. The Consortium's facilities include classrooms, laboratories, dormitories, cafeteria, library, recreation rooms, and two research vessels.

Dr. Ed Keller was there, as usual, and I noticed that he had a special way with the students. This time, a wheelchair bandit made off with Dr. Keller's crutch. In addition to being a friend to the students, Dr. Keller is professor of biology at West Virginia University and director of this summer program in marine science for outstanding handicapped students. Each year during the month of July, 11th and 12th grade students meet to learn from each other, from college instructors, and from field trips and laboratory sessions. They study marsh and beach ecology, shipboard techniques, physical oceanography, marine biology, etc.

Most of these kids are more familiar with the mountains, plains, and cities of the United States than with salt marshes and salt water. Here they have a chance to explore the marshes, bays, sand dunes and beaches around Wallops Island, Chincoteague Island, and the Assateague Island National Seashore Park.

Yes, these kids are physically handicapped, but they have a lot of fun and learn a lot. I'm sure those of us who had an opportunity to interact with them enjoyed it and learned much from them, too.

©

1,000,000 Bytes

PET/BETA-1

THE FLOPPY DISK ALTERNATIVE

FLEXIBLE DATA MANAGEMENT . . . MICRO-PROCESSOR CONTROLLED BETA-1 UNIVERSAL TAPE DRIVE BY MECA* . . . PERIPHERAL DEVICE WITH COMPLETE SOFTWARE SUPPORT

The PET/BETA-1 digital tape system provides all the features of a disk, with powerful data handling capabilities. Your PET/CBM handles big jobs with a data capacity of one megabyte per drive, fast seek times, and 1k per second data transfer rates. Put your records on line with PET/BETA-1.

BETA-1 DRIVE, PET/BETA-1 OPERATING SYSTEM MANUAL $700.00
MANUAL (applicable to purchase) $10.00

PET/CBM Software

Terminal

Turn your PET into an intelligent terminal with one of our terminal packages. These are complete assembled hardware and software packages. All include line editing/resend, repeat key, shift lock, output to CBM printer, and more . . . Delivered on PET cassette with manuals. Inquire for modem prices.

PETTERM I	All features above	$75.00
PETTERM II	All features of I, plus local text editor with down-loading capability	$90.00
PETTERM III	All features of II, plus 80/132 column scrolling window for viewing formatted outputs wider than 40 columns.	$100.00

FORTH

Interactive high level compiler and operating system 5-10 times faster than PET BASIC. High level block structured language. A Fig Forth. See August 1980 BYTE—featuring FORTH $60.00
With interactive assembler providing high level logic constructs and macro capability $90.00

Games

Fast paced multiplayer games with single player mode. Delivered on PET cassette.
Each $15.00

NUCLEAR WAR	Nuclear confrontation on a global scale. Many scenarios.
GALAXY	Pillage a 3-D galaxy collecting loot from captive worlds.
STOCK MARKET	Rags to riches game of buy and sell with computer stocks.
ENCRYPT	Challenging puzzles, decipher coded messages.
ALIEN	Protect the federation. Real time 3-D navigation.

F S S
software for small computers

1903 Rio Grande
Austin, Texas
78705

1-512-477-2207

P.O. Box 8403
Austin, Texas
78712

* FSS IS AN AUTHORIZED MECA INC. DEALER
**TEXAS RESIDENTS ADD 5% SALES TAX
PET IS A REGISTERED TRADEMARK OF COMMODORE INC.

www.commodore.ca

Small Computers and Small Libraries

Arthur L. McNeil
Seattle University

It would appear that many tasks that must be carried out in small libraries could be done quickly with small computers. Since these machines are already available in many high schools, junior and small four year colleges, their use could cut down on the expenditure of scarce dollars. Of all the tasks which can be undertaken by computers, the typing of catalogue cards lends itself most readily to computer use because of the repetitious information found on each of these cards.

Since most small libraries do not need a system which involves the use of complex cards like those issued by the Library of Congress or various library networks and since the use of these systems requires the outlay of unavailable dollars, it would seem reasonable to have computers print catalogue cards which follow most of the recommended library procedures without all the unneeded information found in the more involved cards.

The task of typing out multiple file cards is monotonous, to say the least, and is expensive in the time thus expended by librarians. In contrast, multiple file cards can be printed very quickly using a small computer and a printer. The author has developed a program, written in Basic, which will do this print-out of main entry card, shelf list card, author card, subject cards, etc., etc., in quick order. PET personal computer and a TTY Model 43 printer were used to accomplish this task. Of course, other computers and other printers can accomplish the same job. The program is listed in Figure 1, and with few changes, can be used by most other systems. This program will print-out all the catalogue cards needed by these small libraries.

Figure 2 shows the print-out of this program.

Explanation of the Program

Statement numbers (ST. NO'S) 100 to 450 input data into the program and assign variables to each item.

ST. NO'S 460 to 500 open the file and print the first card, the author card which is also the main entry card. Notice the use of the subroutine which prints the data common to all cards. The author card is configured as follows:

line 1: space
line 2: space
line 3: The classification number and the principal author's name.
line 4: The author designation.
line 5: space
line 6: The author's name and joint author(s), if any.
line 7: The title of the book.
line 8: City of publication, publishers, date of publication.
line 9: space
line 10: Collation information.
line 11: Notes
line 12: space
line 13: space
line 14: Subject heading 1, 2 and 3.
line 15: Title, joint author(s) and series, if any.
line 16: space
line 17: space
line 18: space

It should be noted here that if a certain piece of information is not pertinent to this book, e.g. extra subject headings, joint authors or series, etc., etc., a number sign (#) is typed instead of the data requested. This procedure will print a blank or print an empty line and so preserve the eighteen line format. Since the size of library cards is three by five inches and since the printer is set to print six lines per inch, an eighteen line format will fill the first card and be ready to immediately start the second card. The author would advise the use of tractor library cards which come in a roll or fan fold where one card immediately follows another and so printing can be continuous. A number of library supply firms merchandise these cards; two are listed below:

Gaylord Brother Inc.
P.O. Box 8489
Stockton, CA 95208

Josten's
Library Service Division
1301 Cliff Rd.
Burnsville, MN 55337

ST. NO'S 510 to 530 print the shelf list card which is a duplicate of the author card.

ST. NO'S 540 to 620 print the title card. This part of the program is divided into two parts. The choice depends on the length of the title. If the title contains less than 51 characters, ST. NO'S 560 to 580 print the card. If the title exceeds 50 characters, ST. NO'S 590 to 620 do the job.

ST. NO'S 630 to 830 print succeeding cards, i.e.; subject headings, joint author(s) and serial cards, if these apply.

ST. NO'S 840 to 1260 is the subroutine which prints the data common to all cards, namely that

www.commodore.ca

Skyles Electric Works

Q. Your students are gathering around the several PET computers in your classroom. And they all are hungry for hands-on turns at the keyboards. Some students are just beginning to understand computers; others are so advanced they can help you clean up the programs at the end of the period. How do you set up a job queue, how do you keep the beginners from crashing a program, how do you let the advanced students have full access? And how do you preserve your sanity while all this is going on?

A. With the Regent.

Q. What is the Regent?

A. The ultimate in classroom multiple PET systems. A surprisingly inexpensive, simple, effective way to have students at all levels of computer capability work and learn on a system with up to 15 PETs while the instructor has complete control and receives individual progress reports.

Up to 15 PETs, one dual disk drive and as many as five printers can interface with the Regent, and do all those good things we promised. It's designed to operate with 8K, 16K, 32K PET/CBM models and with the Commodore disk drives and new DOS.

Five levels of user privilege, from the Systems Level, through Levels One and Two, Student; Levels One and Two, Operator. From only the use of system commands to complete control for the exclusive use of the instructor.

There's complete system protection against the novice user crashing the program; the instructor has total control over, and receives reports concerning, usage of all PETs.

A complete set of explanations for all user commands is stored on the disk for instant access by all users. And a printout of the record of all usage of Regent is available at the instructor's command.

The Regent includes a systems disk with 100,000-plus bytes for program storage, a ROM program module, together with a Proctor and a SUB-it . . . and complete instructor and student user manuals.

Q. SUB-it? Proctor? What are they?

A. The SUB-it is a single ROM chip (on an interface board in the case of the original 2001-8 models) that allows up to 15 PETs to be connected to a common disk via the standard PET-IEEE cables. The Commodore 2040, 2050 or 8050 dual disks and a printer may be used.

(The SUB-it has no system software or hardware to supervise access to the IEEE bus. The system is thus unprotected from user-created problems. Any user—even a rank novice—has full access to all commands and to the disk and bus. This situation can, of course be corrected partially by the Proctor, completely by the Regent.)

The SUB-it prevents inadvertant disruption when one unit in a system is loading and another is being used.

The Proctor takes charge of the bus and resolves multiple user conflicts. Each student can load down from the same disk but cannot inadvertently load to or wipe out the disk. Good for computer aided instruction and for library applications, offering hundreds of programs to beginning computer users.

A combination of hardware and software protects the disk from unexpected erasures and settles IEEE bus usage conflicts. Only the instructor or a delegate can send programs to the disk. Yet all the PETs in the system have access to all disk programs. Available for all PET/CBM models. SUB-it and PET intercontrol module and DLW (down-loading software) are included.

Q. How expensive are these classroom miracles?

A. We think the word is **inexpensive**. The **Regent** system is **$250** for the first PET; **$150** for each additional PET in the system. The **SUB-it** is **$40**. (Add an interface board at **$22.50** if the PET is an original 2001-8.) And the **Proctor** is $95.

There are cables available, too: 1 meter at $40 each; 2 meter, $60 each; 4 meter, $90 each.

Phone or write for information. We'll be delighted to answer any questions and to send you the complete information package.

Skyles Electric Works

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

www.commodore.ca

which is contained in lines 4 to 18 of the described library card.

ST. NO 1250 must be explained. Here five empty lines are generated; three of these finish printing one card and the next two provide for two empty lines at the beginning of the next card. It takes about one minute to enter the data needed by the program and another minute to print up to eight library cards.

Even if a small computer is not available for library use, it might be worthwhile to consider the purchase of one of these machines together with a printer and cassette recorder in order to take advantage of the possibilities which these could provide. Some of those advantages could be not only the printing of library cards, but also the keeping of various listings. Some of these listings would be as follows: a list of books ordered, by author and title, date of the order, price, publisher, bookseller, department ordering, etc., etc. Likewise, a list of periodicals, containing their names, the completed volumes, recording of year and volume and the individual numbers of current year. Another list could contain a listing of books received by the library but not yet accessioned; this list could be alphabetized both by author and title for ready use. All these lists could be stored on magnetic tape and updated at the end of each week. Copies of these lists could be distributed to those individuals whose work would be made easier by the use of such lists.

In closing, a few bits of information might be useful. The small computer, the printer and tape deck will cost somewhere between fifteen hundred and twenty-five hundred dollars, depending on the brands of machines purchased. It might also be useful for the librarian to learn a computer language, preferably BASIC, so that programs could be written or modified to fit the need of the situation.

Only the ingenuity of the librarian can limit the computer use in libraries. Tasks which must be done over and over again are "duck soup" for the computer and they can save great amounts of librarians' time. The librarians, even in small libraries, should not be afraid of computers and their use. They are the wave of the future!

```
100 POKE 59468,14
110 REM A PROGRAM TO PRINT LIBRARY CARDS
120 REM BY ARTHUR L. MCNEIL, SEATTLE ¬
      ¬UNIV. SEATTLE, WA.
130 REM STATEMENT NO'S 140 TO 400 ¬
      ¬INPUTS DATA FOR CARDS
140 PRINT"TYPE CLASSIFICATION NO. OF ¬
      ¬BOOK"
150 INPUT C:PRINT
160 PRINT"TYPE AUTHOR DISIGNATION"
170 INPUT G$:PRINT
180 PRINT"TYPE NAME OF PRINCIPLE AUTHOR"
190 INPUT A$:PRINT
200 PRINT"TYPE TITLE OF BOOK"
210 INPUT T$:PRINT
220 PRINT"TYPE CITY OF PUBLICATION,
      ¬PUBLISHERS,DATE OF PUBLICATION"
230 INPUT D$:PRINT
240 PRINT"TYPE COLLATION INFORMATION"
250 INPUT C$:PRINT
260 PRINT"IF THE REQESTED DATA IS NOT ¬
      ¬USED FOR"
270 PRINT"THIS BOOK, E.G. SOME SUBJECT ¬
      ¬HEADING OR"
280 PRINT"NOTES ETC. A NUMBER SIGN (#) ¬
      ¬IS TYPED"
290 PRINT"AND THE PROGRAM WILL STILL ¬
      ¬PRINT "
300 PRINT"THE REQUIRED EIGHTEEN LINES ¬
      ¬PER CARD."
310 PRINT:PRINT
320 PRINT"TYPE NOTE, IF ANY"
330 INPUT N$:PRINT
340 PRINT:PRINT:PRINT"TYPE 1ST SUBJECT ¬
      ¬ENTRY"
350 INPUT E1$:PRINT
360 PRINT"TYPE 2ND SUBJECT ENTRY"
370 INPUT E2$:PRINT
380 PRINT"TYPE 3RD SUBJECT ENTRY"
390 INPUT E3$:PRINT
400 PRINT"TYPE JOINT AUTHOR(S), IF ANY"
410 INPUT E5$:PRINT
420 PRINT"TYPE NAME OF SERIES,IF ANY"
430 INPUT E6$:PRINT
440 E4$="TITLE"
450 I=LEN(T$)
460 OPEN 1,5:CMD 1
470 PRINT:PRINT
480 REM PRINTS AUTHOR CARD
490 PRINT TAB(1)C;TAB(10)A$
500 GOSUB 850
510 REM PRINTS SHELF LIST CARD(AUTHOR)
520 PRINT TAB(1)C;TAB(10)A$
530 GOSUB 850
540 REM PRINTS TITLE CARD
550 IF I>50 THEN 590
560 PRINT TAB(1)C;TAB(10)T$
570 GOSUB 850
580 GOTO 640
590 PRINT TAB(1)C;TAB(10)MID$(T$,1,50)
600 PRINT TAB(10)MID$(T$,51)
610 GOSUB 850
620 GOTO 640
630 REM PRINTS 1ST SUBJECT CARD
640 IF E1$="#" THEN 760
650 PRINT TAB(1)C;TAB(10)E1$
660 GOSUB 850
670 REM PRINTS 2ND SUBJECT CARD
680 IF E2$="#" THEN 760
690 PRINT TAB(1)C;TAB(10)E2$
700 GOSUB 850
710 REM PRINTS 3RD SUBJECT CARD
720 IF E3$="#" THEN 760
730 PRINT TAB(1)C;TAB(10)E3$
740 GOSUB 850
750 REM PRINTS JOINT AUTHOR CARD
760 IF E5$="#"  THEN 800
770 PRINT TAB(1)C;TAB(10)E5$
780 GOSUB 850
790 REM PRINTS SERIAL CARD, IF ANY
800 IF E6$="#" THEN 1270
810 PRINT TAB(1)C;TAB(10)E6$
```

www.commodore.ca

The COMPUTING TEACHER

Computers in Teacher Education **Computers in Junior High School**

The Computing Teacher is a journal for people interested in the instructional use of computers at the precollege level. It deals with teaching using computers, teaching about computers, and impact of computers upon the general curriculum. The journal also carries articles about uses of calculators in education.

The Computing Teacher began publication in May 1974 as the Oregon Computing Teacher.

Computers in High School **Computers in Elementary Education**

For the academic year 1980-1981, seven issues of The Computing Teacher will be published. A typical issue is 64 pages in length. It contains material of interest to teachers at all levels, with main emphasis at the precollege level. The journal carries paid advertising, which helps us keep the subscription price down.

Subscription Rates:

U.S.	Foreign
$10 (7 issues)	$15
$20 (16 issues)	$31
$27 (25 issues)	$44

Name ______________________

Address ______________________

____

______________ Zip __________

Send payment to:

The Computing Teacher
% Computing Center
Eastern Oregon State College
La Grande, Oregon 97850

www.commodore.ca

```
820 GOSUB 850
830 GOTO 1270
840 REM SUBROUTINE TO PRINT DATA COMMON ¬
      ¬TO ALL CARDS
850 PRINT TAB(2)G$:PRINT
860 IF E5$="#" THEN 890
870 PRINT TAB(10) A$;"    ";E5$
880 GOTO 900
890 PRINT TAB(10) A$;"    "
900 IF I>50 THEN 930
910 PRINT TAB(10)T$
920 GOTO 970
930 PRINT" ";TAB(10) MID$(T$,1,50)
940 PRINT TAB(10) MID$(T$,51)
950 PRINT TAB(10) D$
960 GOTO 1000
970 PRINT TAB(10) D$
980 PRINT:PRINT TAB(10)C$
990 GOTO 1010
1000 PRINT TAB(10) C$
1010 IF N$="#" THEN 1040
1020 PRINT TAB(10) N$
1030 GOTO 1060
1040 PRINT
1050 GOTO 1060
1060 PRINT:PRINT
1070 PRINT TAB(5)"1-";E1$;"  ";
1080 IF E2$="#" THEN 1120
1090 PRINT"2-";E2$;" ";
1100 GOTO 1120
1110 PRINT" "
1120 IF E3$="#" THEN 1150
1130 PRINT"3-";E3$
1140 GOTO 1160
1150 PRINT
1160 PRINT TAB(5)"I-";E4$;" ";
1170 IF E5$="#" THEN 1210
1180 PRINT "II-";E5$;" ";
1190 GOTO 1210
1200 PRINT
1210 IF E6$="#" THEN 1240
1220 PRINT "III-";E6$
1230 GOTO 1250
1240 PRINT
1250 PRINT:PRINT:PRINT:PRINT:PRINT
1260 RETURN
1270 CLOSE 1:END
```

```
523.2          Smith Ronald Y
S369

        Smith Ronald Y    Jones James P
        NOW IS THE HOUR
        New York-Jensen-1980

        P. 963 Illus
        U.S. Constitutiion-Bibliography


    1-U.S. History  2-U.S. Constitution 3-George Washington
    I-TITLE II-Jones James P
```

```
523.2          Smith Ronald Y
S369

        Smith Ronald Y    Jones James P
        NOW IS THE HOUR
        New York-Jensen-1980

        P. 963 Illus
        U.S. Constitutiion-Bibliography


    1-U.S. History  2-U.S. Constitution 3-George Washington
    I-TITLE II-Jones James P
```

```
523.2          NOW IS THE HOUR
S369

        Smith Ronald Y    Jones James P
        NOW IS THE HOUR
        New York-Jensen-1980

        P. 963 Illus
        U.S. Constitutiion-Bibliography


    1-U.S. History  2-U.S. Constitution 3-George Washington
    I-TITLE II-Jones James P
```

```
523.2          U.S. History
S369

        Smith Ronald Y    Jones James P
        NOW IS THE HOUR
        New York-Jensen-1980

        P. 963 Illus
        U.S. Constitutiion-Bibliography
    1-U.S. History  2-U.S. Constitution 3-George Washington
    I-TITLE II-Jones James P
```

PET and APPLE II Users

PASCAL

ABACUS Software makes available its version of TINY PASCAL for the users of two of the most popular personal computers.

TINY PASCAL is a subset of the standard PASCAL as defined by Jensen and Wirth. It includes the structured programming features: IF-THEN-ELSE, REPEAT-UNTIL, FOR TO/DOWN TO-DO, WHILE-DO, CASE-OF-ELSE, FUNC and PROC. Now you can learn the language that is slated to become the successor to BASIC.

TINY PASCAL is a complete package that allows you to create, compile and execute progams written in the PASCAL language. You can save source and object code on diskette or cassette (PET version only). Comprehensive user's manual included. The manual can be examined for $10 (refundable with software order).

REQUIREMENTS

PET 16K/32K New ROMS cassette	$40
PET 16K/32K New ROMS diskette	$35
Apple II 32K Applesoft ROM w/DOS	$35
Apple II 48K Applesoft RAM w/DOS	$35
TINY PASCAL User's Manual	$10
6502 Interpreter Listing	$20

FREE postage in U.S. and CANADA
All orders prepaid or COD

master charge
VISA

ABACUS SOFTWARE
P. O. Box 7211
Grand Rapids, Michigan 49510

www.commodore.ca

```
523.2          U.S. Constitution
S369

        Smith Ronald Y   Jones James P
        NOW IS THE HOUR
        New York-Jensen-1980

        P. 963 Illus
        U.S. Constitutiion-Bibliography


   1-U.S. History  2-U.S. Constitution 3-George Washington
   I-TITLE II-Jones James P
```

```
523.2          George Washington
S369

        Smith Ronald Y   Jones James P
        NOW IS THE HOUR
        New York-Jensen-1980

        P. 963 Illus
        U.S. Constitutiion-Bibliography


   1-U.S. History  2-U.S. Constitution 3-George Washington
   I-TITLE II-Jones James P
```

```
523.2          Jones James P
S369

        Smith Ronald Y   Jones James P
        NOW IS THE HOUR
        New York-Jensen-1980

        P. 963 Illus
        U.S. Constitutiion-Bibliography


   1-U.S. History  2-U.S. Constitution 3-George Washington
   I-TITLE II-Jones James P
```
©

FREE your keyboard—interact directly with the screen. Why waste time typing? Use a **3-G Light Pen.**

Mail Coupon of Call Today for Immediate Delivery

3-G Company, Inc. Dept.
Rt. 3, Box 28A, Gaston, OR 97119
(503) 662-4492

Remember, 3-G offers a 30-Day Unconditional Money Back GUARANTEE

☐ TRS-80 Economy $19.95 ☐ TRS-80 Professional $34.95 ☐ PET Professional $31.95 ☐ Apple Professional $32.95

Yes, I want to make my computer more versatile. Rush me ________ 3-G Light Pens. (Add $1.50 for mailing and handling—$6.00 foreign.)

Enclosed is: ☐ check or money order ☐ MasterCharge ☐ Visa

Card No. ________ Exp. Date ________

NAME ________

ADDRESS ________

CITY ________ STATE ________ ZIP ________

- In his business, Al Zenker of Zenker Dental labs in Penndel, Pennsylvania uses our pens for **data entry.** Harry Lee of Pittsfield, Massachusetts uses the pen to **select telephone numbers** to be dialed by his computer. Thorwald Esbensen of Micro-Ed, Inc. in Minneapolis, Minnesota writes **education software** for the 3-G Light Pen. Dr. Richard Kerns of East Carolina University incorporates our pen in a demonstration with a voice synthesizer to **teach** his students how to use computers.
- These people have discovered the benefits of using a 3-G Light Pen. Wouldn't a 3-G Light Pen make your system more versatile and more functional? Yes, of course it would!
- **Don't Wait**—order your pen today and receive:
  1) 3-G Light Pen
  2) Demonstration cassettes (with Professional TRS-80, PET and Apple).
  3) Sample program listing
  4) Complete documentation and instructions
  5) Other Light Pen software and games available.

- **NO ASSEMBLY NECESSARY, READY TO PLUG IN AND USE**

Computer House Div.

Programs for Commodore & Apple

"Legal Accounting" $1200.00
"Political Party Mailing List" 150.00

ENGINEERING & MACHINE SHOP

"Machine Part Quoting" $280.00
"Trig & Circle Tangent" 70.00
"Bolt Circle" 25.00
"Spur Gears" 35.00
"Beams; Stress & Deflection" 145.00
"Tank Thickness"
For Filament Winding 85.00

All 6 for only $495.00

"SCRUNCH" — $36.00
For Apple II or Apple II Plus. Compacts Basic Programs up to 20%.

Dealer inquiries invited

COMPUTER HOUSE DIV.
1407 Clinton Road
Jackson, Michigan 49202
Phone: (517) 782-2132

Computer House Div.

Programs for Commodore Computers

1 - F.E.T.-Recover; File Editing Tools, Adds 11 commands to assist disk recovery $65.00
2 - SOF-BKUP; copy disks faster including random files. Displays error messages for bad blocks. $40.00
3 - SUPER-RAM; checks every ram address against every other ram address, 2001 series $20.00
4 - VARI-PRINT; prints listing of all variables with every line number where each occurs $25.00
5 - DOCU-PRINT; prints CRT to printer, use in your own program, in basic $20.00
6 - SCREEN DUMP/REPEAT; similar to DOCU-PRINT except in machine language. Use with anyones program, REPEAT similar to Model 8032 . . . . . $35.00
7 - TRACE-PRINT; prints listing of all line numbers in order of execution $25.00
8 - SCRUNCH-PLUS; packs a copy of program to save up to 25% of memory space. $25.00
9- -SORTER; sorts a one dimensional array alphabetically (in machine language). $35.00

PACKAGE PRICE $170.00 FOR ALL 9 PROGRAMS
SAVE $120.00

Dealer inquiries invited

COMPUTER HOUSE DIV.

www.commodore.ca

Efficiency with Subroutines

Mike Richter

Efficient programming is sometimes necessary and always desirable in a microcomputer program. There are two kinds of efficiency, usually in conflict: economical use of memory, and speed of execution. With careful program design, both objectives may be achieved in the same code.

The specific methods outlined here deal primarily with the use of subroutines; other techniques are available, and will be dealt with elsewhere. It is possible to design the program from the start in accordance with these methods, but it is usually easier to develop the code without worrying about efficiency, then add features later to save time and memory. In that effort, the Programmer's Toolkit is a valuable aid. I have also written a program ("SUPERLIST") which provides useful cross-reference tables. Those tools are for the Commodore PET, the machine on which the rules were developed; they should apply without modification to all Commodore machines and probably to all microcomputers employing Microsoft BASIC (Apple, TRS-80 Level II). Most of them will also work on other BASIC systems, but the user will have to try them to be sure.

When to use a subroutine

A subroutine is a convenient, efficient means of executing a segment of program repeatedly. If the code is called only once, it probably belongs in line - that is, written directly in the place it is used. That saves memory (eliminating both the GOSUB and line number and the RETURN) and speed (the operating system does not have to hunt for the line). When a subroutine is called, the operating system goes to the top of the program and traces through each line until it finds the appropriate line number; that process is slow, and becomes very slow when there are many lines to count. In detail, PET knows where the first instruction is and starts from that absolute address. It checks the line number, and if it is not the one desired adds the instruction length (which follows the line number) to the starting point and repeats the process at the new location. Clearly, the search takes little time per test, but it may have to do many tests to find a number occurring late in the program. Remember, BASIC always starts at the *beginning* of the program.

A similar process occurs with variables. PET files array pointers after variables, begins with the first variable defined and traces to the one you specify. Therefore, the program will run slightly faster if it encounters frequently used variables and arrays before those that occur rarely. In initialization, it is worthwhile to declare a high-use variable before one that appears seldom. (The time saving is not usually substantial, since there are seldom enough variables in a practical program to make the search take very long.)

There are also times when you may want to make code which is not programmable as a subroutine perform as though it were one. A useful instruction pair for the purpose is the ON ... GOTO. A case in point occurs if an error condition may be found in a common subroutine. It is frequently necessary to jump out of that code for error processing. If subroutine is used, its call remains on the stack; repeated use will cause the stack to overflow, and PET will give you an "OUT OF MEMORY ERROR". You look at FRE(0), and have plenty of memory left, but the program won't CONTinue. To avoid the problem, use a variable, say DE for destination. Assign it a value whenever the program is "calling" the "subroutine". Instead of GOSUB, GOTO the subroutine. Then instead of RETURNing, use ON DE GOTO the collection of return points. Now, you may GOTO the error code without cluttering up the stack.

Functions and FOR ... NEXT

When a numeric function is used repeatedly in the code, a function may be defined. For example, I employ a packing function FNP(X) to count from 0 through 200 with single characters that PET can INPUT.

It is defined by:

```
DEF FNP(X) = X + 48 + 7*(X>9).
```

The corresponding unpacking function is:

```
DEF FNU(X) = X - 48 - 7*(X>65).
```

Note that a numeric equivalent of an IF test is used; a function must be a single statement, and cannot include an IF. Invoking a function carries along only a single variable (here, X), but other variables or constants can be invoked by writing them into the definition directly. For example, one may declare a variable (say, I) early by putting it into the parentheses of a DEFinition; when the function is invoked, any variable or number may be substituted by putting it into the parentheses. No other term in the definition allows substitution.

The FOR ... NEXT loop has many properties of a subroutine, but has different rules for efficiency. The normal execution of such a loop (the one taken most often) dictates its speed. Let's look at a simple loop:

```
1000 FOR I = 0 TO 255
1010 IF I <> 19 GOTO 1100
1020 PRINT "HOME";
1040 NEXT I
1100 IF I <> 147 GOTO 1200
1110 PRINT "CLEAR";
1120 GOTO 1040
1200 PRINT CHR$(I);
1210 GOTO 1040
```

www.commodore.ca

INTRODUCING
THE NEW IMPROVED

BUSINESS ENHANCEMENTS COMPUSERVICE BUSINESS SOFTWARE
FOR
COMMODORE
AND
APPLE COMPUTER SYSTEMS

Micro Mini Computer World Inc. is an execlusive distributor for the BUSINESS SOFTWARE developed by Business Enhancements Compuservice of Escondido, California.

If you are selling or using the COMMODORE BUSINESS MACHINES or the APPLE computer systems, then you should provide yourself and your customers with the MOST COST EFFECTIVE and COMPREHENSIVE business software for a business computer system.

CURRENT B.E.C. SOFTWARE

- General Ledger--Master File 1000 Accounts and Journal File 4400 Entries
- Accounts Receivable--Master File 1170 Accounts and Invoice File 1430 Entries
- Accounts Payable--Master File 1170 and Invoice File 1430 Entries
- Payroll--440 Employees
- Job Costing--1100 Items Per Disk
- Inventory--1100 Items Per Disk
- Mail List/Customer Information--1000 Entries Per Disk

Above figures apply to CBM 2001 computer system with 32K CPU and 2040 dual disk. With the new CBM 8050 Megabyte disk the volumes will be increased significantly.

B.E.C. SOFTWARE FEATURES:

- Complete and total documentation
- Step by step walk through on every program operation, with examples
- Each package is MENU driven and uses dynamic load and overlay once the initial menu is loaded.
- Examples are provided for all reports and other printed forms. All forms are available from New England Business Services Inc. (NEBS).
- All input/output operations use random access
- Sorts are machine language sorts
- Programs are interactive with the General Ledger and update the GL automatically.

B.E.C. VALUE ADDED BENEFITS

- Total commitment to the development of excellent business software for the COMMODORE and APPLE computer systems.
- At reasonable rates Micro Mini Computer World Inc. will provide software modifications to meet customer require ments. (Call MMCWI for further information)
- EXTENDED WARRANTY which entitles users to any enhancements to accounting software during the year of coverage. (Cost is $100 per year)

Dealers and Interested Parties may obtain a copy of the B.E.C. software documentation for $25. If after review you are not interested, send the documentation back, in re-saleable condition, for a full refund or apply the $25 toward your first software purchase.

The NEW B.E.C. BUSINESS SOFTWARE requires a special ROM chip for proper operation.

Suggested Retail Prices are:

1. Rom chip $70 (required on any software package)
2. Individual software package $150
3. All seven software packages $995 (save $55)

DEALER INQUIRIES ARE INVITED

micro mini COMPUTER WORLD Inc.

TOWN & COUNTRY SHOPPING CENTER

74 ROBINWOOD AVE. COLUMBUS, OHIO 43213
(614) 235-5813 (614) 235-6058

www.commodore.ca

That code prints the sequence of characters on the screen, handling two problem characters with special logic.

The first improvement eliminates the GOTO for the 254 cases that are normal, and applies it only to the special two.

```
1000 FOR I = 0 TO 255
1010 IF I = 19 GOTO 1100
1020 IF I = 147 GOTO 1200
1030 PRINT CHR$(I);
1040 NEXT I
1100 PRINT "HOME";
1110 GOTO 1040
1200 PRINT "CLEAR";
1210 GOTO 1040
```

Now, the program simply steps along on the normal path without having to hunt for a line number. Only in two cases out of 256 does it have to search.

Another element of efficiency is eliminating unnecessary lines, packing them with colons instead of separate numbers. Each time you do that, you save four bytes of storage and the time needed for PET to step across the line number. Another speed-up and memory saving is accomplished by eliminating spaces; the code is harder to read, but is more efficient. Similarly, we can look for simplification of some statements. Combining those, we may try:

```
1000 FORI = 0TO255:IF(iAND127) = 19GOTO1100
1010 PRINTCHR$(I);
1020 NEXTI
1100 IFI = 19THENPRINT"HOME";:GOTO1020
1110 PRINT"CLEAR";:GOTO1020
```

Next, let's use the conditional more cleverly; it will save memory, but at some cost in time. We may also use a dummy print variable, X$, to compress the code. When those actions are combined, we can eliminate the GOTO altogether.

```
1000 FORI = 0TO255:X$ = CHR$(I):IF(IAND127)
 = 19THENX$ = "CLEAR":
IFI = 19THENX$ = "HOME
1010 PRINTX$;:NEXT
```

The final step was to eliminate the last "I". It is not necessary to name the variable of a simple FOR ... NEXT loop; the operating system will take the one at the top of the stack when it can. Another undocumented feature is that in nested loops, one may write FORI = ... FORJ = ... NEXT J,I. I recommend that you name the variable in the NEXT whenever the loop is complex, but save the storage and time (to look up the variable) in very simple cases - such as our example.

Location

Since the time to locate a line depends on how much code is ahead of it (i.e., has lower line numbers), locating programs is important. Putting a high-rate subroutine that is called from many places at a low line number saves in two ways: speed is enhanced since searching is reduced, and memory is saved since there are fewer characters needed in each call. (Although PET stores BASIC addresses as two packed bytes in the listing, within the program they are kept as ASCII strings in GOTO and GOSUB.) The same rule suggests that such rare operations as initialization be moved late in the program. Therefore, I will often have a strange-seeming beginning to my code:

```
10 GOTO9000:REM M.RICHTER 90064 SEP80
```

The first line goes to the initiation routine and identifies me, my zip code, and the approximate date of the program version. The remaining two-digit line numbers are used for utility subroutines, saving speed and memory.

When you need a very "tight" (high-speed) loop, an incomplete FOR ... NEXT structure is quite useful. Suppose you need to transfer characters between files 2 and 3. You might write

```
1000 GET#2,X$:IFST = 0THENPRINT#3,X$;:
GOTO1000
1010 RETURN
```

That's economical in memory, but sloooow. Instead, try:

```
1000 FORK = 0TO999:GETx$:IFST = 0THENPRINT#3,
X$;:NEXT:GOTO1000
1010 K = 0:RETURN
```

When you have finished developing a program, try the methods outlined here to improve it. In the simple example we would have reduced memory from 132 bytes to 70, and in a large program might have speeded execution by a substantial factor. The three cases listed above have been coded, instrumented and run. Table 1 shows the size of each program in bytes and the time it takes to run essentially as listed. In addition, it reports the effect of having 100 and 500 instructions in the code at lower line numbers. Frequently, I find that even good programs can be shortened in execution time by at least 20%; they can be made to release 10-30% of their code with these and similar techniques. Those savings are frequently the difference between products that sell and those that don't.

Table 1

EFFICIENCY IMPROVEMENT RESULTS

Case No.	Size (bytes)	Execution basic	time + 100	(jiffies) + 500
0	132	470	536	1277
1	130	264	268	269
2	100	258	263	264
3	70	298	299	300

©

www.commodore.ca

Why

PEDISK ?

FLOPPY DISK SYSTEM FOR PET, AIM, SYM, KIM

FAST — loads programs at least 3 times faster than 2040. It is the fastest disk available for PET.

VERSATILE — With its built-in S100 expansion chassis, any peripheral imaginable can be added to your computer.

SOPHISTICATED — The CRS/DOS op system provides advanced file handling, indexed-sequential, but very easy to use.

RELIABLE — single chip disk controller using IBM 3740 recording format gives maximum timing margins — super reliability. Auto error recovery.

SIMPLE — An enhanced BASIC command set makes PEDISK very easy to use. !LOAD, !SAVE, !OPEN, !CLOSE, !RUN, etc. No initialization required.

COMPATIBLE — read diskettes from other systems — 6502PDS, AIM, SYM, and even RADIO SHACK (special software). IBM 3740 soft sector single density.

PEDISK PACKAGE 1 .... $799.95
5" DISK SYSTEM, CASE AND POWER SUPPLY

PEDISK PACKAGE 2 .... $895.00
5" DISK SYSTEM, S100 CARD CAGE, CASE AND POWER SUPPLy

PEDISK PACKAGE 4 .... $1495.00
8" DISK SYSTEM, S100 CARD CAGE, CASE AND POWER-SUPPLY

EXS100 DISK CONTROLLER BOARD $49.95
BARE BOARD

EXS100 DISK CONTROLLER KIT $225.00

AIM, SYM, KIM ADAPTOR KIT $25.00

CRS/PDOS SOFTWARE SYSTEM $75.00
SPECIFY OLD OR NEW ROMS, MEMORY SIZE 8K, 16K 32K

Introducing . . .

6502PDS !

6502 PROFESSIONAL DEVELOPMENT SYSTEM

6502PDS: MINI **$2695.00**

The 6502 PDS is a versatile multi-card microcomputer designed and programmed for professional engineering and program development work, scientific computing, and general processing. This system provides the maximum in capability at the lowest possible cost by utilizing the industries must widely used computer bus - the S100. With a choice of over 500 peripherals including telephone interface, speech synthesizers, vocoders, and even associate memory, the potential end use is unlimited. The 6502 PDS is housed in a sturdy S100 mainfrain containing the 6502 MPU, Multiple I/O Card, RAM, and Disk Controller Board. This leaves room for future expansion. The system can be connected to any RS232 terminal or used with the optional internal Video Board.

6502PDS: 8" DISK **$3495.00**

Standard 6502 PDS except that full size 8" floppy disks are included instead of minifloppy disks. The S100 Disk Controller can handle up to four 8" disk drives. At 250 Kbytes of capacity each, the total system capacity could reach 1 Megabyte.

fullFORTH +

FULL FEATURE "FORTH" FOR 6502 SYSTEMS

INTERPRETER — can be executed directly in an interpretive mode to speed testing and debugging.

CROSS-COMPILER — words can be individually compiled and tested, the entire program can also be cross-compiled for maximum efficiency.

COND. ASSEMBLER — Machine language modules can be intermixed and conditionally assembled to fullFORTH.

STRING HANDLING — variable length constants and variables are allowed. Processes compare, move, concatenate and sub-string words.

FLOATING POINT — process 5 or 9 digit integer and floating point numbers for arithmetic operations.

SCREEN EDITOR — contains a unique full cursor visible screen editor.

SPECIFY PEDISK, 6502PDS, COMMODORE 2040 DISK **$65.00**

The Spacemaker

***SPACEMAKER** **$29.00**
Spacemaker is a utility device for new PETs which allows user selection between ROMs that occupy the same address space. Spacemaker is a verticle 4.25 inch by 2.5 inch board that plugs into a ROM expansion socket inside the PET. The user simply installs a pair of conflicting address ROMs on Spacemaker, plugs it into the appropriate socket, and is up and running. ROM selection is provided by flipping a switch.

USER I/O **$12.95**
This option for Spacemaker allows the user software selection and switching of ROMs. User I/O consists of a user port connector, specially designed Spacemaker jumpers for connection to Spacemaker, and our utility software Spacectl supplied on Commodore or PEDISK diskett.

*NEED MORE ROM ROOM?

ROMDRIVER **$39.00**
A second mode of Spacemaker provides greater flexibility: with ROMDRIVER, you can control ROM switching and selection from software control. PET owners can daisy-chain Spacemakers together, selecting (under software control) one of two, one of four, even one of six ROMs.

ROM I/O **$9.95**
A utility software package for ROMDRIVER owners allowing software switching of ROMS. Available on commodore or PEDISK diskettes.

FOR INFORMATION, SEE YOUR DEALER OR:

cgrs MICROTECH

P.O. BOX 102 • LANGHORNE, PA 19047 • (215) 757-0284

www.commodore.ca

Train Your PET® with MICRO-ED instructional tapes

All programs work with any 8K PET, old or new. PET is the registered trademark for Commodore Business Machines, Santa Clara, CA.

MICRO-ED has educational software for the PET microcomputer. We specialize in programs with these features:

- They have been written by professional educators. Our main author is Thorwald Esbensen, named in 1980 as one of North America's top school executives by Executive Educator magazine.
- Our tapes are independent modules, each one a complete lesson in itself.
- Every instructional lesson ends with a summary of student performance.
- Our programs are attractive and motivating to students.
- We will gladly replace any tape that fails to load or run properly.

Unless otherwise specified, each tape can be purchased for $7.95

MUSIC (these tapes use sound)

- •MU-1 Lines and Spaces of the Treble Clef (elementary)
- •MU-2 Higher, Same, Lower (elementary)
- •MU-3 Matching Rhythms (elementary)

REFERENCE SKILLS

- •RS-1 Dictionary Guide Words (elementary)
- •RS-2 Library Terms (elementary)
- •RS-3 Making an Outline (elementary)
- •RS-4 Putting Fiction Books in Alphabetical Order (elementary)

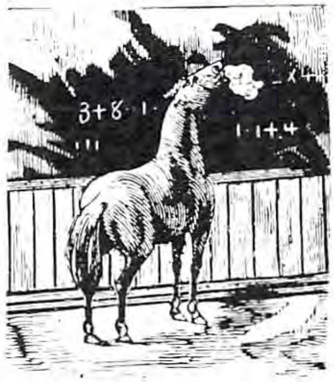
VOCABULARY

- VO-1 Synonym Series (high school and adult) 3 tapes $21.00
- VO-2 Vocabulary Series (upper elementary and high school) 24 tapes $168.00
- VO-3 Antonym Machine (elementary)
- VO-4 Homonym Machine (elementary)

You may wish to order tapes by the MICRODOZEN. Any twelve $7.95 tapes can be purchased for $84.00.

www.commodore.ca

Also from MICRO-ED



MATHEMATICS

- •MA-1 Math Bid (elementary)
- •MA-2 Count 'Em (kindergarten and first grade)
- •MA-3 Story Problems in Addition and Subtraction (elementary)
- •MA-4 What Number is Missing? (kindergarten and first grade)
- •MA-5 Target Math (elementary)
- •MA-6 Adding with Objects (primary grades)
- •MA-7 Subtracting with Objects (primary grades)
- •MA-8 Working with Basic Addition Facts (primary grades)
- •MA-9 Working with Basic Multiplication Facts (elementary)
- •MA-10 Adding or Subtracting Two- and Three-place Numbers in Columns (elementary)
- •MA-11 Math Shootout (elementary)
- •MA-12 Bar Graph (elementary)
- •MA-13 Which Number Comes Next? (uses a 3G Light Pen) (first grade)
- •MA-14 Locomotive (uses a 3G Light Pen) (kindergarten and first grade)
- •MA-15 Math symbols: Greater Than, Less Than (elementary)
- •MA-16 Math symbols: Greater Than, Less Than (uses a 3G Light Pen) (elementary)
- •MA-17 Addition with Carry (by Don Ross) (elementary) $20.00
- •MA-18 123 Digit Multiplication (by Don Ross) (elementary) $20.00
- •MA-19 Long Division (by Don Ross) (elementary) $20.00

SPELLING

- •SP-1 Guess That Word (elementary and up)
- •SP-2 Level A (second grade) 7 tapes $49.95
- •SP-3 Level B (third grade) 7 tapes $49.95
- •SP-4 Level C (fourth grade) 7 tapes $49.95
- •SP-5 Level D (fifth grade) 7 tapes $49.95
- •SP-6 Level E (sixth grade) 7 tapes $49.95
- •SP-7 Compound Words (elementary)
- •SP-8 Hard and Soft C (elementary)
- •SP-9 Hard and Soft G (elementary)
- •SP-10 Dropping the Final E (elementary)

WORD DEMONS

- •WD-1 To/Too/Two (elementary)
- •WD-2 There/Their/They're (elementary)
- •WD-3 Its/It's — Your/You're (elementary)
- •WD-4 Sit/Set (elementary)
- •WD-5 Lay/Lie (elementary)

GRAMMAR

- •GR-1 Agreement of Subject and Verb (elementary and up)
- •GR-2 The Noun (elementary and up)
- •GR-3 The Verb (elementary and up)
- •GR-4 The Adverb (elementary and up)
- •GR-5 The Adjective (elementary and up)

READING

- •RE-1 Tachistoscope (elementary)
- •RE-2 Reading Racer One (elementary)
- •RE-3 Matching Capital Letters (pre-school and kindergarten)
- •RE-4 Matching Capital Letters (uses a 3G Light Pen) (pre-school and kindergarten)
- •RE-5 Which Letter Comes Next? (primary grades)
- •RE-6 Which Letter Comes Next? (uses a 3G Light Pen) (primary grades)
- •RE-7 Matching Words (kindergarten and first grade)
- •RE-8 Matching Words (uses a 3G Light Pen) (kindergarten and first grade)
- •RE-9 Identifying Complete Sentences (elementary)
- •RE-10 Make a Sentence (uses sound) (first grade)

PUNCTUATION

- •PU-1 Run-on Sentences (elementary)
- •PU-2 The Apostrophe (elementary and up)
- •PU-3 End Punctuation (elementary)

OTHER

- •OT-1 Trail West (elementary through adult)
- •OT-2 Direction and Distance (primary grades)
- •OT-3 Haiku (elementary and up)
- •OT-4 Pet Counselor (adult)
- •OT-5 Clock (primary grades)
- •OT-6 U.S. Time Zones (elementary and up)
- •OT-7 States and Capitals (elementary and up)
- •OT-8 Answer Box (teacher in-service)
- •OT-9 Hat in the Ring (A Presidential Election Game) (elementary and up) $9.95

USAGE

- •US-1 Usage Boners (elementary and up) 15 tapes $99.00

Send for free catalogue: **612-926-2292**

MICRO-ED, Inc. • P.O. Box 24156 • Minneapolis, MN 55424

www.commodore.ca

Computing Correlation Coefficients

Brian J. Flynn

"The difference between socialism and capitalism is that in capitalism man exploits man, but in socialism it's the other way around" Unknown

Almost everyone sometimes has the urge to do it. You see a plot of points between two variables, such as gold and silver prices, or wheat harvest and rainfall, or inflation and interest rates, or unemployment rates in Canada and the United States. And you want to draw a line amongst the points which best reflects the apparent trend. Simple and partial correlation coefficients, which are distinguished later, tell us how well a straight line fits the data, or how close the data is to the line, depending on how we want to look at it. They never, however, tell us that one variable is the cause of another. To try to learn this, an amalgam of logic, knowledge, and insight is sometimes our only hope, and is often not enough. For as Tolstoy tells us in **War and Peace,** "However accessible may be the chain of causation of any action, we shall never know the whole chain, since it is endless... ." Nevertheless, a measure of the degree of association between observations on two variables is often handy. And further, if we feel plucky and decide to generalize about the population versus sample relationship between the variables, then the simple or partial correlations are needed for this also. A level II basic program for calculating correlation coefficients is presented here.

A mercantilist, speculator, inflation-hedger, or stock market dabbler might wax wide-eyed at the gold and silver prices shown in Table 1. A quick glance at the data suggests that the two sets of prices are strongly associated. Both, for instance, have risen sharply since about 1973 or 1974. Further, gold achieves its highest price at the same time that silver does, for the data displayed. And the time periods for lowest price also coincide. But if the relationship between the two variables is not mild, then exactly how robust is it? A numerical answer to this type of question, according to James R. Newman's **The World of Mathematics,** was first proposed by the mentally spray 19th century Englishman, Sir Francis Galton, cousin of Charles Darwin. Almost obsessed with measuring, Galton's creed was "Whenever you can, count." This passion to tally, Newman tells us, led Sir Francis to conduct several intriguing social experiments. In one of these, Galton counts the number of fidgets per minute among people attending lectures, and presumably from this calculates a coefficient of boredom. And in a more elevated exercise, Sir Francis creates a "Beauty Map" of the British Isles by classifying girls that he passes on the streets as "attractive, indifferent or repellent." He records observations by pricking a hole in a piece of paper "torn rudely into a cross with a long leg," which he conceals in his pocket. Since beauty and beastliness are recognized only by comparision to the common, their relatively few numbers were probably recorded on the arms of the cross. And the leg probably held the tally for the category with largest membership, "indifferent." For those of you Britain bound, London ranked highest, and Aberdeen lowest.

Historian Newman tells us that the idea of an "Index of Correlation" for measuring the degree of association between two variables came to Galton one morning while he waited at a train station, "pouring over a small diagram in (his) book." Sir Francis' study of heredity, and more specifically his quest for learning the degree to which traits are passed from father to son, probably inspired discovery. At any rate, Galton's idea, refined by Karl Pearson and others, is today called a coefficient of correlation, or Pearson product moment correlation in august circles, and now takes this form:

$$r = \frac{\sum_{i=1}^{N}(X_i-\overline{X})(Y_i-\overline{Y})}{\sqrt{\sum_{i=1}^{N}(X_i-\overline{X})^2}\sqrt{\sum_{i=1}^{N}(Y_i-\overline{Y})^2}}$$

Σ is the greek symbol for "sum of." X and Y are variables, such as gold and silver prices, with X_i and Y_i representing the "ith" observation on each. N is the number of observations. And the bars over the X and Y signify mean or arithmetic average.

The mystery of the formula is revealed by dividing the numerator and each term within square roots by N-1, with the value of r unchanged. The numerator becomes the covariance, or co-variation, of X and Y. This measures the degree to which X and Y are associated in linear or straight line fashion. If the two variables usually rise and fall together, the covariance is positive. But if one usually rises when the other falls, it is negative. And if they move without any linear relation at all, it is zero.

One problem with using the covariance as an index of correlation is that its value is influenced by the size or scale of the observations used to compute it. For example, if the pairs of observations on X

www.commodore.ca

PET ATARI APPLE SYM

INTRODUCING ROM PET RABBIT OR CASSETTE

The PET RABBIT contains high-speed cassette routines, auto-repeat key feature, memory test, decimal to hex, hex to decimal, and other features. Coexists with the BASIC PROGRAMMERS TOOLKIT. Works with 2.0 ROMS (New) and new style cassette deck.

Cassette versions configured for $1800, $3000, $3800, $7000, and $7800.
Cassette and manual — $29.95

ROM version configured to plug into P.C. board at $A000.
ROM and manual — $49.95

FREE ROM RABBIT with purchase of 8K PET and tape deck.
SPECIAL — ROM RABBIT and cassette deck — only $134.95

MACRO ASSEMBLER AND TEXT EDITOR

Macro and conditional assembly, string search and replace, 10 char./label, AUTO line numbering, MOVE, COPY, DELETE, NUMBER, and much more. 20+ commands, and 20+ pseudo ops.

PET cassette version (ASSM/TED) — $49.95
PET disk version (MAE) — $169.95
ATARI cassette version with machine language monitor — $53.95

FREE ASSM/TED and ROM RABBIT with purchase of 32K PET and cassette deck.
FREE MAE with purchase of 32K PET and disk drive.

TINY-C FOR PET

An adaptation of the TINY-C interpreter sold by Tiny-C Assoc. Useful for learning a modern structured programming language. Diskette — $45.00, Owners manual — $40.00

FREE MAE and TINY-C with purchase of 32K PET, disk drive, and printer.

COMPILERS

Graphics Drawing Compiler for PET and SYM. Works with Macro ASSM/TED. The GDC is composed of a number of macros which emulate a high-level graphics drawing language. In addition to the macros, GDC provides some very useful enhancements to the ASSM/TED. Manual and Cassette — $29.95

Music and Sound Composer for PET. Works with Macro ASSM/TED. The MSC is composed of a number of macros which emulate a high-level computer music language. In addition to the macros, MSC provides some very useful enhancements to the ASSM/TED. Manual and Cassette — $29.95

I/O KIT

PET I/O Experimenters Kit. Allows easy access to IEEE or user port for the construction of external circuits. Kit — $39.95

ORDERING TERMS

Send check or money order in U.S. dollars. Add 2% for postage for CBM orders. Overseas software orders add $5.00. All software mailed free in USA and Canada. Purchase orders acceptable.

EHS IS NOW A COMMODORE DEALER

EHS offers a number of software products for PET, ATARI, APPLE, and other 6502 computers. Now we sell CBM hardware. If you're in the market for PET products, be sure to look for our FREE software offers.

CBM	PRODUCT DESCRIPTION	PRICE
2001-8KN	8K RAM-Graphics Keyboard	$ 795.00
2001-32KN	32K RAM-Graphics Keyboard	$1295.00
2001-32KB	32K RAM-Business Keyboard	$1295.00
8032	32K RAM-30 Col.-4.0 O/S	$1795.00
2023	Friction Feed Printer	$ 695.00
2022	Tractor Feed Printer	$ 795.00
2040	Dual Floppy-343K-DOS 1.0	$1295.00
2050	Dual Floppy-343K-DOS 2.0	$1295.00
8050	Dual Floppy-974K-DOS 2.0	$1695.00
C2N Cassette	External Cassette Drive	$ 95.00
CBM to IEEE	CBM to 1st IEEE Peripheral	$ 39.95
IEEE to IEEE	CBM to 2nd IEEE Peripheral	$ 49.95
8010	IEEE 300 Baud Modem	$ 395.00
2.0 DOS	DOS Upgrade for 2040	$ 50.00
4.0 O/S	O/S Upgrade for 40 Column	$ 100.00

EDUCATIONAL DISCOUNTS
BUY 2 — GET 1 FREE

TRAP 65

TRAP 65 prevents the 6502 from executing unimplemented instructions. Have you ever had your system to crash on a bad upcode? This is a real machine language debugging tool and time saver. Also useful for teaching trap vectoring and extension of instruction set in schools. 3½ x 4¾ printed circuit board which plugs into 6502 socket of any PET, APPLE, SYM. Only $149.95

ATARI M.L. MONITOR

Load and save binary data on cassette. Display and change 6502 registers.
Monitor uses the screen editing capabilities of the ATARI to allow easy use. Cassette and manual — $9.95 (specify memory size).

ATARI MEMORY TEST

When you purchase a new ATARI or add on new RAM modules, you need to be sure that the memory is working properly. (Remember, you only have a short guarantee on your memory!) Cassette and manual — $4.95

APPLE PRODUCTS

Macro ASSM/TED — includes manual, on cassette — $49.95, on disk — $55.95
Apple MAE — similar to PET MAE. A powerful assembly development system on diskette. (Requires license agreement) — $169.95
PIG PEN — 100% M.L. word processor for use with Apple ASSM/TED. Fast text formatting, vertical and horizontal margins, right and left justification, centering, titles, foots, shapes, etc. Manual and source included, on cassette — $40.00, on diskette — $45.00

Apple Mail List System. Provides sorting on zip code or last name. Approximately 1000 names/diskette. Manual and Diskette — $34.95

EASTERN HOUSE SOFTWARE

3239 Linda Drive, Winston-Salem, N.C. 27106
Ph. Orders — 9-4 EST (919) 924-2889 or 748-8446
Send SASE for free catalog

www.commodore.ca

and Y are (1,2), (2,3), and (3,4) versus (10000, 10002), (10002, 10004), and (10004, 10006), then the covariance is 1 in the first case and 4 in the second. Most people, however, will disagree that the X's and Y's in the second instance are any less associated with each other than in the first case. A solution to this dilemma is to divide the covariance of X and Y by the product of the square roots of their variances. When this is done, r emerges.

Simple correlation coefficients are always between -1 and 1, inclusive. If the correlation is close to either extreme, the linear relationship between the two variables is strong. If the correlation is close to zero, however, then the relationship is weak. Figure 1 illustrates this. A word of warning: only *linear* association is measured by the correlation coefficient. In a circle, for example, the linear relationship between the X's and Y's is zero, while the circular association is perfect.

Returning to our example, the simple linear correlation between gold and silver prices is a herculian 0.98. And herein lies a pitfall.

> *"The cause of lightning,"* Alice said very decidely, for she felt quite sure about this, *"is the thunder - no, no!"* she hastily corrected herself, *"I meant it the other way."*
>
> *"It's too late to correct it,"* said the Red Queen, *"When you've once said a thing, that fixes it, and you must take the consequences."*
>
> Lewis Carrol

A strong relationship between two variables, either direct or inverse, often tempts us to call one the cause and the other the effect. At times this is reasonable. But at other times it is ridiculous. For example, there may be a strong, positive correlation between consumption of scotch whiskey in the US and the number of catholic priests. To presume that the later influences the former is folly. If the truth be told, an increase in a third variable, such as the nation's population, may largely explain increases in the other two. Hence, we should always rely upon common sense or well established theory in determining which correlations are reasonable and which are coincidence. A third variable which may influence both gold and silver prices is speculative fever, fed by a fear of inflation. If the general level of prices in the economy increases, some investors may try to keep the purchasing power of their assets intact by, among other things, buying gold and silver. Prices of the metals are likely then to rise since supplies are largely fixed. But rising prices may attract the attention of avaricious but heretofore dubious investors, and induce them to plunge pocket-first into the market. Hence, gold and silver prices may end up rising more sharply than the general price level.

If the temperature of speculative fever could be taken, and its influence on both gold and silver values isolated, then what we think is the true association between prices of the two metals could be measured. A partial correlation coefficient is used in doing this. It measures the linear association between one variable and another, with all other specified variables held constant. Partial correlations are calculated in the computer program by inverting the matrix of simple correlations, and by then performing a couple of simple steps in arithmetic.

Finally, although speculative fever is not measured here, even if it were our results would have to be regarded with healthy suspicion since: "In statistics things are seldom as they seem, skim milk masquerades as cream."

TABLE 1

PRICES PER TROY OUNCE OF GOLD AND SILVER IN THE UNITED STATES

YEAR	PRICE OF GOLD	PRICE OF SILVER
1967	$ 35.0	$ 1.5
68	39.0	2.1
69	41.5	1.8
1970	36.2	1.8
71	41.0	1.6
72	58.1	1.7
73	96.5	2.6
74	158.1	4.8
1975	161.7	4.4
76	125.9	4.4
77	147.5	4.6
78	192.9	5.4
79	303.7	10.7
Jan 1980	737.8	43.8
Feb. 80	690.2	38.5

Source: from prices and indices compiled by the Bureau of Labor Statistics

NOTES ON THE COMPUTER PROGRAM

1. The simple correlation coefficient is calculated in the computer program thusly:

$$r = \frac{N \sum_{i=1}^{N} X_i Y_i - \left(\sum_{i=1}^{N} X_i\right)\left(\sum_{i=1}^{N} Y_i\right)}{\left[\left\{N \cdot \sum_{i=1}^{N} X_i^2 - \left(\sum_{i=1}^{N} X_i\right)^2\right\}\left\{N \sum_{i=1}^{N} Y_i^2 - \left(\sum_{i=1}^{N} Y_i\right)^2\right\}\right]^{\frac{1}{2}}}$$

This formula produces a more accurate answer than the one in the article. It also requires just a little bit more memory, however.

2. If the observations on a variable are all the same, then any simple correlation coefficient involving that variable cannot be computed since division by zero would be required. The computer program handles this thusly:

DATA			SIMPLE CORRELATION MATRIX	
X1	X2		X1	X2
3	7	X1	Undef'd	Undef'd
3	8	X2	Undef'd	1
3	20			

www.commodore.ca

Hard Working Software for PET/CBM™ Micros

MATRIC™

MATRIC expands Commodore BASIC with fourteen new commands for handling arrays.

Display a matrix on the screen and change its values. Transfer data between matrices or fill a matrix with a constant. Transpose. Transfer diagonals between matrices, or from a matrix to a vector, or from a vector to a matrix, or fill a diagonal with a constant. Do vector or matrix addition, subtraction, multiplication; elementwise multiplication, division, squares, and square roots. Inversion. Determinant. Eigenvalues and eigenvectors of a square, symmetric matrix.

Algebraic style syntax, checks for conformability, extended error messages.

The 5K machine language program comes on tape or disk with a 32-page manual. Specify size and ROM set of your machine. Price: $125.

PRO-GRESS™

This multiple regression program reads data from tapes or CBM disk. Concatenates files. Selects records. Deletes cases with missing data. Transforms variables. Generates new variables. Allows keyboard input of means, standard deviations, correlations. Names variables.

Provides means, standard deviations, correlations; R, R-square, F, degrees of freedom; constant and coefficients, betas, Student's t's. Output to screen, or to ASCII or CBM printer.

Analyzes 10 predictors in 8K; 25 in 16K; 45 in 32K. No limit on number of records. Approximate timing: ¼ minute per 100 records with one predictor; 30 minutes per 100 records with 45 predictors.

You get two programs and a manual. Program I has elaborated instructions, requires 16K or 32K. Program II has abbreviated instructions, runs in 8K or more. Cassette tape: $45. Disk: $50.

TEXTCAST II™

A word processor for 8K and larger machines, old or new ROMs. All in machine language.

Edit with tapes or disks in any combination. Produces ASCII files that can be used by other programs.

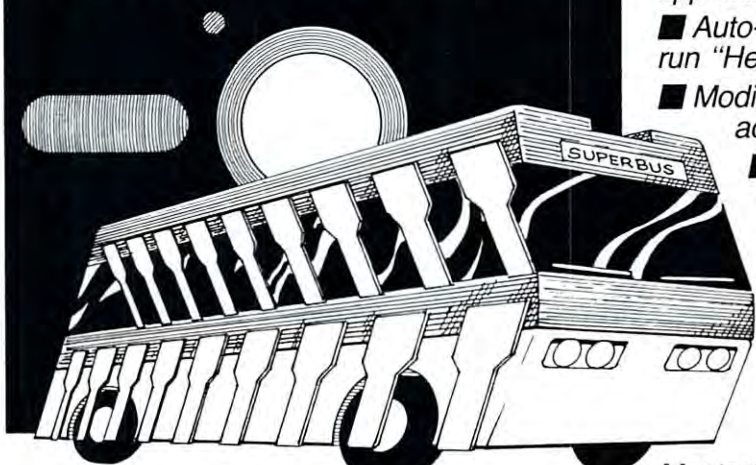
Unfinished words at the end of a line leap to the next line while you type. Powerful screen editor with full control over visible cursor. Repeat action on all keys and commands.

Use printer in typewriter mode. Prints files with centering (enhance on PET printer), underlining (reverse on PET printer), right justification, page numbers at top or bottom. Set left margin and line length. Transmits all NEC Spinwriter® characters and commands.

You get old and new ROM versions of the program, and revised manual. On tape: $75; on diskette, $80.

Order your WorkerWare from:
COGNITIVE PRODUCTS,® P.O. Box 2592, Chapel Hill, NC 27514

"SuperBus" will haul it all!



- 18 Computers can share disk drives, printers or other devices on the IEEE bus.
- Includes RS-232 and terminal programs. (Hardware will soon be available for approximately $50.)
- Auto-boot. You can even initialize disks and run "Hello" program.
- Modify your basic commands for restricted access (better security).
- RUN/STOP key. Disabled, or modified to "Return to menu."
- Chain disk programs automatically.
- Auto clear.
- All keys repeat (adjustable).
- Low cost. 2 ROM and 1 disk set (one set required for each computer). **$195⁰⁰**

For PET/CBM users. Schools, banks, laboratories and businesses can greatly multiply their capabilities at an unbelievably low cost.

MasterCard, VISA and C.O.D. orders accepted. 15-day return privileges. Specify N or B keyboard. Dealer inquiries invited.

CYBERIA INC. (515) 292-7634
2330 Lincoln Way, Ames, Iowa 50010

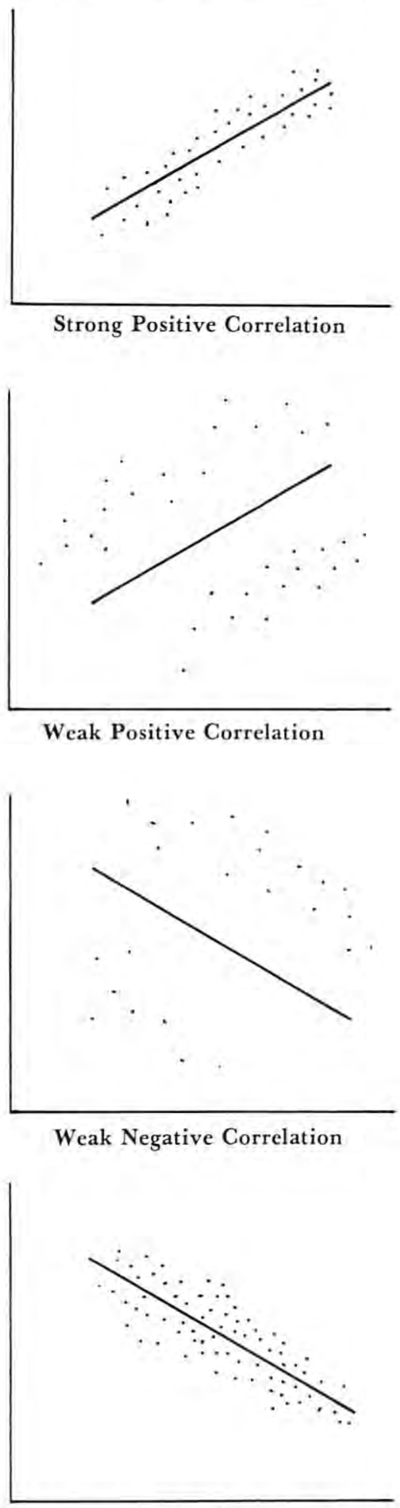
LIMITATIONS WHILE ON THE SUPERBUS: 1) THE CASSETTES CANNOT BE WRITTEN TO. 2) ONLY THE SECOND CASSETTE CAN BE READ FROM. 3) ONLY ONE USER CAN BE USING THE SYSTEM RESOURCES AT ONE TIME, BUT ANY NUMBER CAN BE WAITING.

www.commodore.ca

Further, if any simple correlations are undefined, then the simple correlation matrix cannot be inverted, and hence, the partial correlation coefficients are not computed.

3. Line 1120 uses Disk Basic syntax. For systems without this, the following substitute will work just as effectively:
FOR I = 1 TO K: A$(I) = "X" + RIGHT$(STR$(I), LEN(STR$(I)) - 1): NEXT

VARIETIES OF SIMPLE LINEAR CORRELATIONS



Strong Positive Correlation

Weak Positive Correlation

Weak Negative Correlation

Strong Negative Correlation

Figure 1

```
10 CLEAR 1000
20 REM COMPUTING SIMPLE & PARTIAL ¬
     ¬CORRELATION COEFFICIENTS; ¬
     ¬B.FLYNN; SPRING 80
30 REM MODULE 1:ENTER DATA
40 GOSUB 1000
50 REM MODULE 2: COMPUTE SIMPLE ¬
     ¬CORRELATION COEFFICIENTS
60 GOSUB 2000
70 REM MODULE 3: PRINT COEFFICIENTS
80 GOSUB 3000
90 REM MODULE 4: COMPUTE PARTIAL ¬
     ¬CORRELATION COEFFICIENTS
100 GOSUB 4000
110 REM MODULE 5: PRINT COEFFICIENTS
120 GOSUB 5000
130 END
1000 REM MODULE 1
1010 REM SET CEILING ON # OF OBSERVATION
     ¬S (OB) & VARIABLES (VA)
1020 OB = 50:VA = 20
1030 DEFDBL B,D,H,R,S,X
1040 DIM A$(VA),R(VA,2*VA),S(VA),SS(VA),
     ¬X(OB,VA)
1050 CLS: PRINT"THIS PROGRAM COMPUTES ¬
     ¬SIMPLE & PARTIAL CORRELATION
1060 PRINT"COEFFICIENTS FOR ALL ¬
     ¬COMBINATIONS OF UP TO ";VA;"VARIAB
     ¬LES.
1070 PRINT"UP TO ";OB;"OBSERVATIONS MAY ¬
     ¬BE ENTERED FOR EACH TERM.
1080 PRINT:INPUT"HOW MANY VARIABLES DO ¬
     ¬YOU HAVE";K
1090 IF K>VA CLS:PRINT"SORRY, ONLY";VA;"
     ¬VARIBLES ARE PRESENTLY ALLOWED."
1095 :PRINT"CHANGE LINE 1020 TO REMEDY ¬
     ¬THIS.":END
1100 IF K<2 PRINT "SORRY, IT TAKES AT ¬
     ¬LEAST 2 VARIABLES TO TANGO.":END
1110 REM NAME VARIABLES
1120 FOR I=1 TO K: A$(I)=STR$(I):
     ¬MID$(A$(I),1,1)="X":NEXT
1130 REM ENTER DATA FOR 1ST VARIABLE ¬
     ¬ONLY TO DETERMINE # OF OBSERVATION
     ¬S
1140 CLS:PRINT"PLEASE INPUT DATA. HIT ¬
     ¬'ENTER' WHEN THROUGH.":PRINT
1150 N=OB
1160 FOR J=1 TO OB:X(J,1)=-9999.9
1170 PRINT A$(1);"(";J;")=";:INPUT X(J,
     ¬1)
1180 IF X(J,1)=-9999.9 THEN N=J-1:J=OB
1190 NEXT
1200 IF N<2 PRINT"SORRY, IT TAKES AT ¬
     ¬LEAST 2 OBSERVATIONS TO TANGO.":
     ¬END
1210 REM ENTER DATA FOR OTHER VARIABLES
1220 FOR I=2 TO K
1230 CLS:PRINT"ENTER DATA:":PRINT
1240 FOR J=1 TO N
1250 PRINT A$(I);"(";J;")=";:INPUT X(J,
     ¬I)
1260 NEXT J,I
1270 RETURN
2000 REM MODULE 2
2010 CLS:PRINT"COMPUTING SIMPLE ¬
     ¬CORRELATION COFFICIENTS...
```

www.commodore.ca

```
2020 REM COMPUTE SUM OF OBSERVATIONS ¬
     ¬(S) AND
2025 REM SUM OF SQUARED OBSERVATIONS ¬
     ¬(SS) FOR EACH VARIABLE
2030 FOR I=1 TO K
2040 S(I)=0 : SS(I)=0
2050 FOR J=1 TO N
2060 S(I)= S(I)+X(J,I): SS(I)=SS(I)+X(J,
     ¬I)*X(J,I)
2070 NEXT J,I
2080 REM COMPUTE SIMPLE CORRELATION ¬
     ¬MATRIX
2090 FOR I=1 TO K
2100 FOR J=I TO K
2110 REM SUM OF CROSS PRODUCTS
2120 SC = 0
2130 FOR L=1 TO N: SC=SC+X(L,I)*X(L,J):
     ¬ NEXT L
2140 REM DBL PRECISION SQR ROOT
2150 SQ=(N*SS(I)-S(I)*S(I))*(N*SS(J)-S(J
     ¬)*S(J)):GOSUB 8000
2160 IF SQ<>0 THEN R(I,J)=(N*SC-S(I)*S(J
     ¬))/SQ ELSE R(I,J)=-9999.9
2170 R(J,I)=R(I,J)
2180 IF SQ<>0 THEN R(I,I)=1
2190 NEXT J,I
2200 RETURN
3000 REM MODULE 3
3010 T$="SIMPLE":GOSUB 9000
3020 RETURN
4000 REM MODULE 4
4010 CLS:PRINT"COMPUTING PARTIAL ¬
     ¬CORRELATION COEFFICIENTS...
4020 IF SFLAG$="UNDEFINED" PRINT"PARTIAL
     ¬ CORRELATIONS CAN'T BE COMPUTED ¬
     ¬SINCE":
4025 :PRINT"SOME OF THE SIMPLE CORRELATI
     ¬ONS ARE UNDEFINED."
4030 REM INVERT R
4040 FOR I=1 TO K
4050 FOR J=I TO K
4060 R(I,K+I)=0: R(J,I+K)=0
4070 NEXT J
4080 R(I,K+I)=1: M(I)=I
4090 NEXT I
4100 FOR Q=1 TO K
4110 IF Q=K THEN 4230
4120 REM SEARCH WOULD-BE KEY ELEMENTS ¬
     ¬FOR HIGHEST ABSOLUTE VALUE
4130 HE=ABS(R(Q,Q)): HROW=0: HCOLUMN=0
4140 FOR I=1 TO K-Q
4150 DUMMY=ABS(R(Q+I,Q)): IF DUMMY>HR ¬
     ¬THEN HR=DUMMY: R=Q+I
4160 DUMMY=ABS(R(Q,Q+I)): IF DUMMY>HC ¬
     ¬THEN HC=DUMMY: C=Q+I
4170 NEXT I
4180 IF HE>=HR AND HE>=HC THEN 4230
4190 REM SWITCH ROWS, IF APPROPRIATE
4200 IF HR>=HC: FOR J=1 TO 2*K: ¬
     ¬HOLD=R(R,J):R(R,J)=R(Q,J):
     ¬R(Q,J)=HOLD:NEXT J
4210 REM SWITCH COLUMNS, IF APPROPRIATE
4220 IF HR<HC: FOR J=1 TO K:HOLD=R(J,C):
     ¬R(J,C)=R(J,Q):R(J,Q)=HOLD:NEXT J:
     ¬M(Q)=C
4230 REM ADJUST KEY ROW
4240 B=R(Q,Q)
4250 IF B=0 PRINT"SINGULAR MATRIX": END
4260 FOR J=Q TO 2*K
4270 R(Q,J)=R(Q,J)/B
4280 NEXT J
4290 ADJUST REMAINING ROWS
4300 FOR L=1 TO K
4310 IF L=K AND K=Q THEN 4370
4320 IF L=Q THEN L=L+1
4330 D=R(L,Q)
4340 FOR J=Q TO 2*K
4350 R(L,J)=R(L,J)-D*R(Q,J)
4360 NEXT J,L,Q
4370 REM SWITCH ROWS IN NEXT-TO-FINAL-FO
     ¬RM MATRIX, IF APPROPRIATE
4380 FOR Q=1 TO K-1
4390 IF M(Q)<>Q:FOR J=1TOK:HOLD=R(Q,
     ¬K+J):R(Q,K+J)=R(M(Q),K+J):
     ¬R(M(Q),K+J)=HOLD
4395 :NEXT J
4400 NEXT Q
4410 REM CALCULATE PARTIAL CORRELATIONS
4420 FOR I=1 TO K
4430 FOR J=I TO K
4440 SQ=R(I,K+I)*R(J,K+J):GOSUB 8000
4450 IF SQ<>0 THEN R(I,J)=-R(I,K+J)/SQ ¬
     ¬ELSE R(I,J)=0
4460 R(J,I)=R(I,J)
4470 IF SQ<>0 THEN R(I,I)=1
4480 NEXT J,I
4490 RETURN
5000 REM MODULE 5
5010 T$="PARTIAL": GOSUB 9000
5020 RETURN
8000 REM SUBROUTINE: DOUBLE PRECISION ¬
     ¬SQUARE ROOT
8010 IF SQ=0 THEN 8050
8020 X1=SQR(SQ)
8030 X2=(SQ/X1-X1)/2
8040 IF X1<>X1+X2 THEN X1=X1+X2:
     ¬ GOTO 8030 ELSE SQ=X1
8050 RETURN
9000 REM SUBROUTINE: PRINT CORRELATIONS
9010 REM PRINT IN 10 BY 5 BLOCKS
9020 FOR I=1 TO K STEP 10
9030 FOR J=I TO K STEP 5
9040 CLS:PRINT TAB(10)T$;"CORRELATION ¬
     ¬COEFFICIENTS:":PRINT:C=8
9050 FOR L=J TO J+4
9060 IF L<=K PRINT TAB(C);A$(L);:C=C+10
9070 NEXT L: PRINT
9080 FOR L=I TO I+9
9090 IF L<=K PRINT A$(L); TAB(3);
9100 FOR M=J TO J+4
9110 IF T$="SIMPLE": IF L>=K AND M<=K:
9115 :IF R(L,M)<>-9999 THEN PRINT ¬
     ¬USING"####.#####";R(L,M)ELSE ¬
     ¬PRINT"  UNDEF'D";
9120 IF T$="PARTIAL":IF L<=K AND M<=K:
     ¬ PRINT USING"####.#####"; R(L,M);
9130 NEXT M: PRINT:NEXT L
9140 INPUT"READY";Z
9150 NEXT J,I
9160 RETURN
READY.
```

©

www.commodore.ca

THE apple® GAZETTE

Al Baker's Programming Hints: Apple

Different computers confront the software designer with different problems. The most difficult task in the design process is making efficient use of a computer's assets while avoiding its liabilities. Often, a good design will convert a potential liability into an asset. Far more often, poor design will accentuate a liability. In this case, the software user is left with the results and must live with the problem or go elsewhere.

Five potential design problems with the Apple II immediately come to mind. These are:

No lower case.

Only 40 characters per line on the screen.

No mixed graphics and text.

Missing up-arrow and down-arrow keys.

No Joysticks.

If you own another computer, don't feel superior. I'm sure you have your own list of "Why did they do it that way?"

Look at the list. If you've bought much Apple software, you can probably think of products which successfully bypass, or even capitalize, on these "liabilities". The exception is the lack of Joysticks.

Why Joysticks?

Some software designs demand the use of joysticks. A joystick gives the user instantaneous control over direction of motion. Move the joystick left and the object on the screen instantly starts moving left. Let go of the joystick and it stops. Push the joystick to the right and the object immediately begins moving right.

A joystick has a natural center. Motion away from center is obvious, easy to program, and natural for the user. This control doesn't come naturally to a paddle. The paddle has no natural center. Everything is relative. One approach often used is to divide the paddle's turning radius into thirds: left, center, and right. This normally doesn't work because the user gets lost in the action of the game and loses track of where the center region is.

Most software designers give up and convert the values returned by the paddle into absolute positions on the screen. This is the most unsatisfactory solution of all. The user has lost instantaneous control of screen motion and is often left with a frustrating playing experience. If he has played the same game elsewhere, he must learn a new set of reflexes -- or give up.

The best example of this is the game SPACE INVADERS. The official MIDWAY arcade and Atari home versions of this game use joysticks, or the equivalent, to move the gun. The player has instantaneous control over the gun motion and can spend his psychic energies trying to shoot while avoiding enemy fire. The Apple versions I have played do not provide this level of user control. The programmers chose to use paddle values as absolute gun locations and the user is forced to deal with a gun that seems to have a life of its own. It is always moving in an attempt to reach the screen location that matches the paddle.

Paddle as Joystick

In the listing is an Applesoft example of a joystick simulation routine using the Apple II paddle. Type in the program and run it. You will see a snake made of X's running down the center of the screen. To control the snake, grab the paddle and yank it to the left -- either a lot or a little, and then yank it to the right. The snake started moving left and then stopped. Now yank the paddle right and then left. The snake started moving right and then stopped. instantaneous control that feels right!

The joystick simulator had several major design constraints. First, the center of the paddle must be umimportant. Second, how much the user turns the paddle has to be ignored, as long as it is enough to register. It has to be OK for the paddle to jitter with no effect. Third, the speed the user turns the paddle has to be ignored, as long as it is fast enough to register. If the snake is moving left and the player turns the joystick right slowly for most of its radius, the snake should stop, but it shouldn't stop and then move right. To do this, the user must turn the paddle to stop the snake and then turn the paddle again to start the snake moving right -- yank right, yank right.

Substituting this routine in SPACE INVADERS would recreate the proper "feel" of joystick control that is now missing.

The Program

Lines 1000 to 1080 form the body of the joystick simulator with JY as the value of the joystick. JY can

www.commodore.ca

have three values: -1 if the joystick is pushed left, 0 if it is centered, and +1 if it is pushed right. To properly simulate the joystick, the routine needs two facts: whether or not the paddle has been moving and which direction it is moving now. If the paddle was previously moving, then the user is still in the middle of yanking the paddle and the routine must ignore his input. If the paddle was still, then the routine should change the value of JY based on the current paddle motion.

Two variables are used to compute the joystick's current motion. PA is the value of the paddle now and PM is the value the paddle had last time. PA is obtained in line 1000. In line 1070, PM is assigned the value of PA prior to the RETURN.

PM is also one of the two variables used to compute the paddle's previous motion. PG is the other and always contains the value of PA from two times ago. Line 1060 sets up PG from the value of PM.

Let's follow the routine. Line 1000 obtains the current paddle value and line 1010 determines if the paddle was moving last time. This is determined by looking at the difference between PM and PG. If it was moving, the routine ignores the paddle but updates historic data beginning at line 1060.

```
10  REM        THE PADDLE AS JOYSTICK
20  REM
30  REM
40 PM =  PDL (0)
50 H = 20
60  REM
70  REM          DEMO LOOP
80  REM
90  GOSUB 1000
100  PRINT  TAB( H)"X"
110 H = H + JY
120  IF H > 39 THEN H = 39
130  IF H < 1 THEN H = 1
140  GOTO 90
910  REM
920  REM
930  REM          JOYSTICK SIMULATION
940  REM
950  REM  JY=JOYSTICK READING
960  REM  PA=CURRENT PADDLE READING
970  REM  PM=MOTHER PADDLE READING
980  REM  PG=GRANDMOTHER PADDLE READING
990  REM
1000 PA =  PDL (0)
1010  IF  ABS (PM - PG) > 20 THEN 1060
1020  IF  ABS (PA - PM) < 20 THEN 1060
1030  IF PA < PM THEN JY = JY - 1
1040  IF PA > PM THEN JY = JY + 1
1050 JY =  SGN (JY)
1060 PG = PM
1070 PM = PA
1080  RETURN
```

Line 1020 ignores the paddle if it is only jittering or is being moved very slowly. If the paddle is moving quickly enough, lines 1030 and 1040 update the joystick value by subtracting 1 if it is turning counterclockwise and adding 1 if it is turning clockwise. Finally, line 1050 keeps JY within the range -2 to +1 and lines 1060 and 1070 update historic paddle data.

The program between lines 10 and 140 tests the joystick simulator with the moving snake discussed earlier. Lines 40 and 50 set up the historic joystick data and positions the head of the snake. After calling the joystick routine and moving the snake forward in lines 90 and 100, the position of the head is changed by the position of the joystick in line 110. Lines 120 and 130 keep the snake on the screen.

Conclusion

We've explored the Apple paddle and looked at various ways to use it. If you have other ways of using the paddles or improvements on what I have discussed, please send them to me. I will give full credit for anything I use. Also, I am interested in any ideas you have or specific problems you would like explained.

Al Baker Programming Director The Image Producers, Inc.
615 Academy Dr. Northbrook, IL 60062 ©

NEW FOR APPLE 2 & APPLE 2 Plus
ASTEROID
A REAL TIME HI-RESOLUTION GRAPHICS GAME WITH SUPER SOUND EFFECTS. THREE SKILL LEVELS - NOVICE, AVERAGE ("Arcade" difficulty") and EXPERT WHERE ASTEROIDS ARE ACTUALLY ATTRACTED TO YOUR SHIP!
ASTEROID
Now Available
32K TAPE - $14.95
32K DISK - $19.95
From Your Local Computer Store
Or Send Order to:

© Copyright 1980
Adventure INTERNATIONAL
A DIVISION OF SCOTT ADAMS, INC
BOX 3435
LONGWOOD FLA 32750
(305) 862-6917
DEALER INQUIRIES INVITED

www.commodore.ca

A Model for Structured Programming

The Anatomy Of A Word-Research Processing Program for the APPLE

Derek A. Kelly

Sophisticated and broadly-ranging readers will hopefully forgive the biological analogy in the title. While we most usually look upon programs and APPLEs as simply robot-like physical mechanisms, I've found that I can't regard programs simply in terms of the static analogies of machines, but that I must also regard them in biological terms, terms more suitable to living creatures and their structures which may evolve over time. Computer programs and programming follow a recursive pattern, generally, in that the structure of a program, its design, and its joints, is never finished, but is constantly changing in response to new situations and requirements.

One of the aims of this article is to present a sort of broadly ranging tutorial on programming and system design. Another aim is to discuss the structure--and some of the programming features--of a program that I use frequently, and which other research-oriented, word processing, scholars, undergraduate or graduate, may also find to be useful. Since my program is not "finished" in the sense that while every routine now in it works, I am not satisfied with the routines I selected for inclusion, and am constantly adding new ones. So I'll be talking about the "anatomy" of a growing and evolving program, not about a static and finished product. After having explained the problem and design of the program, I think a new programmer will be able to take up where I left off, and either code the entire program by himself, or he will be armed with the tools with which to design and program his own version of this program.

By word-research, I mean simply research carried out by reading & studying books containing mainly verbal information.

Problem Definition and Analysis

Anyone who wishes to prepare programs in a "structured" manner will need to follow the steps of this article, or ones quite similar. The first step in program construction is not--unless only a short, experimental program is desired--to turn on the APPLE and begin to code in lines, but to think out the program ahead of time, in the head, and on paper. This thinking out ahead of time is not just a short-lived matter. It may take up to 50 or 60% of your programming time, leaving the rest of the time for actual coding and debugging.

The first decision I must make when constructing a program is: What does this program do? What is the goal of the program? What problem does this program solve? What practical applications does it have? What results do I want to achieve from a use of such a program?

Let me explain the alternatives and the decisions on these questions that I made for the program of this article.

Millions of college students--and a proportionate number of professors--periodically face the prospect of writing a "paper" based on original research, for various courses in the Humanities and Social Sciences, and less frequently in the Physical Sciences.

The papers written to report on research projects all share a common set of characteristics on the formal level: all include an alphabetized bibliography, a series of footnotes, basic divisions, and a semi-standard format. A computer program could assist the student in conceiving a research project, organizing and designing a research report, assist him in gathering references and bibliography, and finally, can be of help to the writer in those final, frantic, hours when some order and organization must be imposed on the hundreds of 3x5 inch cards sitting in heaps on the floor!

Such a program could serve as a computerized version of the standard classic of academic writers, **The Manual of Style.** It could also serve the pedagogical purpose of instructing people on how to organize and implement a research project, a skill that is not all that prevalent on most campuses. In addition, as already said, the program can also serve as a tutorial. This article-program package has been written with these possibilities in mind. And the program itself assists the researcher-writer in all of the ways mentioned above, and more.

www.commodore.ca

Anatomical Design of Word-Research Processing Program

Now that I have an idea about the general problem, and the goals and results of the program that I have as an idea in my mind, I must develop a design of the basic components and parts of a program that does what I want it to do.

What are the basic components? What basic functions do I want to appear on the main menu of the operating program?

Notice that even at this stage I am concerned with how my program "screens" will appear. This is an important consideration for what is called structured programming. Structured programming has as one of its main goals the integration of all the phases and parts of a program into a comprehensive view so that things will "hang together" better in a working program. In this particular case, it is good to try to "en-VISION" how an idea will look when it is coded and appears on the monitor of your comupter.

On the screen, my menu will appear as follows:

Operational Choices:
- **I. Conceptualizing of Research routines**
- **II. Documentation**
- **III. Organizing and Writing-assistance routine**

These three are the basic components or modules of the program. I chose these as the components since these are the three basic steps in any research program, and I could thus gather the routines that work for each stage together, work on each one separately, and develop the main organization of the layout of the finished program in my APPLE's memory.

When conceptualizing research projects, certain functions are constants. We must conceive of a topic, plan and organize a project, plan and organize a report, and generally keep these two structures in mind simultaneously.

When documenting research, we need to be able to keep track of authors, books, notes, comments, and the alphabetization of a bibliography.

Finally, when organizing and writing a report, we need to be able to sort out our notes and comments into manageable sections so that a smooth flow of writing plus documentation can be maintained, and we need constantly to be able to check back to previous work, to previously written about notes and comments. These functions are performed in the third module of the program.

Having decided on my main modules, I now need to decide on what functions each will perform. That is, I need the menus for the three modules. These menus will appear on the monitor whenever I select one or another of the main modules when the program is RUNning.

Module I Menu: Conceptualizing & Planning
1. **Built-in model project design**
2. **Assisted development of researcher's own project model**
3. **Built-in model of paper**
4. **Researcher's model of paper**
5. **Model of the entire structure of both project & paper**

Module II Menu: Documentation
1. **Author/title listing**
2. **Author/title search**
3. **Notes**
4. **Notes and comments by book**
5. **Alphabetized bibliography**

Module III Menu: Organizing & Writing
1. **Pre-ordered notes & comments**
2. **Outline + Notes & Comments**
3. **Freeform organizations**

Another aspect of structured programming which is worth mentioning is that the better structured a program is, the more of its aspects there are that interlock and support each other. Since programs have logical, physical, and RUN features, a structured program is one where these three kinds of features support each other. Take a simple example. On RUN, Applesoft BASIC jumps to the earliest line number of the program, and works its way through the program from there. In addition, every time a GOSUB is executed, the APPLE's BASIC jumps back to the earliest line number and scans for the line number of the GOSUB in question. Now, if it takes time to make these scans, wouldn't it make sense to put the most frequently called GOSUBs on the earliest line numbers? And if this were done, wouldn't this be the matching up of the machine's RUN features with the physical layout of the program? Wouldn't it be more economical in time, for the machine to be able to find most called GOSUBs on the earliest lines of the program?

Notice here that if this suggestion is followed, then the format of most programs in BASIC on the APPLE would have to be revised. Instead of putting the Main Program at the early line numbers, and the subroutines at the highest line numbers, it would be better to invert this structure and to place subroutines on early lines and infrequently called Main Programs on the highest line numbers. This is what I have done in the present program.

While I try to make the physical layout of the program match up with the way the machine works, I also try to get these two to match the logical structure of the program. These three different features that match can be visualized by the diagram below:

Features of:

BASIC	Physical Layout	LOGIC
Line # 1000	Data	Information/Research
Line # 900	Main Program & Menu	Basic steps in Research
Line # 800	Conceptualizing Module	Subprogram #1
Line # 700	Documentation Module	Subprogram #2
Line # 600	Organizing Module	Subprogram #3
Line # 500	Built-in Data	Subprogram #1
Line # 400	Data Read subroutine	Subroutine to Main program
Line # 300	Printing subroutines	General use subroutines
Line # 200	Control subroutines	General subroutines
Line # 100	Menu subroutines	General subroutines
Line #0- 99	Micro subroutines	Most frequently used subroutines

www.commodore.ca

There is yet a fourth thing correlated with these three, and that is the process whereby a research paper is written. The program is so structured that while one may switch at will from one routine or module as frequently as one wishes, nonetheless by following the RUN of the program and choosing each menu selection sequentially, then one will be simultaneously undertaking and following the actual steps that need to be taken from conception to finished paper. This will be taken up further in the next part.

Thus far we have--on paper--the basic outline of the program: we know how the program will look organizationally when it is typed into the APPLE, we know how efficiently that layout is likely to work, and we know the logical structure of the program.

Constructing the Program

Before proceeding to map out the flowcharts of the program, and beginning to code, we need to understand the basic flow of work in researching that the program outlines. Whenever someone has a research project to undertake, there are seven steps or stages in the process. A program intended to help someone along this process should mirror that process in some way. Here are the steps:

1. *An assignment or idea for a project arises in the mind.*
2. *How to formulate the idea in words?*
3. *How to design an outline of the parts of the idea?*
4. *How to construct the report of the project?*
5. *How is the data structured logically and sequentially?*
6. *Gather notes & documentation, and store together.*
7. *Organize research.*

A suitable research-assistance program would be one that was able to formalize, that is, put into an algorithm, the little steps involved in each of these steps, and to find ways of aiding the process along. Now is the time to start coding the program, and developing the routines to carry out all the tasks that will be required.

Coding and keying in the basic outline of the program as depicted in the correlated features diagrammed earlier is an easy task, relatively speaking. I program the way some painters paint--I begin with the broad strokes (lines of code) and work progressively and sequentially on more detailed features. Thus when I begin coding, my first few lines of code will map out the outline of the program:

```
1 REM BY D. KELLY
2 GOTO 900: REM TO MAIN PROGRAM
3: :
4 REM MICRO SUBROUTINE LOCATION
99: :
600 REM MAIN SUBMODULE #1
698 RETURN
699: :
700 REM MODULE #2
798 RETURN
799: :
800 REM MODULE #3
898 RETURN
899: :
900 REM MAIN PROGRAM
998 END
999: :
1000 REM DATA
```

Note that this is already a working program. If I added just one line, 910 GOSUB 600: GOSUB 700: GOSUB 800, the program will run through it's paces, in exactly the way it will do it when all the routines are coded in and the program approaches 1000 lines of code or more. All I will in effect do to this outline, from here on out, is fill in the details. If you key in the above program and RUN it, nothing will appear on your screen except the cursor. So add the following lines at 601,701, and 801: PRINT "XXXXXXX": Get G$. Each GOSUB on line 910 will now pause for you to hit a key before proceeding to the next GOSUB. This is how you proceed in the program coding: you imagine the steps in research, and you devise coded programs to implement them. You also visualize how you want the screen to appear when the program RUNs.

Since words and characters will be appearing on the screen, it will always be necessary to have lines of code that print out lines of various characters, so lines 4-10 can be filled with one-line subroutines that can be called whenever I want them in the program. Since they'll be called frequently, I'll put them on the earliest lines, though I may reserve line 4 for *the* most frequently used subroutine in my programs. So this little block of code may appear as follows:

```
4 PRINT "?-";: GET G$: RETURN
5 FOR A = 1 TO 40: PRINT "%";: NEXT A: RETURN
6 FOR A = 1 TO 40: PRINT "-";: NEXT A RETURN
8 FOR A = 1 TO 40: PRINT " + ";: NEXT A RETURN
9: :
```

The function of line #4 is to give me a way of controlling the process and flow of a RUN, enabling me to make immediate choices. Using the above subroutines, I can improve on my main program, and on a RUN by adding and revising the following lines of code. Key them in now, and try them for yourself.

```
900 REM MAIN PROGRAM
902 HOME: GOSUB 5
904 GOSUB 800: GOSUB 4
906 GOSUB 700: GOSUB 4
908 GOSUB 600: GOSUB 4
910 PRINT "WHAT NOW? 1 = END:2 = REPEAT:"
912 GOSUB 4
914 IF G$ = "1" THEN END
916 IF G$ = "2" THEN GOTO 902
```

At this point, I am ready to get down to the task of keying code for the menu-screens of the three main modules, and to develop the code for the main menu itself. After that is done, I will be ready to turn to the task of developing the code to perform actually useful functions.

You can make the choice of placing your menus in the modules or as calls to subroutines located elsewhere. Thus the main menu can be keyed in

www.commodore.ca

somewhere between 900-999, or a call can be made to it if you locate it between lines 100 and 599.

Having served the purpose of illustrating a point, lines 902 to 916 can be erased (DELeted), and programming decisions made.

Since I am not yet ready to list and DIMension my variables, and want to locate these in my main program, I will skip 20 lines and begin my main menu at say 920. This main menu will be the point to which the program will always finally return for further user choices. The complete Main program plus main menu appear below:

```
900 REM MAIN PROGRAM
902 REM DIMENSIONED VARIABLES
919: :
920 REM MENU
921 HOME: VTAB 2: GOSUB 8: GOSUB 6
922 PRINT TAB(10) "MAIN MENU:"
924 PRINT TAB (12) "1. CONCEPTUALIZING
OPERATIONS"
925 PRINT TAB (12) "2. DOCUMENTATION"
927 PRINT TAB (12) "3. ORGANIZING & WRITING"
929 GOSUB 6
930 PRINT "WHICH DO YOU WANT?": GOSUB 4
931 IF G$ = "0" THEN END
932 IF G$ = >4 THEN GOTO 920
933 ON VAL(G$) GOSUB 800,700,600
940 GOTO 920
950: :
```

Using the well-known "top-down" strategy of structured programming, I have located my main program at the "top" of the program, as befits its status as the logical top of the program. Now I can proceed on my way down to the next level, in this case the menus for each of the three modules, then following that, the specific operations. Each of the three modules will have programs that resemble the main program and its menu. In its turn, each of the menu options in the modules will be located on the lower line numbers as subroutines of various sorts, called from the module menu. In this way I will be using the top-down approach by using the motto: Never call a GOSUB from a place with a lower line number than the location of a subroutine. The program line 100 GOSUB 2011 is inefficient & badly constructed, while the line 2011 GOSUB 100 is well structured.

With the Main Program in hand, you can either DIMension variables on 902-919, or go on to code the module control programs. Then one could move to the development of each of the routines in each of the models. Assuming that these tasks are finished, the next coding step is to begin work on the 13 main routines in the process of researching, beginning with the first:

Three Modules & Thirteen Subparts

I. Conceptualizing routines

1. Model of project
2. Present project
3. Model of research report
4. Present research report
5. Data organization model

Computer Supplies for Small Computer Systems

Commodore Printer Ribbons IN STOCK

Our full line of supplies includes:
Dikettes (3M, basf)
Diskette retrieval devices
BPI Business Package for the Commodore
Tractor Printer Paper
Tractor Printer Labels
Printer Ribbons for NEC, Commodore and Xymec
Dust Covers
Adapters
Uncrashers
Computer Desks
NEC Spinwriters
Print Thimbles
Xymec Intelligent Printers
"New Dealer" Kits

ZEIGLER ELECTRONIC PRODUCTS

DEALERS:
We no longer market COMMODORE Equipment. (Jerry Zeigler is with Commodore) but Connie and I have the computer supplies you need to support Commodore System Sales.

Give us a call:
Trudie Zeigler
or
Connie Rittenberry
(404) 289-1596 or 289-2265
3661 CALUMET RD.
DECATUR, GEORGIA, 33034

www.commodore.ca

II. Documentation Routines
1. Author/title list
2. Search
3. Notes print
4. Notes & Comments print
5. Alphabetized bibliography

III. Organizing
1. Pre-Ordered method
2. Re-Ordering
3. Deadline pressure method

Proceeding in a structured manner, my first task is to locate these thirteen routines. Calls to these thirteen routines will be made *from* lines 800,700, or 600. Since the module control programs are as short, or shorter, than the main program, I reason that lines 820-899,720-799, and 620-699 are empty and can be used to locate the routines.

That done, my next task is to determine what DATA will be READ and PRINTed out in a RUN of the program. Items I.1,I.3, and I.5 each involve data that must be READ and printed out. This data will have to be an integral part of the program, and not modifiable data, so I place this unmodifiable data on lines 500-599. The data on line 1000 onward is modifiable data, data on books and notes and comments entered by each individual user. The data that will be READ and manipulated in the Documentation routines will involve data drawn from line 1000-. The organizing routines will involve both changeable and unchangeable data drawn from both locations.

I now know that each of the thirteen routines will have the same structure: (1) An Option selection entry, (2) A READ DATA function, and (3) a PRINT data function. That being so, can I use the same subroutine to handle each of these three phases for all thirteen? If not thirteen, can I use just three versions of the option-read-print triad? Or do I have to use thirteen different traids? Obviously, using just one triad would be easier.

So I set about seeing how far I could stretch one of the option-read-print triads. First, I tried out various forms of the DATA...READ phase of the triad. Since the computer needs the Data to do its work, I decided to dispense with storage problems, and ways of accessing through a floppy-disk data only as needed. I decided to have one routine that would READ all of the data in one stroke. That left me with the problem of finding a way to standardize the **option-print** aspects of the triad.

The options function had to be done by several different routines as the options in the three main modules were not all of the same kind. On the PRINT routine, I managed to develop two slightly different routines to handle that feature. Instead of thirteen worst-case models of the option, read and print routines, and instead of the best-case situation of one triad to handle all thirteen, I end up with a middle position: 2 Print + 1 READ + 4 Option routines. The present program is five times more efficient than the worst-case program, but it is also 2.3 times less efficient than the best-case program. In an ideally structured program, the best-case program would be achieved in all cases. It should be obvious by now that even the most inexperienced structured programmer will still, if he should have followed me to this point, be able to construct programs that approach best-case rather than worst-case programs. He will be able to use routines at many different points, many different times, in a program. He willknow how to make routines that are simultaneously *general* and able to handle a great many different functions--as line 4 functions in this program--and at the same time able to handle the affairs of each place from which it is called as if it had been created just for that one call. Ideally, program routines and models should be simultaneously general *and* specific.

Consider the I.1 routine on the model project. While it would be useful to have put in a Critical Path determination subprogram to chart the way to the finished project, I used a simple linear model of the project with twenty steps from start to finish. Let's use these as data:

Line 502 DATA 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17, 18,19,20.

Let's assume that you want to print out these steps using the standard academic sequence of : I,A,1,(a),(1). These need to be added to data, so at line 501 I put: DATA I,II,III,IV,A,B,C,D.

Now we need to put the data on lines 501, and 502 together in a printout on the screen of the two data lines combined. Below is a short program that will printout something like this:

```
I. 1
     A. 2
     B. 3
     C. 4
     D. 5
II.6
```

```
210 FOR I = 1 TO 4: READ S$(I): NEXT
211 FOR I = 1 TO 4: READ P$(I): NEXT : DIM D$(20)
212 FOR I = 1 TO 20: READ D$(I): NEXT
213 FOR I = 1 TO 4: PRINT S$(I)".";:GOSUB 60
(60 PRINT D$(I): I = I + 1:RETURN)
214 FOR J = 1 TO 4: PRINT TAB(3)P$(J)".";:
GOSUB 60
215 NEXT J: GOSUB 4: NEXT I: RETURN
```

This little printing routine does not include an option routine, so you may want to add one. If you have two options, then you'll need two subroutines like the one at 60. So add:

```
205 PRINT "OUTLINE(1):or:FLOWGRAPH?";:
GOSUB 4
206 IF G$ = "1" THEN GOSUB 210
207 IF G$ = "2" THEN GOSUB 220
208 END
•
•
•
•
220 FOR I = 1 TO 20: PRINT I"."S$(I):GOSUB 70
(70 PRINT TAB(20)"1": PRINT TAB(20)"1":
```

www.commodore.ca

```
PRINT TAB(20)"V": PRINT: RETURN)
221 GOSUB 4: NEXT I: RETURN
```

This little program will RUN, though it does have bugs in it as it stands now. For instance, by adding an option, the data should be read *before* either of the options is exercised, i.e., it should be done before line 205. In that case, you could add a GOSUB 210 to line 205, and a RETURN at the end of line 212, thus turning lines 210-212 into a callable subroutine. Then line 206 would have to be changed to : 206 IF G$ = "2" THEN GOSUB 213.

It is in developing these thirteen routines that creativity may enter in. The programmer will want to try out different ideas of what the first of the thirteen steps should be, or of what this particular Model Project step needs to do to help the researcher. Of course, the exact coding steps may have been developed ahead of time, in which case the programmer cannot experiment, but must simply follow the line laid down by his flowcharts, etc.

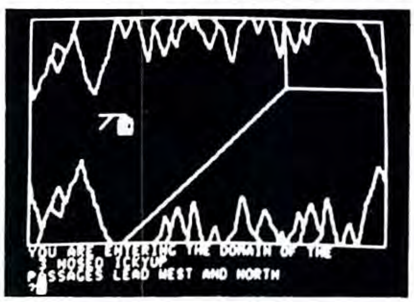
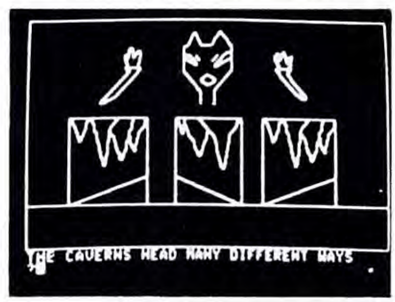
As one goes sequentially through the list of thirteen functions, only a few variations are apparent: I.2 & 4 use INPUT instead of READ data; II.2 requires a "search" routine, while II.5 requires an alphabetization routine (a sort). Other than that, the programming should be a relatively simple matter of coding in the seven variations in the **option-read/input-print triad.**

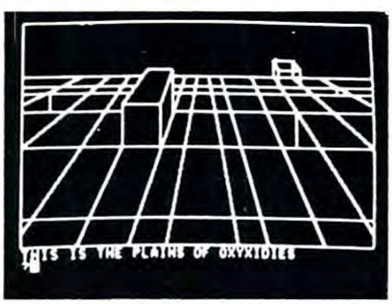
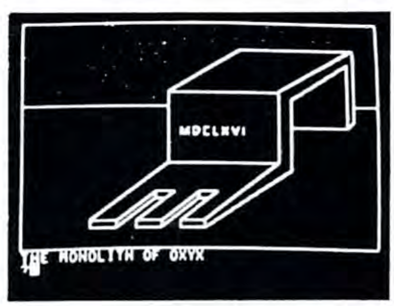
After having completed this step in the programming effort, the final step is to tidy-up, debug, and put the program to use.

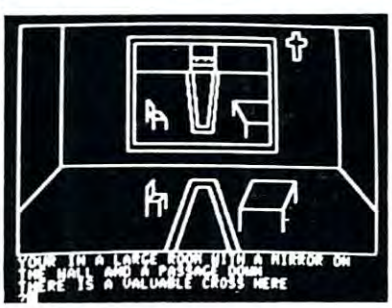
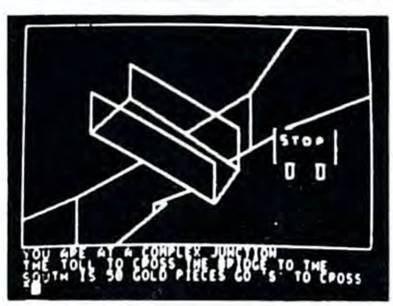
Implementation and Use

While I do have a couple variations on this program that I can supply (together with a manual) for a nominal charge, my aim is rather to get you, the reader, to develop your own variant of this word-research processor program. An even more basic aim is to get programmers to see the visual (aesthetic), as well as practical, benefits in structured programming, and to encourage the use of such a versatile and task-simplifying method of programming & coding.

I don't mean to leave the reader with the impression that all programming should be structured programming. Programmers should also take every opportunity to explore and experiment with ideas and try to get them into programs. Sequential and other tools like the GOTO can be used in these cases. But when the program is long, complex, or growing, then GOSUBs and other structured programming tools should be used. ©

Oldorf's Revenge

WIZARD I

WIZARD 1 is a well done and exciting Hi-Res game using over 100 Hi-Res pictures. WIZARD 1 requires 48K, Applesoft Rom, and Disk. As you explore the caverns and castles (each locale is done in Hi-Res) looking for treasure, you must battle the one-eyed, two-thumbed torkie; find the grezzerlips' sword; visit the snotgurgle's palace and get through the domain of the three-nosed ickyup plus more!

Wizard 1 on Disk.................$19.95

SEE YOUR LOCAL DEALER
OR SEND CHECKS TO

HIGHLANDS COMPUTER SERVICES
14422 S.E. 132nd
Renton, Washington 98055
(206) 228-6691

Washington residents add 5.3% sales tax. Applesoft and Apple are registered trademarks of Apple Computers, Inc.

www.commodore.ca

Hard Disks For The APPLE

Philip Castevens

One thing seems obvious about this microcomputer business - it is very much a function of time. Back in April 1980, I purchased what I thought at the time was one of the most advanced micro-systems available: an Apple II Plus, two Disk II drives, and a Paper Tiger (IDS 440) printer.

Well, that was "way back then," and what was "most advanced" then is not necessarily "most advanced" now (four months later). My ego suffers a little when I think of the Apple III with 128K of main memory, an 80-column monitor, twice the speed, a numeric Keypad, and a correspondence-quality IDS 460 printer.

But, I really should not complain. This experience helped me to remember that over-involvement in material concerns such as these ultimately leads to disappointment. And, I have not nearly explored fully the potential of my current system.

And what about hard disks? It is also very difficult to draw a bead on this rapidly moving target. However, since I am involved in developing business systems for the Apple, I have done some research into this situation and would like to share my information with you. Keep in mind that I do not have any personal experience with any of the drives mentioned below and that these comments are my opinions concerning the information I have gathered from magazines, brochures, conversations, etc.. In many ways this is a beginners perspective.

The Past

The following is a brief list of the milestones leading up to the development of hard disks for the Apple:

YEAR	EVENT
1939	Work began on the Mark I (proposed by Howard Aiken), the first operational automatic computer.
1943	The ENIAC (an electronic version of the Mark I) project began.
1959	Transistors began to replace vacuum tubes. The beginning of "second-generation" machines.
1965	"Third-generation" computers (e.g., IBM 360) became available. Featured miniaturized circuits, data communication, etc..
1970	Intel develops the first microprocessor chip.
1975	Steve Jobs and Steve Woznak invent the Apple.
1979	Corvus Systems, Inc. introduces a 10 Megabyte (10M) hard disk for the Apple.

The Present

So here we are in 1980 and there are several hard disk systems for the Apple.

CORVUS 11AP

Corvus is into this thing in a big way. They are doing lots of promotion and offer related systems such as The Mirror (to back the 11AP up on tape) and The Constellation (multiplexer that allows simultaneous access by multiple Apples). Also, there seems to be more software developed by independent sources for use with the 11AP than is available for the other hard disks.

Physically, the Corvus 11AP has the approximate size and shape of a shoe box (but is heavier). It is a high-performance "Winchester" drive and is not removable. It is controlled by a Z-80 processor (which can handle up to 4 disks) and comes with a standard interface card which plugs into slot 6 of the Apple.

Both DOS (for BASIC files) and PASCAL are available. Under the DOS interface, the disk is formatted into multiple images of a standard diskette. This makes it necessary for the program to specify which "volume" to access when dealing with files on the 11AP. Since there can be as many as 82 volumes on-line simultaneously, this could prove awkward unless a file management system such as "Corvus FMS" (see "SOFTWARE" section below) is used. This is not a problem with PASCAL and the drive can be configured as one 10M volume. Cost is $5350 with controller and Apple interface.

LOBO Model 1850

LOBO recently introduced the Model 1850 which "consists of an 8-in. floppy disk drive and an 8-in. fixed Winchester drive housed in the same cabinet. The two drives share the same power supply and disk controller, with the floppy acting as backup medium for the fixed-disk drive." The hard disk can be either 5M or 10M. The floppy drive "is available in a maximum configuration of 1.6M bytes" (COMPUTERWORLD, 6-16-80, pp. 69-70). With prices ranging from $3495 to $4695, this is very attractive.

WIZARD 10

This is a 10M Winchester drive which can be formatted as one file and comes with controller and interface for the Apple. Cost is $4975.

CAMEO DC-500 Cartridge Disk

This is a 10M hard disk system offered by Cameo Data Systems, Inc.. The big advantage here is that it consists of a 5M fixed portion and a 5M removable portion. The removable cartridges can be used for backup purposes, to switch application, and for archival storage. Cost is $5995.

The volume approach to DOS files is similar to Corvus and Cameo is reportedly working on a PASCAL interfacce. They offer an interesting 10-day free trial program.

www.commodore.ca

www.commodore.ca

CORVUS Mirror

Earlier this year, Corvus announced The Mirror to solve the backup problem for their hard disks. It works in combination with a video cassette recorder and can store up to 100M on one cassette. Cost is $790 in addition to the cost of the recorder. It takes only 10 minutes for the complete backup process.

CORVUS Constellation

The Constellation multiplexer allows from two to 64 Apples to be linked together and share up to 40M. Cost is $750 plus $235 for each additional interface.

MECA Tape II System

This system includes the hardware and software interface for their Beta-1 Cassette Tape System. It allows random access to 500K bytes per drive. A master and from one to three slave drives are available.

MECA promotes their product as (1) a backup system for disks and (2) a disk replacement. Under $1000.

NESTAR Cluster-One (Model A)

This allows up to 64 Apples to ''share data, access the same files and communicate with one another at distances up to 1000 ft.'' (Computerworld, 2-4-80, p.59) and to use the same peripherals. The Cluster-One comes with either 1.2M 8-in. floppies (double-sided) or 16.5M Winchester-type hard disk subsystem. Prices start at $6000.

Software

There are at least two companies offering access methods for the Corvus 11AP. Alpine Software has developed ''Corvus FMS'' ($395) which is like an advanced DOS that includes an indexed sequential access method (ISAM) for communicating with large Applesoft files. It also minimizes unused space within files. United Software of America sells ''KRAM'' (keyed random access method) which works with integer BASIC files and costs $99. Peripherals Unlimited is developing software for the Cameo DC-500 system.

Many products, such as Datacopes Single Disk Sort ($50) will work with hard disks like the Corvus, but only one volume at a time. In fact, it seems that, in general, sorting large files on one of these hard disks is likely to take a long time. This is because of the speed and main memory limitations of the Apple II. The Apple III should work better in this respect.

Note that software is not as sensitive to the passage of time as is hardware. The same software -especially ANS COBOL and UCSD Pascal programs - can be used within many different hardware configurations with minimal conversion.

The Future

Shugart Technology has recently announced a 5.25-in. 6.3M Winchester hard disk drive which sells for under $1000 each in large quantities. Perhaps this unit will become available for use with Apple processors.

It has been rumored that Apple Computer itself may come out with a hard disk of some sort (perhaps as a component along with the Apple III in some business-oriented system).

Venture Development Corp. predicts the rate of shipments of ''low-cost, low-performance ... Winchester drives ... will reach 375,000 units in 1984'' and that ''shipments of high-performance 8-in. fixed drives will reach a level of 54,000 units, while 8-in. cartridge drives will be selling at twice that level.'' (COMPUTERWORLD, 2-18-80, p. 66)

Personally, I think that a pair of small, low-cost 5M cartridge drives would be a hot-selling addition of Apple systems.

Summary

I do not want to over-emphasize either the products currently available or those which seem likely to become available in the future. For someone considering the purchase of hard disks for the Apple, the decisions should be based on past, present, and future considerations. A balanced approach such as this will probably yield the most satisfaction.

Addresses

ALPINE SOFTWARE
4874 Ridenour
Colorado Springs, CO 80916

APPLE COMPUTER INC.
10260 Bandley Dr.
Cupertino, CA 95014

CAMEO DATA SYSTEMS INC.
1626 Clementine
Anaheim, CA 92802

CORVUS SYSTEMS INC.
2029 O'Toole Ave.
San Jose, CA 95131

DATACOPE-PO DRAWER AA
Hillcrest Station
Little Rock, AR 72205

LOBO DRIVES INT.
935 Camino Del Sur
Goleta, CA 93017

MECA
7026 O.W.S. Road
Yucca Valley, CA 92284

NESTAR SYSTEMS INC.
430 Sherman Ave.
Palo Alto, CA 94306

PERIPHERALS UNLIMITED
6012 Warwood Road
Lakewood, CA 90713

SHUGART TECHNOLOGY
340 El Pueblo Road
Scotts Valley, CA 95066

UNITED SOFTWARE OF AMER.
750 Third Avenue
New York, NY 10017

VENTURE DEVELOPMENT CORP
One Washington Street
Wellesley, MA 02181

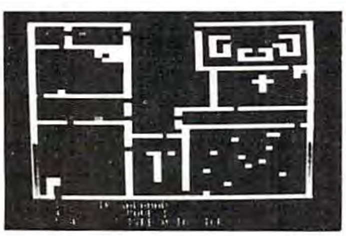
WIZARD
COMPUTER DISTRIBUTORS
P.O. Box 9194
Austin, TX 78766

©

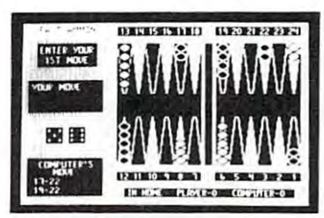
www.commodore.ca

Exciting, entertaining software for the Apple II and Apple II Plus*

If you liked "Invaders", you'll love ASTEROIDS IN SPACE by Bruce Wallace. Your space ship is traveling in the middle of a shower of asteroids. Blast the asteroids with lasers, but beware — big asteroids fragment into small asteroids! The Apple game paddles allow you to rotate your space ship, fire its laser gun, and give it thrust to propel it through endless space. From time to time you will encounter an alien space ship whose mission is to destroy you, so you'd better destroy it first! High resolution graphics and sound effects add to the arcade-like excitement that this program generates. Runs on any Apple II with at least 32K and one disk drive.
On diskette — **$19.95**

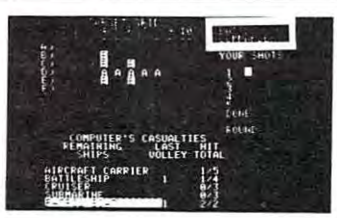


FRACAS™ by Stuart Smith. A fantastic adventure game like no other — up to eight players can participate in FRACAS at the same time. Journey in the land of FAROPH, searching for hidden treasure while warding off all sorts of unfriendly and dangerous creatures like the Ten Foot Spider and the Headless Horseman. You and your friends can compete with each other or you can join forces and gang up on the monsters. Your location is presented graphically and sound effects enliven the battles. Save your adventure on diskette or cassette and continue it at some other time. Requires at least 32K of RAM. Cassette: **$19.95** Diskette: **$24.95**



FASTGAMMON™ by Bob Christiansen. Sound, hi res color, and cartoons have helped maked this the most popular backgammon-playing game for the Apple II. But don't let these entertaining features fool you — FASTGAMMON plays serious backgammon. Requires at least 24K of RAM.
Cassette: **$19.95** Diskette: **$24.95**

BATTLESHIP COMMANDER™ by Erik Kilk and Matthew Jew. A game of strategy. You and the computer each start out by positioning five ships of different sizes on a ten by ten grid. Then the shooting starts. Place your volleys skillfully — a combination of logic and luck are required to beat the computer. Cartoons show the ships sinking and announce the winner. Sound effects and flashing lights also add to the enjoyment of the game. Requires at least 32K of RAM.
Cassette: **$14.95** Diskette: **$19.95**



QS

QUALITY SOFTWARE

6660 Reseda Blvd., Suite 105, Reseda, CA 91335

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. MasterCharge and Visa cardholders may place orders by telephone and we will deduct $1 from orders over $19 to compensate for phone charges. Or mail your order to the address above. 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 — payable in U.S. currency.

*"Apple II" and "Apple II Plus" are trademarks of Apple Computer, Inc.

www.commodore.ca

THE STAR MODEM

From Livermore Data Systems

IEEE 488 MODEM
SALE $288

RS232 MODEM
SALE $139

The STAR modem from Livermore represents a significant breakthrough in the development of acoustic modems. The small, lightweight case houses a high-performance modem that competes with the highest quality standard-sized couplers available. Yet, because of its costs effective design, the STAR has become the price/performance leader in the industry.

CIRCUITRY

The switchable, four-section bandpass filter provides the user with excellent out-of-band rejection to assure accurate processing of the received carrier, even at signal levels of less than –47 dBm. Further, the proven soft limiter and phase lock loop discriminator yields data that is essentially jitter free.

The oscillator is built using highly stable, state-variable circuitry that delivers a nearly harmonic free, phase coherent sine wave to the telephone network, assuring compatibility with all other 103 type modems. Because of the pureness of the sine wave, the STAR modem exceeds even the stringent harmonic requirements of all CCITT countries.

CARRIER DETECT

To assure accurate teleprocessing connections, the carrier detect circuitry prevents the modem from attempting to operate when excessive noise would produce errors or cause marginal operation. The circuitry also has a special amplitude sensor that prevents chatter when the received signal fades.

EXCLUSIVE ACOUSTIC CHAMBERS

The exclusive triple seal of Livermore's new flat mounted cups locks the handset into the acoustic chamber yielding superior acoustic isolation and mechanical cushioning. Designed to adapt to most common handsets used throughout the world (also fits GTE handsets), the STAR offers the utmost in flexibility and transmission reliability.

SELF TEST

The self test feature on the STAR allows the user to verify total operation of the acoustic modem by using the terminal in the full duplex mode. No need for remote assistance in diagnosing terminal or modem products.

MIPLOT Intelligent Plotter
by Watanabe Instruments (Digiplot)

SPECIAL $1045

- Incorporates all intelligent functions required for producing graphs and drawings including 8 vector and 4 character commands.
- Distance accuracy within 1%, repetition accuracy within .01 inches, programmable step size .004 inches, internal interpolation in .002 steps.
- Solid and broken line types can be specified.
- Character generator for letters, numerals and symbols. Characters can be enlarged and rotated in four orientations.
- Coordinate axes can be drawn by simply specifying the graduation interval and number of repetitions.
- Self test mode automatically draws complete test pattern.
- Printer mode outputs character data in 16 sizes with 4 orientations.
- Connection to any microcomputer using parallel 7-bit ASCII code.
- Can use simple cable to parallel port, or special interface to IEEE or other.
- Modular control circuit and mechanical construction.
- Uses any hard fiber-tip pen.
- Uses 11 x 17 paper.

Includes power supply, I/O connector, 2 pens, 50 sheets of paper, and complete manual.

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	6.90	10/6.50	50/6.10	100/5.70
6532	7.90	10/7.40	50/7.00	100/6.60
2114-L450		4.45	20/4.25	100/3.95
2114-L300		5.65	20/5.35	100/4.95
2716 EPROM		13.45	5/12.75	10/11.85
2732 EPROM				39
4116-200 ns RAM (NEC)				**8 for 39.00**
6550 RAM (PET 8K)				12.70
S-100 Wire Wrap $2.65				Solder Tail 2.15

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.

ATARI 800 $777

All Atari Modules 25% OFF

SPECIAL–purchase ATARI 800, receive extra 8K memory FREE.

EDUCATIONAL PLAN - buy 2 ATARI Computers, receive 1 ATARI 400 FREE!

DISKS

(write for quantity prices)

SCOTCH (3M) 8"	10/3.00	50/2.85	100/2.75
SCOTCH (3M) 5"	10/2.95	50/2.80	100/2.70
Maxell 5"	10/3.65	50/3.40	100/3.15
Maxell 8" Double Dens	10/4.10	50/3.95	100/3.80
Verbatim 5"	10/2.40	50/2.35	100/2.30
(add 1.00 for plastic storage box)			
BASF 5"	10/2.40	20/2.35	100/2.30
BASF 8"	10/2.40	20/2.35	100/2.30
Diskette Storage Pages			10 for 3.95
Disk Library Cases		8" – 2.85	5" – 2.15

Commodore
CBM-PET SPECIALS

FREE Up to $235 free merchandise with purchase of one of following CBM-PET items!

		FREE
8032 32K - 80 column CRT	$1795	235
8016 16K - 80 column CRT	$1495	205
8050 Dual Disk Drive - 950,000 bytes	$1695	220
CBM Modem - IEEE Interface	$395	50
CBM Voice Synthesizer	$395	50
8N full size graphics keyboard	$795	100
16K Full Size Graphics or Business Keyboard	$995	150
32K Full Size Graphics or Business Keyboard	$1295	205
2040 Dual Disk Drive - 343,000 bytes	$1295	205
2022 Tractor Feed Printer	$795	100
C2N External Cassette Deck	$ 95	12
Used PETs (8, 16, and 32K)		CALL

WRITE FOR SYSTEM PRICES

EDUCATIONAL DISCOUNTS

Buy 2 PET Computers, get 1 FREE

CBM Full Size Graphics Keyboard	$ 74
WordPro I - for 8K PET	$ 25
WordPro II - 16K CBM, disk, printer	$ 85
WordPro III - 32K CBM, disk, printer	$170
WordPro IV - 8032, disk, printer	$255
VISICALC for PET (Personal Software)	$170
BPI General Ledger, A/P, A/R for PET	$270
Programmers Toolkit - PET ROM Utilities	$ 36.90
PET Spacemaker Switch	$ 22.90

Dust Cover for PET	$ 7.90
IEEE-Parallel Printer Interface for PET	$ 65.00
IEEE-RS232 Printer Interface for PET	$149.00
Centronics 737 Proportional Spacing Printer	$800
NEC Spinwriter - parallel	$2450
XYMEC HI-Q 1000 Intelligent Daisy Wheel	$1975
Leedex Video 100 12" Monitor	$ 129
Zenith Z19 Terminal (factory asm.)	$ 739
Zenith Z89 with 48K	$2150
Strobe Model 100 Plotter	$ 600
SYM-1	$209
SYM BAS-1 BASIC or RAE-½ Assembler	$ 85
KTM-2/80 Synertek Video Board	$349
KIM-1 (add $34 for power supply)	$159
Seawell Motherboard - 4K RAM	$195
Seawell 16K Static RAM - KIM, SYM, AIM	$320
KL-4M Four Voice Music Board and Visible Music Monitor (4 voice) for PET	$59.90
MICR-REVERSI for PET by Michael Riley machine language version - you can't win at Level 5	$ 9.95
PAPER-MATE 60 Command PET Word Processor full-featured version by Michael Riley	$29.95

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

ALL BOOK and SOFTWARE PRICES DISCOUNTED

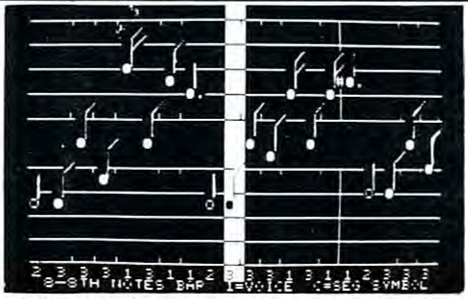
The 8086 Book (Osborne)	$12.75
Z8000 Assembly Language Programming (Osborne)	$10.60
PET Personal Computer Guide (Osborne)	$12.75
PET and the IEEE-488 Bus (Osborne)	$12.75
6502 Assembly Language (Osborne)	$ 9.90
Programming the 6502 (Zaks)	$10.45
6502 Applications Book (Zaks)	$10.45
6502 Software Cookbook (Scelbi)	$ 9.45

115 E. Stump Road
Montgomeryville, PA 18936 215-699-5826 **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.

www.commodore.ca



FOUR PART HARMONY FOR THE PET

A-B Computers announces a combination system consisting of the KL-4M DAC Board and the Visible Music Monitor for Commodore PET-CBM computers. The package enables PET users to easily create and play musical compositions of up to 4 parts.

The KL-4M Board includes an 8-bit Digital to Analog Converter, a low pass filter to eliminate high frequency computer generated hiss, and an on-board audio amplifier. An RCA-type jack is also included for quick attachment of your speaker. Amplification of the 6522 CB2 generated single note sound is incorporated as well, so that no additional hardware (other than a speaker) is required. Connection is made via the parallel and cassette ports. Both ports are extended with duplicate connectors (with keyways) so I/O capabilities are not reduced in any way. Board orientation is parallel to the back of the PET so additional table space is not required. The KL-4M is compatible with any of the 4 part music monitors, for which a number of precoded songs are available.

The Visible Music Monitor is intended to support 4-part harmony systems such as the KL-4M. Visible Music Monitor is written entirely in 6502 machine language. VMM provides an easy way to enter 4-part music. The user can see the notes on the screen as they are entered, and can make changes both with the insert and delete keys, and by using cursor up and down to "move" notes on the screen. Other features include "record changer" mode to load successive songs without intervention, user definable keyboard, and entry of whole notes through 64ths including dotted and triplet notes. Additionally, you can specify or change tempo, set key signature, and transpose at any time. Wave form modification makes it possible to create new instrument sounds. Voices can switch from one instrument to another or gang up on one instrument during the course of the song. Music can be played either with note display (useful for debugging songs), or with no display.

KL-4M Music Board & Visible Music Monitor Program $59.90

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

Full FORTH + $65.00

for APPLE or PET/CBM

A full-featured FORTH with enhancements. Fast and easy to use.

- conforms to FORTH Interest Group Standards
- cross compiler to produce either standard object modules or self contained ROMable modules
- built in macro conditional assembler
- complete string processing capabilities (including variable length strings)
- disk virtual memory for large programs (or small systems!)
- arrays (single or multiple dimensioned)
- full screen cursor controlled editor
- floating point and integer processing
- User's Guide and documentation

A product of IDPC Company

Requires 16K RAM and one disk drive.

FLOPPY DISKS

as low as $1.85

(write for price list)

3M

BASF

maxell

Verbatim

KMMM PASCAL for PET

A subset of standard Pascal (Jensen, Wirth) for PET

Includes following standard identifiers:

Types: INTEGER, BOOLEAN, CHAR, TEXT

Procedures: RESET, READ, READLN, REWRITE, WRITE, WRITELN

```
+ - * = <> < > <= >=
() [ ] { } or (* *) := . , ; '
```

div, mod, or, and, not , if , then, else, case, of, repeat, until, while, do, for, to, downto, begin, end, const, var, array, function, procedure.

Sequential I/O supported (both disk and tape)

Extensions:

Memory can be manipulated as an array (MEM)

% sign allows hex values in literals and I/O

Floating Point available in 4th quarter 1980 (nominal upgrade charge)

Included in package:

- Machine Language Pascal Source Editor
- Machine Language Compiler (generates P-Code)
- P-Code to machine language translator (produces optimized machine language object code —not just a P-Code interpreter).
- P-Code Interpreter (for debugging and learning)
- Run-time package
- User Manual
- Sample programs

Versions available for ROMs 2.0 (cassette only), 3.0, 4.0. Requires 16K minimum.

KMMM PASCAL on tape or disk
(with User Manual) $75
User Manual Only $15

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 ranoom 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.

DATA BASE and REPORT WRITER $45
DATA BASE and MAILING LABELS $45
DATA BASE, REPORT WRITER, and MAILING LABELS $60

Flex File available November, 1980. Specify machine size (32K recommended) and ROM type for both disk and computer.

EARL FOR PET

Editor, Assembler, and Relocator/Linker
Disk File Based for PET

Editor can edit files larger than memory.

Assembler features:

- 2 Pass Assembler
- Generates Relocatable Object Code
- Uses standard MOS Technology mnemonics
- Disk file input
- Listing output to screen or printer

Relocator/Linker capabilities:

- Relocates assembler output to desired memory location
- Resolves symbols declared external in assembly
- Links multiple object programs as one memory load

EARL for PET (with User Manual) .. $65
User Manual Only $15

115 E. Stump Road
Montgomeryville, PA 18936 215-699-5826

A B Computers

WRITE FOR CATALOG
Add $1.25 per order for shipping. We pay balance of UPS surface charges on all prepaid orders.

www.commodore.ca

THE ATARI® GAZETTE

Times Square On Your Atari

Neil Harris

This program is flashy, but it gets your message across. It creates a moving signboard, Times Square style, in the center of a marquee border. The message may be modifed to your own by simply changing the DATA statements. It makes use of a short machine language routine and a very interesting trick with the graphics.

The machine language portion is documented in the REMs at the beginning of the program. This routine simply moves the middle line of the screen one space to the left, leaving the two edge spaces of the marquee alone. While experimenting with this program I tried printing a "delete character" symbol at the start of the line, but this is not effective in graphics modes 2 or 3. Taking the characters in BASIC and POKEing them across was just too slow. The program itself takes the decimal values of the machine language code and POKEs them into memory. It must check the top of memory in order to determine where the screen is (lines 150-170).

The moving border of the marquee is where the trick comes in. A row of stars is drawn around the whole border, which you can see at the very start of the execution if you look closely. However, the first star is in color register 0, the next in register 1, and the third in register 2. Once the program begins, it flips the color values in these registers. Register 1 goes to black, and all the stars drawn with this register disappear. Register 1 goes to a color, and register 2 becomes a different color. Next, register 0 gets the color that was in register 1, register 1 becomes register 2's color, and register 2 becomes black and disappears. Only three statements are needed for all the stars to change color, and these execute so quickly that the eye cannot detect them.

The DATA statement at line 180 has two numbers: the number of strings in the message and the total combined length of these. The length together cannot exceed 256 characters. If the machine had string arrays this limit wouldn't exist, but it can't have everything.

The message is put on the screen character by character at the rightmost position of the text line. This is the sixth line down (of ten) and the 19th across (of 20). Line 720 contains the machine language call which moves the text line over. Incidentally, this overwrites the rightmost character.

Note very carefully the POKE instruction in line 690. This takes care of the problem wherein the Atari starts rotating the screen colors after the machine isn't touched for about seven minutes. This memory location contains the "attract mode" flag, which is simply a counter which increments every few seconds. When the count hits 128, the operating system starts its games with the colors, which is fine to save your TV from having a starbase permanently burned into the picture but does no good when you're changing everything yourself. By the way, it's called "attract mode" because in arcade games the machine goes into its special sales pitch to attract customers. Check out Space Invaders between games to see what I mean.

VARIABLES

I = general loop variable
M = page # for top of available RAM
A$ = message string
B$ = piece of message string, temporary
C$ = three asterisks of different color registers
C = a color # from 1 to 15, chosen at random in line 290
J = loop variable
K = loop variable, used with I and J for moving asterisks
Q = loop variable, causes stars to move twice for each letter
X = dummy variable returned from machine language
A = length of message

PROGRAM NOTES

10-105 DATA and mnemonics for machine language program
110-140 POKES machine language into memory
150-170 calculates value for SCRN and adds to program
180-220 DATA for message
230-280 load A$ with complete message
290-520 plots outside of screen in different asterisks
530-550 setup for register rotation
560-680 rotate registers, moving stars one space
690 resets "attract mode" flag
700 move stars one more space before moving text

www.commodore.ca

The ATARI® Tutorial

COMPUTER Calligraphy?

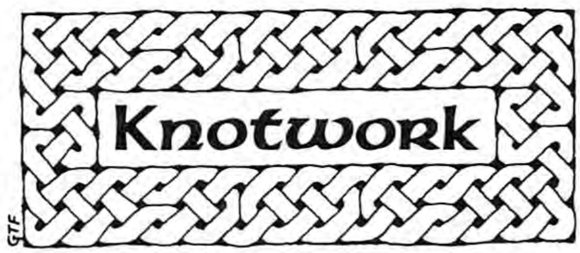
FONTEDIT

FONTEDIT

FONTEDIT

Well, not really! But with the FONTEDIT program in **IRIDIS #2** you can design your own character sets (or fonts) for the ATARI. For example, you can create a Russian alphabet, or APL characters, or even special-purpose graphics symbols. These special *fonts* can be saved on disk or tape for later use by your programs. FONTEDIT is a friendly, easy-to-use program: just grab a joystick and start designing.

With our KNOTWORK program, you can design patterns of *Celtic interlace*, (a technique used by 7th century Irish monks to illuminate manuscripts). After you have produced a pretty pattern on the screen of your ATARI, you can save it on disk or tape. As you might expect, KNOTWORK uses custom graphics characters that were created with FONTEDIT.



FONTEDIT and KNOTWORK are available *now* in **IRIDIS #2**, the second of our ATARI tutorial program packages. You get a C-30 cassette or an ATARI diskette with our excellent programs ready to load into your ATARI. Best of all, **IRIDIS #2** comes with a 48-page *User's Guide*, which gives clear instructions on how to use the programs. The *Guide* also provides detailed, line-by-line descriptions of how the programs work. (IRIDIS programs are written to be studied as well as used.) Our *Hacker's Delight* column explains many important PEEK and POKE locations in your ATARI.

The *User's Guide* also includes *Novice Notes* for the absolute beginner. We don't talk down to you, but we do remember how it feels to be awash in a sea of *bytes* and *bits* and other technical jargon. If you are new to programming, **IRIDIS** is one of the easiest ways you can learn how to get the most out of your ATARI. If you are an old hand, you'll be delighted by the technical excellence of our programs. (We are the people who have published **CURSOR** for the Commodore PET since July, 1978.)

ATARI is a trademark of ATARI, Inc.

Published By

The Code Works™

Box 550
Goleta, CA 93017
805-967-0905
Dealer Inquiries Invited

ORDER FORM

IRIDIS #2 - Fontedit and Knotwork
☐ **$15.95 Cassette** ☐ **$18.95 Disk**

IRIDIS #1 - Clock, Zap, Logo, Polygons
☐ **$9.95 Cassette** ☐ **$12.95 Disk**

Name ______________________________

City ____________ State ________ Zip ________

☐ Visa ☐ MC Card No. ______________________

Expires ____________ Signature ____________

____

IRIDIS requires 16k for cassette, 24k for disk.

www.commodore.ca

710-750 move text and add the next letter, and back to line 560

```
10 DATA 162,1
15 REM ---------- LDX #1
20 DATA 189,213,0
25 REM ---------- LDA SCRN,X
30 DATA 202
35 REM ---------- DEX
40 DATA 157,213,0
45 REM ---------- STA SCRN,X
50 DATA 232
55 REM ---------- INX
60 DATA 232
65 REM ---------- INX
70 DATA 224,18
75 REM ---------- CPX #18
80 DATA 208,243
85 REM ---------- BNE -13
90 DATA 104
95 REM ---------- PLA
100 DATA 96
105 REM ---------- RTS
110 FOR I=5120 TO 5136
120 READ A
130 POKE I,A
140 NEXT I
150 M=PEEK(106): REM HIGH MEMORY
160 POKE 5124,M-2
170 POKE 5128,M-2
180 DATA 4,239
185 REM ------# OF TEXT STRINGS, TOTAL LENGTH
190 DATA THIS IS TEST OF THE SCROLLING
    MARQUEE PROGRAM....
200 DATA YOUR MESSAGE CAN BE DISPLAYED
    HERE CONTINUOUSLY....
210 DATA AMAZE YOUR FAMILY.... MAKE YOUR
    NEIGHBORS JEALOUS... SELL YOUR PRODUCTS...
220 DATA THE ATARI COMPUTERS CAN DO MORE
    THAN JUST PLAY STAR RAIDES....
230 READ B,A
240 DIM A$(A),B$(A),C$(3)
250 FOR I=1 TO B
260 READ B$
270 A$(LEN(A$)+1)=B$
280 NEXT I
290 C=INT(RND(1)*15)+1
300 GRAPHICS 18
310 SETCOLOR 0,C,10
320 SETCOLOR 1,C,10
330 SETCOLOR 2,C,10
340 SETCOLOR 3,C,10
350 SETCOLOR 4,C,2
360 C$=CHR$(10): C$(2)=CHR$(138):
    C$(3)=CHR$(170)
370 FOR I=1 TO 6
380 PRINT #6;C$;
390 NEXT I
400 PRINT #6;C$(1,1)
410 FOR I=0 TO 9 STEP 3
420 FOR J=0 TO 2
430 COLOR (192-32*J)*SGN(J)+10
440 PLOT 0,I+J
450 PLOT 19,11-I-J
460 NEXT J
470 NEXT I
480 C$(2)=CHR$(170): C$(3)=CHR$(138)
490 FOR  I=1 TO 6
500 PRINT #6;C$;
510 NEXT I
520 PRINT #6;C$(1,1);
530 I=1
540 J=2
550 K=3
560 FOR L=1 TO A
570 FOR Q=1 TO 2
580 SETCOLOR I,C,2
590 SETCOLOR J,C+2,10
600 SETCOLOR K,C+4,10
610 I=I-1
620 IF I=0 THEN I=3
630 J=J-1
640 IF J=0 THEN J=3
650 K=K-1
660 IF K=0 THEN K=3
670 FOR M=1 TO 5
680 NEXT M
690 POKE 77,0
700 NEXT Q
710 POSITION 18,5
720 X=USR(5120)
730 PRINT #6;A$(L,L)
740 NEXT L
750 GOTO 560
```

©

Error Reporting System For The Atari

Len Lindsay

One of the disappointing aspects of the Atari Computer System is its lack of user-oriented messages. Particularly disturbing is the error message, or should I say error number? It stops and tells you

ERROR 138

What? Where did I put my manual? You then search through your desk, find the manual, flip pages until you hit the error messages and look up number 138. If you have a disk system the following program will do all the work for you, as well as offer you several options for continuing program execution. (Non-disk users will also find several aspects of the program suitable for use without a disk).

Here is what the program does for you each time an error is encountered:

1) It reports to you that an error was encountered, gives you the error number, and the line number where the error was encountered.

2) If you have an error messages diskette in drive 1 it will next print out an error message in plain English, telling you what went wrong and possibly how to correct it. (Without a disk you won't get this message - but all the rest of the program works fine).

3) It offers you the choice to end program execution or to continue in one of these three ways:

a) continue with the line on which the error was encountered.

www.commodore.ca

b) continue with the line immediately following the error line.

c) continue with the LINK line (equivalent to the TRAP function).

That is the system in a nutshell. It is structured to be of general use - it should be modified to your particular needs. To aid in this, I will explain how the program works.

Program Explanation Error Report

Line 0 is the required DIM statements for string variables used in the system.

Line 1 sets the TRAP to 32500 - the start of the reporting system.
NOTE The TRAP command cannot be used in your program. Instead, simply set the variable LINK to the line you normally would have used for TRAP. Example:

```
250 TRAP 5000
should be entered as:
250 LINK = 5000
```

Line 32500 finds the Line number in which the error occurred. It also finds the error number.

Line 32510 prints the error number and the Line at which it occurred.

Lines 32520-32530 assigns a file name to be used to recover the appropriate error message from disk.

Line 32540 sets a TRAP to report a default message if an error occurs while retrieving the error message (for instance, if your disk is turned off, or if you have no disk).

Line 32550 opens the appropriate disk file and, if successful, skips over the default message.

Line 32570 gets the error message from disk.

Line 32580 jumps to the subroutine to find what the next line after the error line is. It also resets the TRAP for future operation.

Lines 32581-32587 print your options.

Line 32588 asks for your choice.

Line 32589 clears the screen.

Line 32590 turns off the TRAP and ENDS if you hit "S" (for STOP).

Lines 32591-32593 check for other legal choices and goes to the appropriate line.

Line 32599 jumps back to print your options once again if an illegal entry is detected.

Line 32600 starts the routine to find the next line number after the error line. The variable NXLINE is initialized.

Line 32610 finds the first line number in the program.

Lines 32620-32660 finds the line number by starting at the first line and checking one line at a time until it hits the error line. The next line is then used for the next line number.

Announcing
ATARI software™
from the authors of An Invitation to Programming
exciting games and educational programs for kids, teenagers and adults featuring sound and color graphics.
available on guaranteed-to-load cassettes at fine computer dealers in your area or, write us directly for descriptive materials
PDI
Program Design, Inc.
Department CA
11 Idar Court
Greenwich, CT 06830
203-661-8799

www.commodore.ca

Line 32699 Returns back to the line calling this routine. **That's it!**

```
0 DIM ERNUM$(5),ERFILE$(12),XA$(100)
1 TRAP 32500:REM TO ERROR REPORT ROUTINE
2 REM *** ERROR REPORT SYSTEM by
3 REM *** LEN LINDSAY (C) 1980
4 REM YOUR PROGRAM GOES HERE
5 REM SET VARIABLE  LINK  TO THE
6 REM BEGINNING LINE OF YOUR MODULES
7 REM - NEEDS A DISKETTE IN DRIVE 1
8 REM WITH ERROR FILES CREATED WITH
9 REM THE ERROR FILE WRITING PROGRAM
10 REM THANK YOU TO COMPUTE, IRIDIS, AND
 ATARI FOR INFO USED IN THIS
32500 ERLINE=256*PEEK(187)+PEEK(186):ERN
UM$=STR$(PEEK(195)):REM ERROR REPORT SYS
TEM
32501 REM *** NEEDS: DIM ERNUM$(5)
32502 REM ***        DIM ERFILE$(12)
32503 REM ***        DIM XA$(100)
32504 REM *** USES SUBROUTINE 32600 TO F
IND NEXT LINE
32510 PRINT "}ERROR NUMBER ";ERNUM$;" IN
 LINE ";ERLINE
32520 ERFILE$="D:ERROR"
32530 ERFILE$(LEN(ERFILE$)+1)=ERNUM$
32540 TRAP 32560
32550 OPEN #5,4,0,ERFILE$:GOTO 32570
32560 PRINT "ERROR NUMBER ";ERNUM$;" IS
NOT ON FILE":GOTO 32580
32570 INPUT #5;XA$:PRINT XA$:CLOSE #5
32580 GOSUB 32600:TRAP 32500
32581 PRINT " SHALL I :"
32582 PRINT " STOP"
32583 PRINT "     OR "
32584 PRINT " CONTINUE WITH :"
32585 PRINT " ERROR LINE ";ERLINE
32586 PRINT " NEXT LINE ";NXLINE
32587 PRINT " LINK LINE ";LINK
32588 PRINT "  WHICH CHOICE";:INPUT XA$
32589 PRINT "}":REM CLEAR SCREEN
32590 IF XA$="S" THEN TRAP 34567:STOP
32591 IF XA$="E" THEN GOTO ERLINE
32592 IF XA$="N" THEN GOTO NXLINE
32593 IF XA$="L" THEN GOTO LINK
32599 GOTO 32581:REM INVALID RESPONSE
32600 NXLINE=0:REM FIND NEXT LINE NUMBER
32601 REM *** ERLINE   IS INPUT TO THIS
ROUTINE AS THE LINE NUMBER
32602 REM *** NXLINE  IS RETURNED AS THE
 NEXT LINE NUMBER
32605 REM *** BASED ON COMPUTE #4 PAGE 3
2 PROGRAM LISTING
32610 ADDRESS=PEEK(136)+PEEK(137)*256:RE
M GET THE FIRST LINE NUMBER
32620 LINE=PEEK(ADDRESS)+PEEK(ADDRESS+1)
*256
32630 IF NXLINE=1 THEN NXLINE=LINE:GOTO
32699
32640 IF LINE=ERLINE THEN NXLINE=1
32650 ADDRESS=ADDRESS+PEEK(ADDRESS+2)
32660 GOTO 32620
32699 RETURN
```

In order to fully use the ERROR Report System you must have a diskette with all the error messages correctly recorded on it. The following program can be used to create your own custom-made error messages master diskette. It simply asks you for an error number and its matching message. The message is then written to disk under the appropriate error number file.

```
0 REM *** ERROR REPORT WRITER
1 REM *** (C) 1980
2 REM *** LEN LINDSAY
3 REM *** PUTS ERROR INFO TO DISK
10 DIM ERNUM$(5),ERFILE$(12),XA$(100)
90 PRINT "}":REM CLEAR SCREEN
100 PRINT "WRITE ERROR MEANINGS TO DISK"

110 PRINT " GET OUT YOUR ERROR LIST - LE
TS GO-"
120 TRAP 120:PRINT "  WHAT IS THE NEXT E
RROR NUMBER ";:INPUT ERNUM$
125 E=VAL(ERNUM$):TRAP 34567
130 ERFILE$="D:ERROR"
140 ERFILE$(LEN(ERFILE$)+1)=ERNUM$
150 PRINT "  PLEASE TYPE IN ITS MEANING
& HINTS":INPUT XA$
160 OPEN #1,8,0,ERFILE$
170 PRINT "}NOW WRITING ERROR NUMBER ";E
RNUM$
180 PRINT #1;XA$:CLOSE #1
190 GOTO 120
```

Possible System Uses or Modifications

The error reporting system can be used while developing your programs, providing you with messages during your text run - as well as several restart options. The system is presently under manual control after an error is encountered. This of course can be automated to provide error trapping *AND* error correction.

For example, your program may provide a hardcopy printout of the program results. If an error #138 is encountered, you may wish to print a message on the screen such as ''Please turn on the printer'' and then go back to the offending line. Print a cursor-up after the message and you can loop until the printer is turned on, upon which action the program immediately continues executing.

You may also be able to use pieces of this system in your own programs. For example, lines 32520-32530 show how your program can dynamically create its own disk file name based on the value of variables. ©

www.commodore.ca

An Atari BASIC Tutorial

Monthly Bar Graph Program

Jerry White

Atari sounds and graphics are great for game programs. In this monthly graph program, you will see how they also can be used to display data.

Data is often processed and compared on a monthly basis. Reports are generated to monitor things like cask flow or production. Sometimes it is much more meaningful to see totals in bar graph from rather than trying to compare a list of numbers. Using this program, the user types in the monthly totals and the program converts these figures into a beautiful graphic display.

For those who like to know how programs work: I'll break this one down and explain what each section is doing. For those who don't care: just key in the program and input your totals next to the appropriate month. The program will do the rest.

We begin by dimensioning A$ for use as a work string and two numeric arrays to hold 12 items. We go to the subroutine at 2000 and get our monthly totals and return to line 4. Here we get into graphics mode 6 with the text window at the bottom. We position our graphics window X and Y coordinates using PX and PY and put our heading into the A$ string. Now we're off to the subroutine at 20. We will use this routine to convert our scratch string so that we can put text in the graphics window. Returning to line 8 - we use color 1 and draw a large rectangle. This is where we will draw our data bars. At line 100 we determine the highest amoung (HAMT) so that we can base our key on that figure. The key will give meaning to the lengths of the bars. We set J1 = HAMT divided by 65 which is the length of the longest bar that fits into our rectangle. At line 130 we determine the top position of each bar. Then we make the top key figure (K) into a one or two position numb er and compute the numbers that will appear along the left side of the graph. At line 240 we begin to position and place our key of the screen. Then we set the screen margins as wide as possible and put the abbreviations for each month in the text window directly below the bar it represents. At line 310 we begin to draw our bars.

Being quite fond of sound, I couldn't resist adding line 360 as a finishing touch. This loop creates a tone as each bar is completed. The higher the pitch, the higher the tone. Our purpose, was to display data. Why not let the user use his ears as well as his eyes? Before we exit - we set the screen margins back to normal and loop at line 500. You could replace 500 with an end or exit routine. If you remove the first? : from line 300 there will be one line left in the text window for a message.

```
0 REM MONGRAPH REV.2 JERRY WHITE
1 REM FOR COMPUTE TUTORIAL
2 DIM A$(20),AMT(12),JW(12):GOSUB 2000
4 GRAPHICS 6:SETCOLOR 2,4,4:SETCOLOR 4,4,4:Z=1:SETCOLOR 0,1,10
6 PX=4:PY=0:A$="MONTHLY GRAPH":GOSUB 20
8 COLOR Z:PLOT 18,9:DRAWTO 158,9:DRAWTO 158,75:DRAWTO 18,75:DRAWTO 18,9
10 GOTO 100
20 DL=PEEK(560)+PEEK(561)*256:D1=PEEK(DL+4)+PEEK(DL+5)*256
22 FOR U=Z TO LEN(A$):D2=57344+((ASC(A$(U,U))-32)*8):
   D3=D1+PY*20+PX+U-Z:FOR JZ=0 TO 7
24 POKE D3+JZ*20,PEEK(D2+JZ):NEXT JZ:NEXT U:RETURN
100 FOR MON=Z TO 12:IF AMT(MON)>HAMT THEN HAMT=AMT(MON)
110 NEXT MON
120 J1=HAMT/65
130 FOR MON=Z TO 12:TAMT=75-(AMT(MON)/J1):JW(MON)=INT(TAMT):NEXT MON
140 IF HAMT>=10000 THEN K=INT(HAMT/1000):GOTO 200
150 IF HAMT>=1000 THEN K=INT(HAMT/100):GOTO 200
160 IF HAMT>=100 THEN K=INT(HAMT/10):GOTO 200
170 K=INT(HAMT)
200 KD=K/5:K2=INT(K-KD):K3=INT(K-(KD*2))
220 K4=INT(K-(KD*3)):K5=INT(K-(KD*4))
222 A$=STR$(K):PX=2-LEN(A$):PY=10:GOSUB 20
224 IF K<5 OR K>99 THEN 280
240 A$=STR$(K2):PX=2-LEN(A$):PY=24:GOSUB 20
250 A$=STR$(K3):PX=2-LEN(A$):PY=38:GOSUB 20
260 A$=STR$(K4):PX=2-LEN(A$):PY=52:GOSUB 20
270 A$=STR$(K5):PX=2-LEN(A$):PY=66:GOSUB 20
280 POKE 82,0:POKE 83,40:POKE 752,Z
300 ? :? "  K  J  F  M  A  M  J  J  A  S  O  N  D"
302 ? "  E  A  E  A  P  A  U  U  U  E  C  O  E"
304 ? "  Y  N  B  R  R  Y  N  L  G  P  T  V  C"
310 FOR MON=Z TO 12:JY=MON-Z
312 PLOT 18+(JY*12),JW(MON)
314 DRAWTO 25+(JY*12),JW(MON)
320 DRAWTO 25+(JY*12),75
330 DRAWTO 18+(JY*12),75
340 POSITION 18+(JY*12),JW(MON)
350 POKE 765,3:XIO 18,#6,0,0,"S:"
360 FOR VOL=10 TO 0 STEP -1:SOUND 0,JW(MON),10,VOL:NEXT VOL:NEXT MON
400 POKE 82,2:POKE 83,39
500 GOTO 500
2000 GRAPHICS 0:SETCOLOR 2,0,0:SETCOLOR 1,0,10:SETCOLOR 4,0,0:POKE 752,1
2080 ? :? ," MONTHLY GRAPH "
2100 ? :? "   TYPE AMOUNTS FOR EACH MONTH: ":?
2120 ? "   DO NOT USE NEGATIVE AMOUNTS  ":?
2200 TRAP 2200:? ,"JAN=";:INPUT JAN:AMT(1)=JAN:TRAP 40000
2210 TRAP 2210:? ,"FEB=";:INPUT FEB:AMT(2)=FEB:TRAP 40000
2220 TRAP 2220:? ,"MAR=";:INPUT MAR:AMT(3)=MAR:TRAP 40000
2230 TRAP 2230:? ,"APR=";:INPUT APR:AMT(4)=APR:TRAP 40000
2240 TRAP 2240:? ,"MAY=";:INPUT MAY:AMT(5)=MAY:TRAP 40000
2250 TRAP 2250:? ,"JUN=";:INPUT JUN:AMT(6)=JUN:TRAP 40000
2260 TRAP 2260:? ,"JUL=";:INPUT JUL:AMT(7)=JUL:TRAP 40000
2270 TRAP 2270:? ,"AUG=";:INPUT AUG:AMT(8)=AUG:TRAP 40000
2280 TRAP 2280:? ,"SEP=";:INPUT SEP:AMT(9)=SEP:TRAP 40000
2290 TRAP 2290:? ,"OCT=";:INPUT OCT:AMT(10)=OCT:TRAP 40000
2300 TRAP 2300:? ,"NOV=";:INPUT NOV:AMT(11)=NOV:TRAP 40000
2310 TRAP 2310:? ,"DEC=";:INPUT DEC:AMT(12)=DEC:TRAP 40000
2400 RETURN
```

©
www.commodore.ca

Card Games in Graphics Modes 1 and 2

William D. Seivert

Have you ever wanted to design a card game to play in Graphics Mode 1 or 2, only to find that you couldn't get the suit characters (heart, spade, diamond and club) to appear on the screen at the same time as the characters A, K, Q, J and the digits 0 through 9?

Graphics modes 1 and 2 use the character base pointer (CHBAS, location 756) to point to the table defining the character sets. When location 756 contains 224, you get uppercase letters and the digits and normal punctuation. When you set it to 226, you get small letters and the graphics characters, including the characters for the suits. Since only 64 characters are available in these modes, you can't have both at the same time!

Try this in Direct Mode:

```
GRAPHICS 2: PUT #6,ASC("]"):POKE 756,226
```

When the POKE takes effect, the right bracket changes to its graphics equivalent. (So does the rest of the graphics window!) The table to look at is in the BASIC Reference Manual, Table 9.6.

The 224 or 226 that you POKE into location 756 is the Most Significant Byte (MSB) of the start address of the character set table. Since these tables are in ROM, they cannot be changed directly. Also, since only the MSB of the address is used, the table must begin on a page boundary.

It takes a lot of work and space in BASIC to hold the table and ensure that it is on a page boundary. However, there is an easier way!

The following BASIC subroutine will do the job.

```
25000 REM REDEFINE CHARACTER SET AND
REPLACE [/] > WITH
25001 REM DESTROYS TRAP, USES STRING ST$
      AND VARIABLES I AND J
25010 J = (PEEK(106)-8)*256
25020 IF J< = PEEK(144) + 256*PEEK(145) THEN?
      "PROGRAM TOO LARGE TO REDEFINE CHARS":
      GO TO 25120
25030 FOR I = 0 TO 1023
25040 POKE J + I,PEEK(57344 + I):NEXT I
25050 J = J + 472
25060 TRAP 25070:DIM ST$(32):TRAP 40000
25070 ST$ = " (* - See below for keyboard keying
      sequence) "
25080 FOR I = 1 TO 32
25090 POKE J + I-1,ASC(ST$(I,I))
25100 NEXT I
25110 POKE 756,PEEK(106)-8
25120 RETURN
```

Now I'll explain what this does by line number.

25000,25001 Just some documentation (Remember that GOSUB 25000 will work, BASIC will skip the REMs).

25010 Location 106 contains RAMTOP, the number of pages of RAM. Subtracting 8 leaves enough room for graphics modes 0, 1 and 2, and allows space for the new character set table. Thus, J is the address where the table will start.

25020 Locations 144 and 145 contain MEMTOP which is BASIC's current top of memory. If, at the time the subroutine is called, the program is already too big to allow for the new table, we won't do it and leave. This implies that all arrays should be DIMensioned and variables defined before calling the subroutine.

25030,25040 This loop moves the original table (57344 = 224*256) from ROM to the new location in RAM.

25050 Each character uses 8 bytes (1 byte per TV scan line) to define which pixels should be on for the given character. Adding 472 (= 59*8) to the starting address gives the address of the left bracket ([) character.

25060 The TRAP is used so that if the subroutine is called more than once in a run, we won't get ERROR 9 (String DIM Error). We need 32 bytes for string ST$ (4 characters times 8 bytes per character). Then we cancel the TRAP so other errors don't come to this routine.

25070 Now we define the bytes for the four suit characters. The keying sequence after ST$ = " is: CTRL comma, 6, ESC TAB, ESC TAB, greater-than, ESC CTRL minus, CTRL H, CTRL comma, CTRL comma, CTRL X, less-than, ESC BACK-S, ESC BACK-S, less-than, CTRL X, CTRL comma, CTRL comma, ESC CTRL minus, ESC CTRL minus, lowercase W, lowercase W, CTRL H, ESC CTRL minus, CTRL comma, CTRL comma, CTRL X, less-than, ESC BACK-S, ESC BACK-S, CTRL X, less-than, CTRL comma, and the closing double quotes.

25080 Start the loop to put the bytes.

25090 Convert one character at a time to its ATASCII value and POKE it in the appropriate location.

25100 Finish the loop.

25110 Put the address of the new table in CHBAS (location 756).

25120 Return to the caller.

That's all there is to it! Of course this method will work for any characters you want to redefine. All you have to do is decide which characters you can do without, and the bit patterns of the characters you want.

With the above routine as it is, if you want a

www.commodore.ca

heart, use the left bracket, etc. Use PUTs to the screen for the characters you want. Remember that you can use inverse-video and/or add values to change colors.

For example, without using any SETCOLOR statements, try

```
GRAPHICS 2: GOSUB 25000
PUT #6,ASC("inverse-video of right bracket")
```

to get a blue Club, or

```
PUT #6,ASC("inverse-video of left bracket") + 32
```

to get a red Heart.

A few words of warning

Every time you change graphics modes (even GRAPHICS n + 32 which doesn't change the screen), the Operating System resets location 756 to 224, pointing to the normal character set. If you want the suit characters back again, just GOSUB 25110.

Also, if you use a graphics mode greater than 2, you might destroy the table. So you will probably want to GOSUB 25000 after coming out of graphics mode 3 or above.

Of course you do not have to use the same line numbers, and you might want to remove the memory overlap check at line 25020, but that's up to you.

Try it! You'll like it! ©

Keep Those Cards and Letters Coming

COMPUTE! Needs You!

Address articles, programming notes and comments to:
The Editor
COMPUTE!
P.O. Box 5406
Greensboro, NC 27403

SOFTWARE FOR THE ATARI®400/800

Quality Software™ offers important software to owners of ATARI 400 and 800 computers. All programs are on cassette.

ASSEMBLER by Gary J. Shannon. Create your own 6502 machine language programs with this easy-to-use in-RAM editor/assembler. Requires 16K or more of RAM to operate. Look at all the features you get for less than $25!

- Insert, delete, edit source code lines
- Save source code on cassette
- Save object code (any part of memory) on tape
- Print out assemblies
- View and modify memory
- Pseudo Ops: ORG,OBJ,EQU,HEX,ASC,DA,DS,END
- Printer and video control (PON,POF,VON,VOF)
- All 6502 mnemonics plus BLT,BGE
- Commenting allowed
- Error checking
- Documentation includes notes on interfacing machine language to BASIC
- Price —**$24.95**

6502 DISASSEMBLER by Bob Pierce. This neat 8K BASIC program allows you to disassemble machine code and print out the disassembled listing. If you have more than 8K of memory, programs in RAM can be disassembled. Operating System ROM and the BASIC ROM can be disassembled on any size ATARI. Also works as an ASCII interpreter, translating machine code into ASCII characters. **$11.95**

FASTGAMMON™
by Bob Christiansen. The most popular backgammon-playing program for personal computers is now available for the Atari. This is the best-playing version so far, and includes the option to enter your own dice rolls. Set the display speed to your liking—play fast or slow. Beginners find it easy to learn backgammon by playing against the computer, and even very good players will find it a challenge to beat FASTGAMMON. Includes 12 pages of instructions that include the rules of the game. Written in machine language. Requires only 8K of RAM. **$19.95**

QUALITY SOFTWARE
6660 Reseda Blvd., Suite 105, Reseda, CA 91335
Telephone 24 hrs., 7 days a week: (213) 344-6599

WHERE TO GET IT: Ask your nearest Atari dealer to see *Quality Software's* Atari programs. Or, if you prefer, you may order directly from us. MasterCharge and Visa cardholders may telephone their orders and we will deduct $1 from orders over $19 to compensate for phone charges. Or mail your order to the address above. 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, payable in U.S. currency.

ATARI, ATARI400, and ATARI800 have been trademarked by Atari Personal Computer Systems, a Warner Communications Company.

www.commodore.ca

Using TAB in ATARI BASIC

James L. Bruun

As most of us have discovered, ATARI BASIC has no TAB function. That is, you can't do a TAB(15), as you are used to doing in some other BASICs. However, BASIC was initially designed for use on large systems which might have several different terminals with different TAB keys. The solution was to create the familiar TAB function.

The ATARI machine is a new breed of cat. The designers weren't faced with an unknown remote terminal, so a TAB could be created which worked like the TAB on a typewriter - only better.

Because of the interaction of the CONTROL, SHIFT, and normal keys on the ATARI keyboard, all TAB functions can be placed on the same key. TAB tabs, CTRL-CLR clears the tab at the cursor position, SHIFT-SET sets the tab at the cursor position. This combination of keys works just like the TAB functions on a typewriter. To clear a TAB, move the cursor to the TAB position, (just press TAB) and press the CTRL and CLR keys at the same time. To set a tab, move the cursor to the desired position and press SHIFT and SET. To use the tabs, just press the TAB key, and the cursor will move to the next tab setting.

Using the TAB functions within a program is similar. All you need to do is place the characters for these functions into a text string and print them. They must be inside a text string to function. To get them inside a text string just press ESC before the desired function and the character for that function will appear on the screen.

As an example the following program was written and tested on an ATARI 800 by the author.

```
10 DIM CLEARTAB$(11)
20 CLEARTAB$ = "ESC CTRL-CLR ESC TAB ESC
CTRL-CLR ESC TAB
ESC CTRL-CLR ESC TAB ESC CTRL-CLR ESC TAB
ESC CTRL-CLR ESC TAB ESC CTRL-CLR"
```

This sequence creates a string variable called CLEARTAB$ which contains eleven characters. When printed this string will clear the current TAB and the next five. This should clear all tabs that are set in most programs. If fewer tabs are present, no harm is done. The remainder of the program illustrates their use.

```
30 PRINT "ESC CTRL-CLEAR" : REM CLEAR SCREEN
40 PRINT CLEARTAB$ : REM CLEAR TABS
50 PRINT "JANUARY ESC SHIFT-SET31"
60 PRINT "FEBRUARYESC TAB"
70 PRINT "MARCHESCTAB28"
80 PRINT "APRILESC TAB30"
90 STOP
```

Line 40 prints "JANUARY 31" and sets the tab at the 3 in 31. Line 50 prints "FEBRUARY 28" and so on. Notice that the tab character is embeded in the text. Here is another example: (Add these lines to the previous ones.)

```
100 PRINT "ESC CTRL-CLEAR"
110 PRINT CLEARTAB$
120 PRINT " ESC SHIFT-SETNUMBER ESC SHIFT-
SETSQUARE"
130 FOR COUNT = 1 to 10
140 PRINT "ESC TAB";COUNT;"ESC TAB";COUNT*
COUNT
150 NEXT COUNT
160 END
```

As we left this program we could have cleared the tabs. Since we didn't they are still set. Try them. This should give you a sample of the way that TAB works on the ATARI. Different, and I think better. Run the program, experiment, and write to tell of the marvels that you have discovered in your ATARI. ©

NEW For The ATARI™

PRESCHOOL FUN
(16K BASIC) This readiness program has two parts with several individual modules. Part one reinforces color, shape and number recognition. Part two has units on upper and lower case letters and directions. No reading required. Full color graphics and sound. .... cass. $15.00

MATH FACTS - LEVEL 1
(16K BASIC) First in a series of self-paced instructional programs for elementary school children. The program automatically advances to the next unit when the child has mastered 80% of the work generated by the computer. The previous unit will be reviewed if the child cannot master 50% of the work in a particular unit. Concepts covered are: numbers, number placement and number words (1-20), addition and subtraction (visual and abstract). (Grades K-2) .... cass. $15.00

CRIBBAGE
(24K BASIC) Play cribbage with the computer at two different levels. As a beginner, the computer will point out your errs without penalizing you. But watch out! At the intermediate level, the computer will peg your points if you don't ...... cass. $15.00

CASINO I
(16K BASIC) Try your luck at the Lucky Lady . . . Play BLACKJACK . . . The computer will be the dealer for you and your friends (1-4 players). You can split and double your hand as you attempt to break the house. OR hit it big on the SLOT MACHINE. Two programs on one tape..................... cass. $15.00

*ATARI is a trademark of Atari, Inc.

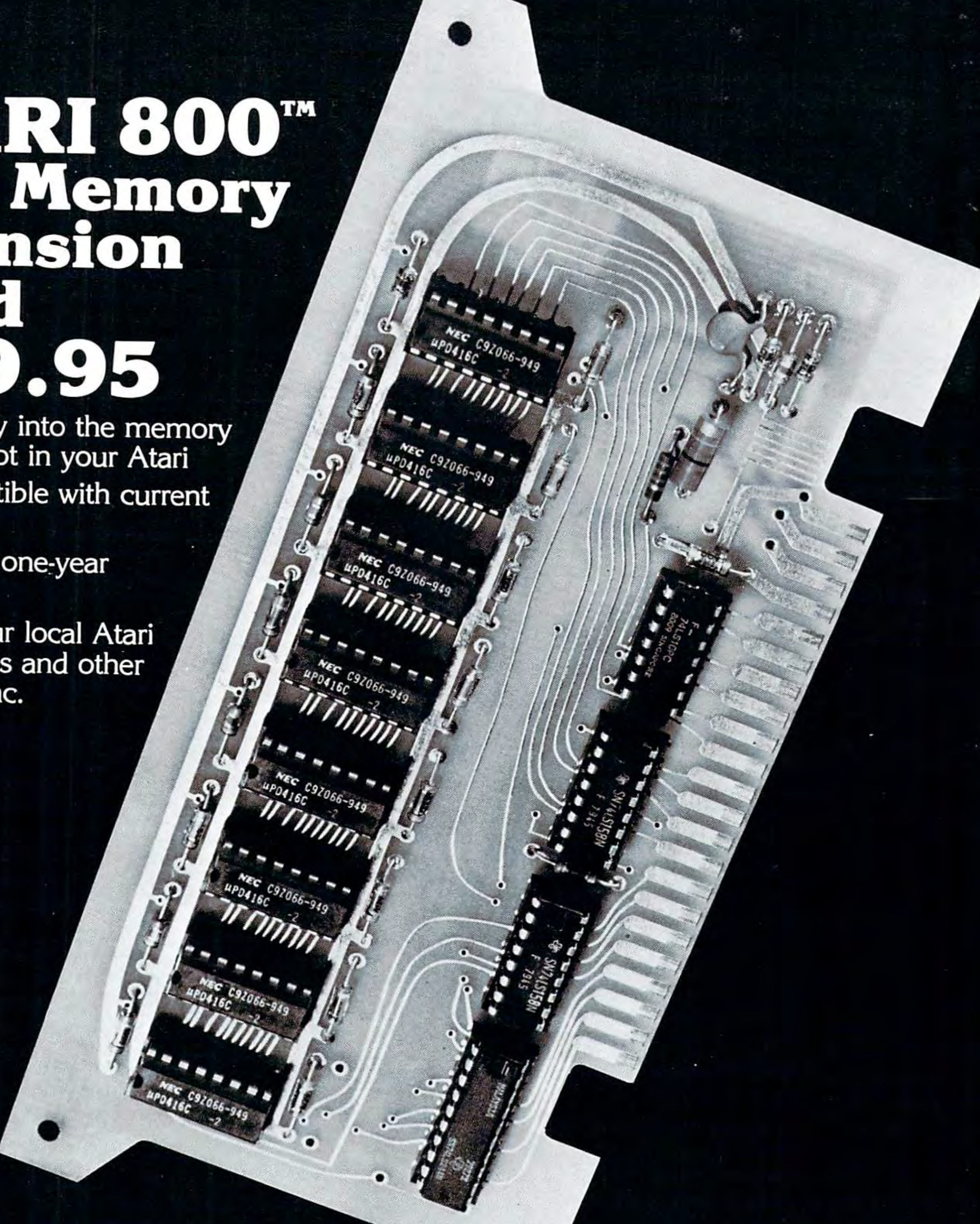
T.H.E.S.I.S.
P.O. Box 147
Garden City, MI 48135
or call (313)595-4722 for C.O.D.

Please add $1.50 for shipping
Mich. residents add 4% tax
WRITE for FREE FLYER
DEALER INQUIRIES WELCOME

www.commodore.ca

ATARI 800™ 16-K Memory Expansion Board $179.95

- Plugs directly into the memory expansion slot in your Atari
- Fully compatible with current hardware
- Backed by a one-year warranty
- Inquire at your local Atari Dealer for this and other fine JACC, Inc. products



JACC, Inc.

(Just Another Computer Company)
543 West Golf Road
Arlington Heights, IL 60005
Dealer/Distributor Inquiries Invited
Dealer Line: 312-364-6268
On the west coast contact Micro Distribution, Inc.
2612 Croddy Way, Santa Anna, CA 92704 714-641-0205

Atari is a trademark of Atari, Inc.

- **Three New Software Products**
- **Copy/Verify Program**
  Single & Double Disk Copy
- **Ron's Disk Utility**
  Intelligent Absolute Disk Editor
- **Inside Atari BASIC**
  Disassembler and complete documented listing of Source Code for BASIC

Check your dealer or write for prices and full description

www.commodore.ca

Pokin' Around

Charles Brannon

Perhaps one of the most useful commands in BASIC is POKE. Why? Because POKE allows you to do some things that cannot be done as easily in BASIC. I recall the earlier days of the PET, where every time a nifty memory location was discovered, it was published with glee -- indeed, they were real "tid-bits". Nowadays, however, there are several very good memory maps that document the inner workings of the PET quite well.

In the Atari Basic Reference Manual, there is an appendix entitled "Memory Locations" (Appendix I). Although it is not a true memory map, since it is incomplete, it does list some very interesting locations.

During the execution of a program, the cursor does not disappear. Rather, it moves with the print statements and sometimes is left behind, cluttering up the screen with little white squares. Fortunately, the visibility of the cursor can be zeroed out with a simple statement: POKE 752,1. To restore the cursor, press the BREAK key or POKE 752,0. The well-known problem of the non-standard behavior of the Atari's GET statement has led to the discovery of memory location 764. Here is stored the code representing the last key pressed. This is not in ATASCII, but is a code used in the scanning of the keyboard. If no key has been pressed, a value of 255 will be found here. I first found this technique right here in COMPUTE. In "Adding a Voice Track to Atari Programs", the author instructed to use a subroutine like this to check if a key has been pressed:

```
lineno IF PEEK (764) = 255 THEN lineno (same lineno)
lineno POKE 764,255: RETURN
```

The first statement waits for a key to be pressed; the second discards that keypress by making BASIC think no key was pressed so that the keystroke would not be printed accidentally.

An example of how POKEing can be easier to use than a BASIC equivalent is to directly control the five color registers. After all, they too are only mere memory locations. Locations 708-712 correspond to SETCOLOR color registers 0-4. Using the notation *SETCOLOR aexp,aexp,aexp* where *aexp* is an arithmetical expression that will evaluate into a number, the first number is from 0-4, so use the appropriate memory location. Then multiply *aexp* number two by 16 and add the third number to it. This gives you an integer in the range 0-255. Now just enter POKE COLR, NUMBER where COLR is the memory location of the color register and NUMBER is that number you obtained. Figuring out what color is already being displayed is done in the reverse fashion. Get the contents of the color register with PEEK(COLR), and assign it to some variable, say X. (e.g. X = PEEK(COLR)) Divide X by sixteen, throw away the fraction using Y = INT(X), then find the luminance (aexp #3) with L = X-16*Y. Now you can set the color by basic with SETCOLOR COLR-708,Y,L or you can just store the numbers so you can meditate on them at a later date.

Here are some more memory locations you should look into:

656:	**Cursor row**
657 & 658:	**Cursor column**
53279:	**Monitors the Console keys (OPTION, SELECT, START) Different bits of the binary equivalent number indicate which one is being pressed. Bit 2 = Option; bit 1 = Select; bit 0 = Start** **The appendix says to POKE 53279,0 before reading.**
53775:	**If bit 2 = 0 then the last key is still being pressed.**

EXPERIMENT!

Have fun with these memory locations you hackers! You beginners -- step right up and add several new functions to your repertoire!

I want to leave you one more thing to try --POKE 755,6. It's weird! (You can get it back to normal with POKE 755,2 or by pressing RESET. ©

WANTED Atari Owners who need support

Send for your FREE Mosaic Electronics software catalog today. Includes games, educational programs, utilities and more.

I have an Atari 400______ Atari 800 ______

Please send me your:

☐ 8K to 16K RAM expansion kit ($79.00)
(My check is enclosed)

☐ Software Catalog

NAME ______

STREET ______

CITY______ STATE______ ZIP______

Send to: MOSAIC ELECTRONICS
P.O. Box 748 (CC) Oregon City, Oregon 97045
Mosaic Electronics is in no way affiliated with Atari.

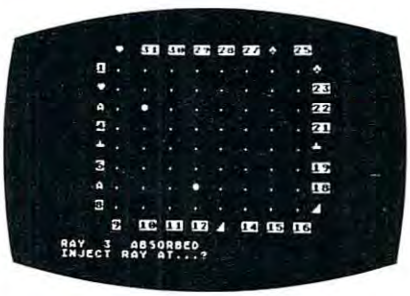
www.commodore.ca

VersaWare for your ATARI®
MIND BOGGLERS-I

THREE THOUGHT PROVOKING MIND BOGGLERS FOR YOUR ATARI® 400 OR ATARI® 800

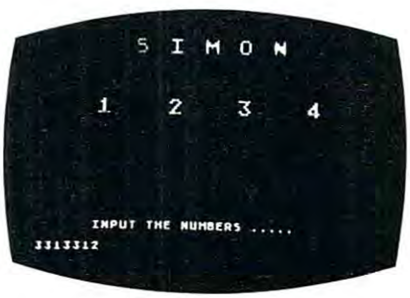
CAPTURE

A strategy game in which you and your computer fight for control of the board. Based on the popular game Othello™. A challenge at any level of play. Features computer vs. player, player vs. player or computer vs. computer.



MYSTERY BOX

Shoot rays into the mystery box to find the hidden atoms. Increase the number of atoms to increase the difficulty but don't hide too many or your mind may be permanently boggled.



SIMON SAYS

A memory teaser in which you must repeat the computer's pattern. The game adapts to the players skill level, as you get better the game gets harder.

ALL THREE GAMES FOR ONLY
$15.95 CASSETTE OR $19.95 DISK.*
*Requires 16K cassette or 24K disk.

Check your local dealer or order direct.

Dealer inquiries invited.

MIND BOGGLERS-I

☐ Cassette	$15.95
☐ Disk	$19.95

NAME

ADDRESS

CITY STATE ZIP CODE

California residents add 6% sales tax

Versa Computing, Inc. 887 Conestoga Circle • Newbury Park, CA 91320 (805) 498-1956

ATARI® is a registered trademark of Atari Inc.
OTHELLO™ is a trademark of CBS, Inc.

www.commodore.ca

Winning Star Raiders

William L. Colsher

Star Raiders...as a Ferrari is to a Ford so is Star Raiders to any other computer game yet introduced. Burned into 8K of imperishable ROM (there are rumors of an unreleased 16K version), using all of the Atari computer's audio/visual capabilities, Star Raiders has revolutionized our ideas of what a personal computer can be made to do.

But Star Raiders is not a simple game. Taking all the best from Atari's arcade and TV game experience and adding the complexities that a full-blown computer makes possible Star Raiders is endlessly challenging. The twelve lavishly illustrated pages of the instruction manual are adequate for the new user but it soon becomes apparent that advice from an experienced player would be of incalculable aid.

Of course, the most important source of information is the manual supplied with the game. You must know how to shoot and manuver, what the best speed of your ship is, etc. For the remainder of this article I will assume that you have at least those minimal skills.

In the section titled "VII. RATING" is a formula which is used to establish you rating at the end of a game. (It is given in Figure 1 here.) You can see that the two most important factors in your score are 1) finishing the game (i.e. getting all the Zylons) and 2) doing it with a minimum of energy expenditure. Time is surprisingly unimportant. Since it is divided by 100 it will rarely cost you more than 5 to 7 points at the Pilot level. (I'll use Pilot as an average level.)

There are a number of techniques for reducing energy usage. The most important one for the beginner is: DON'T FIRE 'TIL YOU SEE THE SUCKERS ON THEIR TENTACLES! When I first played Star Raiders I often wasted hundreds of energy units on a single Zylon. It costs 10 units every time you fire; make every shot count.

Chasing the lousy zwilnicks* around a sector is another good way to waste energy. Quite often the Zylons will come to you! You can save literally thousands of energy units in the course of a game by lying craftily in wait. A corollary of this rule is "Never Fight on the Run". As soon as your attack computer indicates a range of about 120 stop and wait - they'll come to you.
(Remember that your best speed is 6 on the controls.)

With a little practice you will find that it is possible to avoid most of the shots fired by the Zylon dogs. This is made easier if you are dead in space, letting the Zylons do the running around. Simply by keeping your eyes open and steering around the shots you'll find that you'll only take 3 or 4 hits during the course of a Pilot level mission. When you realize that each hit on your shields costs 100 units this little bit of strategy begins to make a lot of sense.

In all but the novice level, Zylon fire is capable of doing considerable damage to your ship. Fortunately, unless you lose your shields they can't destroy you. But they can make the game hard to play. The only things that you actually *need* to fight the zylons are your shields and your photons. Everything else just makes the game easier. For example, you will rarely need your Sector Scan; if it is destroyed there is no need to return to a starbase to get it fixed. Your engines are necessary only if you've found a need to chase the Zylons. If they've been coming to you (and it seems to vary from game to game) engines just waste energy. If you lose your Sub-Space Radio it should be repaired as soon as possible but don't break off a battle if it goes. Finish up and then go get it fixed. (If you can, make short jumps into Zylon infested sectors on your way to a Starbase; you might as well make good use of the trip.)

If your computer goes you have something of a problem. It is possible to carry on your mission without it but... You won't be able to tell how far away the Zylons in your sector are. It will be difficult to steer in Hyperwarp since the cross-hairs on your display will be gone. If you are engaged with the enemy continue the battle. *Then* try to find your way to a starbase. (It is considered *very* poor form to draw the cross-hairs on your monitor screen.)

Your hyperwarp can be your worst enemy in terms of energy usage. If you refer to section "V. ENERGY USAGE" you will find a table (reproduced in part in Figure 2) which gives the cost of hyperwarp jumps in energy units. You can see from the table that a jump of 5 sectors costs 500 units. A jump of 4 sectors and then 1 sector will cost only 360 units. A rule should be obvious: "NEVER JUMP MORE THAN 4 SECTORS AT A TIME". You should be aware that there are "dead zones" in some sectors that will considerably reduce the cost of a jump. Keep an eye out for them but don't spend a lot of time looking either.

There are a number of what might be called "finesse" methods of saving energy in addition to the obvious ones listed above. For example, it requires about 10 seconds to refuel/repair at a starbase. If you turn off your screens and computer when you reach a base sector you'll save at least 25 units of energy. (Of course, there is the danger of ramming an asteroid, but that chance is quite small.) Generally, turning off your screens when they're not needed is a pretty good idea. At 2 energy units per second you can save quite a lot over the course of a game.

www.commodore.ca

Another trick to save a little energy is to use the P(ause) control when you're looking over the galactic chart. That way you can take your time in planning your course without losing any game time or energy.

Once you've played a few games of Star Raiders a number of facts about the Zylon strategy become evident. First, they all move in the same direction. Second, after a game has been in progress, they tend to bunch up around a starbase. As soon as that starts to happen you've got 'em. Since they are bunched up you'll only use a little energy in getting from one infested sector to the next. That means a better score for you (but be careful you don't lose a starbase in the process!).

Another point in the Zylon strategy that is mentioned in the manual but often overlooked is that the Zylons move only when the clock ticks 00 and 50. If you notice the clock getting close to one of those numbers wait the few seconds before jumping. You may jump to an empty sector (the Zylons having moved while you were in hyperwarp) and waste energy or you may find a jump better than the one you had planned after the enemy has made it's jumps.

Of course, the best way to score on the Zylons is to practice. Don't drop out of a game if things are going poorly. Chances are you'll learn a lot more by fighting your way out of a spot than by quitting. You may get a low rating but remember: it's *only a game.*

(In case you've been wondering, my highest score has been Commander, Class 2.)

Figure 1

RATING = 6(# ENEMY DESTROYED) - ENERGY USED/100 - LENGTH OF TIME/100 - 18(# STARBASES DESTROYED BY ENEMY) - 3 (# STARBASES DESTROYED BY YOU) + M

VALUES FOR M:

MISSION LEVEL	MISSION COMPLETED	MISSION ABORTED	STARSHIP DESTROYED
Novice	M = 80	M = 60	M = -40
Pilot	M = 76	M = 60	M = -50
Warrior	M = 60	M = 50	M = -40
Commander	M = 111	M = 100	M = -90

RATING	MISSION SCORE
Rookie	48-79
Novice	80-111
Ensign	112-143
Pilot	144-175
Ace	176-191
Lieutenant	192-207
Warrior	208-223
Captain	224-239
Commander	240-271
Star Commander	272-303

Figure 2

Hyperwarp Distance	Energy Drain
0	100
1	130
2	160
3	200
4	230
5	500
6	700
7	800
8	900
9	1200
10	1250

●●● ●●● ©

Program Note:

Wynn Smith, of MOSAIC Electronics, sent in this "self-explanatory" program. RCL

```
1 REM MOSAIC ELECTRONICS
2 REM WYNN SMITH

Lines 10-22: Display information for the user
 to read while waiting for the next  di splay.
10 GR.0
12 PRINT "NOTICE THE FOLLOWING"
14 PRINT "1... USER DEFINED CHARACTERS"
16 PRINT "2... UPPER & LOWER CASE CHARACTERS"
18 PRINT "3... 3 GRAPHIC MODES SIMULTANEOUSLY"
Lines 25 - 27 : Subroutine line numbers
25 MOVE = 500
27 CREATE = 600
Lines 30-50: Move uppercase character
 descriptions to RAM
30 CHADR = 57600
40 NNEWADR = 8448
45 COUNT = 255
50 GOSUB MOVE
Lines 60-80: Move lowercase
60 CHADR = 58112
70 NNEWADR = 8192
75 COUNT = 255
80 GOSUB MOVE
Lines 84-90: Move numbers
84 CHADR = 57504
86 NNEWADR = 8384
88 COUNT = 39
90 GOSUB MOVE
Lines 100-120: Creates logo characters
100 A1 = 8664
110 A2 = 8695
120 GOSUB CREATE
Lines 130-150: Creates a space character
130 A1 = 8192
140 A2 = 8199
150 GOSUB CREATE
Lines 160-190: Data for logo characters
160 DATA 63,7,3,49,56,60,60,0
170 DATA 248,192,128,24,56,120,120,0
180 DATA 255,255,255,238,231,227,227,0
190 DATA 0,224,248,124,30,142,142,0
Line 200: Data for space character
200 DATA 0,0,0,0,0,0,0,0
Line 210: Graphics 2 + 16
210 GRAPHICS 18:SETCOLOR 1,4,8:SETCOLOR4,13,0:
    SETCOLOR 0,8,8:SETCOLOR2,13,0
Lines 220-230: Change display list
220 LOC = PEEK(560) + PEEK(561)*256
230 POKE LOC+6,6: POKE LOC+7,3
Line 240: Point character generator to RAM
240 POKE 756,32
Lines 250-300: Print text
250 ?#6;"    OSAIC"
260 ?#6;"%,%#42/.)#3"
270 ?#6;"PO BOX ;8< OREGON CITY OR =;089"
280 ?#6:?#6:?#6
290 ?#6;"[ ]  U3%2 D%&"
300 ?#6;?#6; "UPPER !.$ ,/7%2#!3%"
Lines 500-530: Move ROM into RAM
500 FOR OFFSET = 0 TO COUNT
510 POKE NNEWADR + OFFSET, PEEK (CHADR + OFFSET)
520 NEXT OFFSET
530 RETURN
Lines 600-640: Create new characters
600 FOR ADDR = A1 TO A2
610 READ BITPAD
620 POKE ADDR, BITPAD
630 NEXT ADDR
640 RETURN
```

©

www.commodore.ca

Coded Data For OSI1P

Charles Stewart

I have had my OSI1P for several months now and have a number of word game programs including a version of Hangman which I utilize for my children's spelling words. The OSI has a nice feature of listing on the screen the program as it loads from cassette. This is fine for checking proper program loading and recorder levels etc., but at times a method of hiding information would be useful particularly in word games where you don't want the player to have access to the word list.

In search of a solution to this problem I have written a program which hides the information in DATA statements in ASCII code and writes DATA statements so that a file of words can be generated and inputed to the host program such as Hangman. The program to convert raw data to ASCII follows:

```
3 REMSET G$ TO DATA DIM VAR AS LARGE
AS MEMORY PERMITTS
5 G$ = "DATA":X = 1:DIMA$(50),X(50),Y(50):
7 INPUT"DATA LINE TO START";DA
8 INPUT"INCREMENT BY";IN
10 REM TO END TYPE '*' TO QUESTION
'WORD TO HIDE'
20 INPUT"WORD TO HIDE";A$(X):IFA$(X) =
"*"THEN50
40 X = X + 1:GOTO20
50 PRINT"SAVE CODED WORDS":INPUT
"RECORDER READY";B$:SAVE
85 FORX = 1TO10:PRINT:NEXT:X = 1
95 IFA$(X) = "*"THEN140
97 PRINTDA;G$;:FORW = 1TOLEN(A$(X)):
H$ = MID$(A$(X),W,1)
112 PRINTASC(H$);:NEXT
115 PRINT:DA = DA + IN:X = X + 1:GOTO95
140 PRINTDA + IN;G$;"-1":POKE517,0
150 STOP
```

Listing No. 1

How It works:

Line 5 G$ is set to 'DATA', var X set to 1, DIM var to the number of words you want to hide, I used 50 in my example. Line 7 sets the starting point of the generated data statements and should reflect free line numbers preferably at the end of your host program, line 8 is the increment value. Line 10 ends the input portion when a '*' is inputed to the question• "Word to hide" and moves to the output section in line 50. Line 40 increments x by 1 and starts the loop over again.

Lines 50-85 place the computer in the save mode, reset X to 1 and check for the end flag. Line 97 prints the line number selected in line 7 and prints G$ and the coded information, i.e. W is set to the length of the coded word. H$ is set to the letter in A$ for each increment of the for next loop. Line 112 prints the two letter ASCII code and returns for the next letter in A$. Line 115 increments the DATA statement by the number selected in line 8, X is incremented by 1 and loops to line 95 where end flag is checked. When the last word is coded the end flag is set and line 140 is executed giving an end of DATA flag for the decoding program.

The resulting data may be stored on tape and inputed to your word game or any program you may wish to hide data in. A file of ASCII data may be set up allowing children's spelling words, etc. to be inputed to the host program. The host program must also contain the following decoder program.

```
300 DIMX(50):RESTORE:X = 1:DIMA$(50):DIMJ
(50),R$(50),Y(50)
305 REM READ CODED WORDS CHECK FOR
END FLAG
310 READA$(X):IFA$(X) = "-1"THEN430
320 REM GET ASCII CODED DATA
330 FORJ = 1TOLEN(A$(X))STEP4:B$ = MID$(A$
(X),J,1) + MID$(A$(X),J + 1,1)
350 REM CONVERT DATA TO RAW ASCII
360 R = VAL(B$):R$(J) = CHR$(R):NEXTJ
400 REM ADD $ TOGETHER TO RETEVE
WORD
410 A$(X) = R$(1) + R$(5) + R$(9) + R$(13) + R$(17)
 + R$(21) + (21) + R$(25) + R$(29) + R$(33)
411 A$(X) = A$(X) + R$(37) + R$(41) + R$(45)
413 FORY51TO50STEP4:R$(Y) = " ":NEXT
415 REM PRINT WORD LIST
417 PRINTA$(X)
420 X = X + 1:GOTO310
430 STOP
```

Listing No. 2

How it works:

Line 300 DIM VAR to the maximum number of inputs required by the host program operation, restores the data pointer and sets X to 1. The coded data is read in line 310 and checked for end of data flag. Line 330 retrieves the ASCII code for each letter, i.e. sets B$ to the two character code representing one letter (89 = the letter 'Y') for the length of the data line. Line 300 converts the number code to the letter and line 410 retrieves the hidden word. Line 413 erases R$, (utilized in 410) in preparation for the next word. Line 417 prints the word list, shown here for example only. Line 420 increments X and returns to the beginning of the loop.

When the program is run, A$(x) is set to the hidden words, there are various methods utilized in games to randomly select the word used but with this

www.commodore.ca

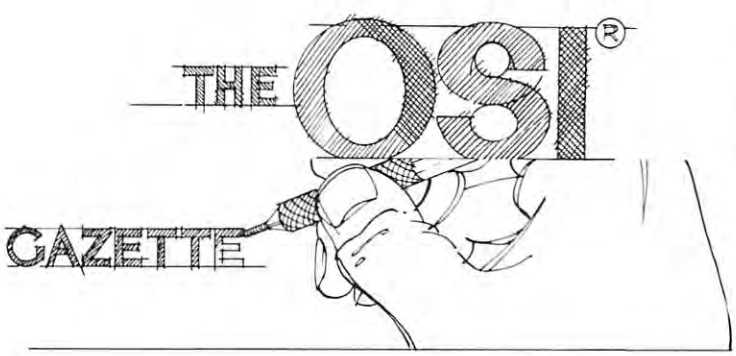
method you can just increment X by 1 each time the user wants to play again thus eleminating the possibility of the selection of the same word twice.

```
1000 DATA 67 79 77 80 85 84 69 3 2 73 73
1010 DATA 80 46 79 46 32 66 79 8 8 32 53 52 48 54
1020 DATA 71 82 69 69 78 83 66 7 9 82 79
1030 DATA 78 46 67 46 32 50 55 5 2 48 51
1050 DATA-1
```

Figure 1: Example of coded data statements generated by listing No. 1

```
OK
RUN300
COMPUTE II
P.O. BOX 5406
GREENSBORO
N.C. 27403
BREAK IN 430
OK
```

Figure 2. Decoding of data above by program listing 2 ©



O.S.I. Graphics Character Set

W. Blaine Garland

We have had our OSI Superboard II (with 8K memory) for six months now, and with its help, my sons and I are continuously discovering something new in the world of 6502 Single Board Computing.

Aside from the terrific low cost, one of the main reasons we chose the Superboard II was its extensive graphics capabilities. However, in attempting to demonstrate to friends all the possibilities of the graphics character set, we were severely limited by leafing through "The Challenger Character Graphics Reference Manual". The table lists all 256 characters. We then tried in vain to explain to our friends how they are called to video display.

Recently, we developed a short, BASIC Demonstration Program which calls each character in the CG-4 ROM to the screen consecutively. The characters are identified by the numeric variable Y, displayed with the typical POKE statement in line 110 and incremented on the screen with the FOR, NEXT loop at lines 80 and 170. The remainder of the program is essentially "window dressing".

Running the Demo Program lists each memory location in decimal and each character in the CG-4 ROM Character generator. It also indicates the two "spaces" in the set (locations 32 and 96) and the beginning and ending of the upper and lower case alphabet, numerals and punctuation which is the standard ASCII character set (locations 33 and 123 respectively). You can change the speed for incrementing characters on the screen by revising the FOR, NEXT timing loop in line 150. With the loop set at 1 to 500, the 256 character set can be displayed in about three minutes. This must approach the maximum attention span of any non-computer addict! You may also want to revise the program by deleting the ASCII characters and displaying only the 165 graphics and gaming symbols or vice-versa.

Incidentally, in developing the program, we had first tried calling the characters to the screen using the PRINT CHR$ (X) function. For some reason locations 10 and 13 would only print a blank space on the screen. We would appreciate another Challenger User's insight into the error of our ways.

To many readers, this program may seem simplistic. But to those of us who possess little experience and even less OSI documentation, it represents yet one more major step toward mastering the Superboard II and 6502 singleboard computing through "experimentation"!

So the next time a friend wants to know more about your micro-computer's graphics capabilities, demonstrate to them - **GRAPHICALLY**!

Program Listing
OSI Graphics Characters

```
10 FOR X = 1 TO 29 : PRINT : NEXT
20 FOR X = 54119 TO 54215 : POKE X, 32 : NEXT
30 PRINT " OSI GRAPHICS CHARACTERS" : PRINT
40 PRINT " IN CG-4 CHARACTER" : PRINT
50 PRINT " GENERATOR ROM" : PRINT: PRINT
60 FOR T = 1 TO 500 : NEXT
70 I=0
80 FOR Y = 1TO 255
90 I = I + 1
100 PRINT TAB(2) I TAB(7) CHR$(45)
110 POKE 54095, Y
120 IF Y = 32 THEN PRINT TAB (12) "(SPACE)": PRINT:
    PRINT TAB (12) "ASCII BEGINS"
140 IF Y = 96 THEN PRINT TAB (12) "(SPACE)"
150 IF Y= 123 THEN PRINT TAB (12) "ASCII ENDS"
160 FOR T = 1 TO 500: NEXT
170 PRINT
180 NEXT Y
190 END
```

©

www.commodore.ca

Atari Joysticks on the OSI C1P

Charles L. Stanford

One of the great advantages of the Ohio Scientific Challenger 1P and Superboard II computers is the easy yet effective graphics programming. The game symbols in the Character Generator ROM, plus the relatively simple 'POKE' programming of the screen refresh memory, opens many possibilities for games and other graphics simulations. The biggest disadvantage is the need to play through the keyboard in order to move the characters around the screen in a Gunfight, Tank, or Spacewar game.

This article will provide both construction and programming details plus a short keyboard input tutorial for interfacing the Atari joysticks to the C1P. In addition, almost every detail is identical for the C2, C4, etc. This joystick was chosen for several reasons, but primarily because Sears, Roebuck carries them at most retail stores for $9.99 each. They are also quite reliable and provide an easily interfaced digital output rather than the analog signal of most other such devices. The article will also include a generalized program for using the joystick with your own games, as well as the program patches (modifications) needed to convert the Space Invader game included on OSI's sample cassette SCX-102 to joystick play.

The Polled Keyboard

First, a little background on the OSI 600 board's polled keyboard. Unlike most other computer input keyboards, it does not convert a key actuation directly into the appropriate ASCII code in hardware. Rather, an input-output port at the address $DF00 (#57088), connected to an X-Y matrix keyboard, allows key closures to be detected and translated in software. (Note: In this article, binary numbers will be prefixed '%', Hex numbers '$', and decimal numbers '#'.) Each row of keys can be set 'On' or 'Off' by poking a binary number to the port address. For example, the instruction POKE 57099,127 actually sets the port input latches to the binary number %0111 1111. Thus rows 0 through 6 (starting with 0 on the right) are held at '1' and row 7 is at '0'. The computer program, whether in the ROM monitor or during a game in BASIC, then watches the port. If the key '7' is pressed, the program will see a %1111 1101 at the port. Key '1' would provide %0111 1111. More than one key being pressed simultaneously can also be detected. Pressing '1' and '3' would return the code %0101 1111. Each of these binary numbers would, of course, be translated automatically by the basic assembler into their equivalents in Decimal. Keys '1' and '3' being pressed at once would thus return a #95. The serious programmer should either develop a facility in converting numbers from binary to hex and decimal at will, or keep a conversion chart handy. Refer to your Graphics Reference Manual, Figure 3-2, for more details on the 600 Board polled keyboard's physical layout and electrical connections.

While you are programming the computer in BASIC, or if a running program asks for an INPUT, the monitor in ROM scans the keyboard constantly until a key closure is detected. Then the row and column are compared, and the appropriate ASCII code is returned to BASIC and to program storage. You can do the same thing in BASIC during a game by following the instructions in the Graphics Reference Manual for POKE'ing and PEEK'ing the keyboard as described above. But even better, you can connect a joystick directly to the keyboard matrix and simulate keyboard input through the motion of a control lever or pushbutton switch.

Keyboard Input Access Connections

Jack J4, located on the left front of the 600 Board, includes connections to seven columns and to rows 1, 6, and 7. This gives access to twenty keys, which is more than sufficient for even the most complicated game setup. This jack takes a 12-pin standard Molex connector, which can then be connected through a cable to any other common multi-pin socket, or directly to the game device. The C1P owner will probably prefer a more sophisticated socket arrangement attached to the outer case, while the Superboard II requires only the Molex jack.

A connector series which has become increasingly popular in recent years, generally known as the 'D', 'DB', or RS-232 types, has been chosen by Atari for their various interface plugs and sockets. These connectors come in various configurations, and can have 9, 15, 25, 37, or 50 pins. Atari uses the DB9S on the Joysticks, and the DB9P on the computer case. This connector series is designed, however, so that a DB25P will accept two of the DB9S's. Thus, I decided to use this connector for this project to allow easy wiring and permit future keyboard connections. These plugs are readily available from mail order houses such as Jade, Jameco, et al. Please note that 'plug' and 'socket' are the opposite you might expect; the plug mounts on the panel, and the socket is affixed to the cable end.

It is relatively easy to mount a DB25P above the keyboard. I chose a location on the upper vertical surface (above and to the right of the RUBOUT key), rather than on the rear panel. There is a cutout on the rear panel for an RS-232 connector, but I expect to use it for that purpose at some future time. To install the connector, completely dismantle the case, removing the board and carefully set it aside.

www.commodore.ca

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.

Four back 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.

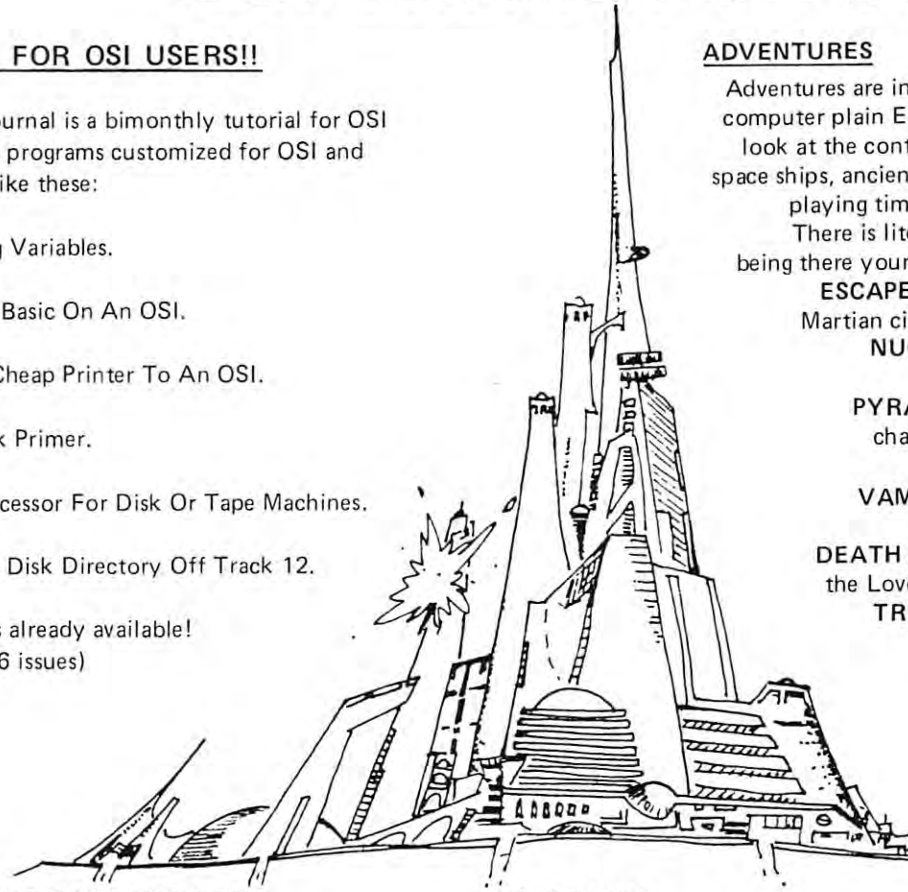
PYRAMID – Our most advanced and most challenging adventure. Takes place in our own special ancient pyramid.

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.

$14.95 each



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.), Software selectable scroll windows, two instant screen clears (scroll window only and full screen.), software choice 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. All that and it sells for a mesly $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 both an extended machine code monitor and a fix for the string handling bug in OSI Basic!! 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. Specify system! $59.95

STRING BUG FIX (replaces basic ROM chip number 3)
All this chip does is to replace the third basic ROM and correct the errors that were put into the ROM mask. $19.95

DATA SHEETS

OS65D LISTING
Commented with source code, 83 pages. $24.95

THE (REAL) FIRST BOOK OF OSI
65 packed pages on how OSI basic works. Our best selling data sheet. $15.95

OSI BASIC IN ROM
Ed Carlson's book of how to program in basic. Now available from Aardvark. $8.95

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!

REAL SOUND FOR THE C1P – and it's cheap! This bare board uses the TI sound chip to give real arcade type sound. The board goes together in a couple of hours with about $20.00 in parts. Bare board, plans, and sample program – $15.95

ARCADE AND VIDEO GAMES

ALIEN INVADERS with machine code moves – for fast action. This is our best invaders yet. The disk version is so fast that we had to add selectable speeds to make it playable.
Tape - $10.95 – 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

STARFIGHTER – a real time space war where you face cruisers, battleships and fighters using a variety of weapons. Your screen contains working instrumentation and a real time display of the alien ships. $6.95 in black and white - $7.95 in color and sound.

SEAWOLFE – this one looks like it just stepped out of the arcades. It features multiple torpedoes, several target ships, floating mines and real time time-to-go and score displays. – $6.95 in black and white $7.95 in color and sound.

SCREEN EDITORS

These programs all allow the editing of basic lines. All assume that you are using the standard OSI video display and polled keyboard.

C1P CURSOR CONTROL – A program that uses no RAM normally available to the system. (We hid it in unused space on page 2). It provides real backspace, insert, delete and replace functions and an optional instant screen clear.
$11.95

C2/4 CURSOR. This one uses 366 BYTES of RAM to provide a full screen editor. Edit and change lines on any part of the screen. (Basic in ROM systems only.)

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".

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

DISK CATALOGER
This utility reads the directory of your disks and makes up an alphabetic list off all your programs and what disks they are on. $14.95

MACHINE CODE RENUMBERER
(C2/4-MF only)
Renumbers all or part of a program at machine code speeds. – $15.95

VISA

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.

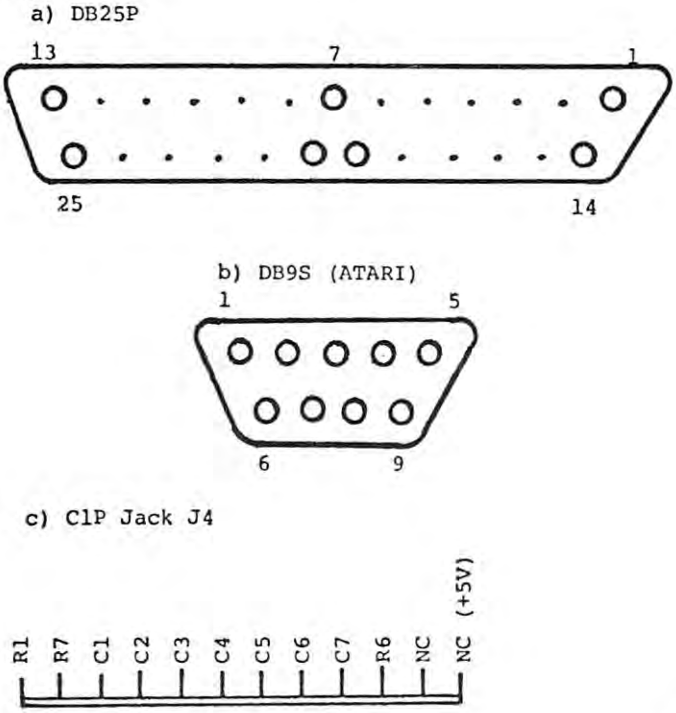
master charge

OSI **Aardvark Technical Services • 1690 Bolton • Walled Lake, MI 48088**
(313) 669-3110 or (313) 624-6316 OSI

www.commodore.ca

Mark the outline of the connector, and make a smooth, even cutout. I drilled several holes around the inside of the pattern, connected them with a side-cutter bit in an electric drill, and smoothed out the edges with a fine file. Be sure to rest the case on a soft surface such as an old towel, and center-punch all holes to avoid slipping and scarring the case with the drill bit. Remove all metal chips thoroughly to avoid shorting the PC board or power supply. Insert the connector, and drill the two mounting holes.

Figure 1



	DB25P		DB9S (L & R)		
	Pin	Connection	Pin	Color	Connection
	1	C4	1L	White	Up
	2	C5	2L	Blue	Down
	3	C6	3L	Green	Left
U	4	C7	4L	Brown	Right
P	5	NC (+5V)	5L	NC	-
P	6	NC (C1)			
E	7	NC (C2)			
R	8	NC (C3)			
	9	C4	1R	White	Up
R	10	C5	2R	Blue	Down
O	11	C6	3R	Green	Left
W	12	C7	4R	Brown	Right
	13	NC (+5V)	5R	NC	-
	14	C3	6L	Orange	PB
L	15	NC (R1)	7L	NC	-
O	16	R7	8L	Black	Common
W	17	NC (gnd)	9L	NC	-
E	18	NC			
R	19	NC			
	20	NC			
R	21	NC			
O	22	C3	6R	Orange	PB
W	23	NC (R1)	7R	NC	-
	24	R6	8R	Black	Common
	25	NC (gnd)	9R	NC	-

Next, connect color-coded wires of sufficient length to each of the connector pins you plan to use. Figure 1 shows the basic connections needed for both the DB25 and keyboard jack J4. You can use either a ten or twelve pin molex connector at J4, but be sure to match-mark it so that the orientation is always correct. You will note that the pins marked 'NC' on Fig. 1 have another connection indicated in parentheses. To save work later, I selected a variety of useful signals and made the connections to the DB25. At some future time, a ten-key input or other useful device can be easily plugged in, using a DB15 of DB25 socket. I ''borrowed the +5 volts from a pad near J4 in case my future peripheral needs power, but put a 100 ohm ½ watt resistor in series to avoid damage in case of a short circuit.

I use surplus ribbon cable as a cheap source for stranded color-coded wire. If you are adept at soldering, and have four hands, leave the ribbon cable intact. Otherwise, carefully separate the individual wires, solder them into proper place, and wrap the bundle every three inches. Double-check each connection before reassembling your computer.

Superboard II owners can just cut the DB9's off the end of the joystick cables and solder the wires directly to the molex connector. Figure 1 shows Atari's color coding, but it's best to check it, as production standards can change.

Readers who have seen articles on connecting Atari joysticks to computers such as the PET, with user ports, may try to combine the 'shoot' function with 'up' and 'down' as described. It won't work! There is already a diode in each keyboard row line, and the combined voltage drop across two diodes in series exceeds the threshold (trigger) voltage of 74LS integrated circuits. The method does work well with ports, and allows both joysticks to be connected to the eight data lines of one port address.

Testing the Joystick

It's pretty easy to run an elementary test of the completed circuit, as can be seen from the decoding of Table 1. If you Cold-start your C1P, the Up, Down, Left, Right, and Shoot functions will write the figures in Row 7 to the screen. The diagonal motions, and combinations of motion plus shoot, will give no screen indication, as they are the equivalent of multiple simultaneous key presses. Key in the following program to test all modes:

```
5 FOR S = 0 TO 30 : PRINT : NEXT
10 DIM G(16) : POKE 530,1
20 POKE 57088,127
25 REM-RT JYSTK POKE 57088,191
30 FOR X = 0 TO 16 : READ G(X) : NEXT
40 DATA 83,0,0,0,0,0,19,17,18,0,21,23,22,0,20,16,79
50 Y = PEEK(57088)
60 Z=Y OR 247:IF Z=247 THEN X=0:GOTO 80
70 X=(Y/16) + 1
80 IF G(X) = 0 THEN PRINT "ILLEGAL INPUT":
   GOTO 100
90 POKE 54134,G(X)
100 FOR T = 0 to 100 : NEXT
110 GOTO 50
```

www.commodore.ca

Table 1 - Output codes for nine-position joystick:

Position Output Codes

JOYSTICK POSITION	EQUIVALENT KEYS DOWN	OUTPUT CODES: SHOOT BUTTON OPEN BINARY	HEX	DECIMAL	SHOOT BUTTON CLOSED BINARY	HEX	DECIMAL
	4 -	1110 1111	EF	239	1110 0111	E7	231
	3 :	1101 1111	DF	223	1101 0111	D7	215
Center	None	1111 1111	FF	255	1111 0111	F7	247
	2 0	1011 1111	BF	191	1011 0111	B7	183
	1 9	0111 1111	7F	127	0111 0111	77	119
	2,3 0,:	1001 1111	9F	159	1001 0111	97	151
	2,4 0,-	1010 1111	AF	175	1010 0111	A7	167
	1,4 9,-	0110 1111	6F	111	0110 0111	67	103
	1,3 9,:	0101 1111	5F	95	0101 0111	57	87

First, check the program for errors by using the equivalent key inputs from Table 1. Then, plug the joystick in and try all nine quadrants. Test both positions of the socket; note that it may be necessary to shave some Atari connectors with a file or sharp knife to fit the plugs. If any errors appear, the schematic and the output chart should be compared with the result of the test program to ascertain the reason and the proper corrective action required.

Game programming for the Joystick

It helps to have some knowledge of Boolean Algebra for the programming, but you can probably muddle through it as I did at first. The Boolean AND and OR operators in Basic can be very handy for masking unwanted inputs. In Boolean Algebra, 0 OR 0 =; 0 OR 1 = 1; OR 1 = 1. This operation is handled bit-by-bit, with no carry as occurs in binary addition. So %1110 0111 OR %1111 0111 OR %1111 0111 © %1111 0111. Thus an input which calls for both a move and a shoot can be masked so the computer sees only the shoot. The first binary number shown above results when the joystick is in the UP position *and* the Shoot button is pressed. If you want a program to stop moving and shoot whenever the button is pressed, just mask all but Column 3 by OR'ing with %1111 0111. This is done in BASIC by the following sequence:

```
200 POKE 530,I:POKE 57088,127
210 Y = PEEK(57088)
220 Z = Y OR 247
230 IF Z = 247 THEN 400
240 GOTO 500
250 REM - LINE 400 IS A SHOOT ROUTINE
260 REM - LINE 500 IS A MOVE ROUTINE
```

Thus, line 220 makes all but column 3 by OR'ing the input with %1111 0111 (#247), and line 230 checks to see if that bit is 0 which would mean that the key at row 7, column three is pressed. In this case, of course, it would mean that the shoot button is pressed.

A generalized subroutine in BASIC which would allow a single-square object to be moved around the screen by a game program is as follows:

```
42000 DIM G(16):FOR X = 1 TO 16:READ
G(X):NEXT :S = 53743
42010 KEY = 57088:POKE KEY, 127
42020 P = PEEK(KEY)
42030 PP = P OR 247
42040 IF PP = 247 THEN 400:REM-SHOOT
42050 X = (240 AND P)/16 + 1
42055 REM-LINE 42050 CONVERTS P TO NUMBERS
1 THRU 16
42060 E = G(X)
42070 GOTO 500 : REM-MOVE
42075 REM-LINE 42080 IS SCREEN MOVE OFFSETS
42080 DATA 0,0,0,0,0,33,-31,1,0,31,-33,-1,0,32,-32,0
```

A program which would use the above subroutine to move the object could be as follows:

```
500 REM-MOVE ROUTINE
510 S0 = 53248
520 SS = S + E:IF SS < S0 OR SS > S0 +
1024 THEN 42010
525 REM-LINE 520 KEEPS OBJECT IN SCREEN
MEMORY AREA
530 POKE SS,161:POKE S,32
540 S = SS
550 GOTO 42010
```

Line 530 POKE's the object to the new location SS, then blanks the old location S. You can vary the speed of movement by inserting a time delay such as "545 FOR T = 0 TO 99:NEXT".

A typical shoot routine could be as follows:

```
400 REM-SHOOT ROUTINE
410 FOR X = 1 TO 16:POKE S + G(X),188:NEXT X
420 FOR T = 0 TO 999:NEXT T
430 FOR X = 1 TO 16:POKE SS + G(X),32:NEXT X
440 Z = Z + 1:IF Z = 5 THEN POKE 530,0:STOP
450 GOTO 42010
```

To put all three sequences together, replace lines 200 through 260, above, with:

```
200 REM-MOVE AND SHOOT DEMO PROGRAM
210 RESTORE:Z = 0:POKE 530,1:GOTO 42000
```

Thus, combining lines 200 through 42080 provides a program which will allow the user to move a block around the screen by either key or joystick input, as well as simulate that block being destroyed by an explosion.

A careful examination of the OSI demo program mentioned earlier reveals a very similar action, except that more than one character is involved. Of course, Space Invader has several other routines such as move and shoot-back at random, scoring, etc.

Program Listing I shows the necessary modifications to the cassette program for the C1P for joystick conversion. It would have been nice to list the complete program; however, copyright laws forbid such

www.commodore.ca

publication without permission of the author.

When you run the program, you may be pleasantly surprised to find that not only have you added four more directions of movement (the diagonals), but play is speeded up by a factor of about 1.5.

```
2      REM - ATARI JOYSTICK 9/80
6      DIM G(16): FOR X=1 TO 16:
       READ G(X): NEXT
10     Z1=1
70     POKE KEY,127: P=PEEK(KEY): PA=P OR 247:
       IF PA=247 THEN 1100
71     X=(240 AND P)/16+1
72     DELETE
73     DELETE
74     DELETE
80     DELETE
1000   IF X=16 THEN 1050
1005   DELETE
1010   IF FND(H+G(X)-1)=0 OR FND(H+G(X)+1)=24
       THEN 50
1015   IF H+G(X)>54268 OR H+G(X)<53349 THEN 50
1020   H=H+G(X): I=H+1: J=H-1
1040   POKE I-G(X),V: POKE J-G(X),W
1047   E=X: GOTO 50
1050   E=0: GOTO 50
1110   GOTO 50
10005 DATA 0,0,0,0,0,33,-31,1,0,31,-33,-1,0,
      32,-32,0
20021 PRINT:PRINT:PRINT:PRINTTAB(6);"SPACEWARS":
      PRINTTAB(6);"----------"
20022 PRINT
20045 PRINT"USE KEYS AS FOLLOWS:"
20050 PRINT:PRINT TAB(8);"4
20052 PRINT TAB(8);"1 4
20055 PRINTTAB(8);"1
20057 PRINTTAB(8);"1 3
20060 PRINTTAB(8);"3
20062 PRINTTAB(8);"2 3
20065 PRINTTAB(8);"2
20067 PRINTTAB(8);"2 4
20070 PRINT"  STOP  NONE
20072 PRINT"  SHOOT  5
20075 ZZ=53800
20077 FOR X=0 TO 7
20080 POKE ZZ+X*32,X+16
20082 NEXT
20085 PRINT"CAREFUL, HE SHOOTS BACK!
20087 INPUT C$
```

©

Missing Listing from Compute II, #3 Fast Graphics by Charles Stanford

```
Listing 3: BASIC Program
  5 REM- CHOO CHOO COLLISION
 10 REM- FAST GRAPHICS DEMO
 15 GOSUB 100
 20 D=99
 25 A=59:B=29:C=29:POKE609,210
 30 GOSUB 50
 35 A=156:B=123:C=11:POKE609,209
 40 GOSUB 50
 45 GOTO 200
 50 REM- SCREEN WRITE SUBROUTINE
 55 FOR X=0 TO C
 60 POKE 11,34:POKE 254,96:POKE 608,A
 65 A=A-1:B=B+1
 70 X=USR(X)
 75 POKE 11,56:POKE 254,157:POKE 669,B
 80 X=USR(X)
 85 FOR T=0 TO D:NEXT T
 90 NEXT X
 95 RETURN
100 REM- MACHINE GRAPHICS WRITE TO RAM
    SUBROUTINE
110 RESTORE
115 POKE 11,34: POKE 12,2: POKE 254,96: POKE 255,2
120 FOR P=0 TO 61: READ M: POKE 546+P,M:NEXT P
130 DATA 160,0,169,32,153,0,211,153,0,210,153,
    0,209,153,0
135 DATA 208,200,208,241,234,234,234,160
    0,177,254,141,86,2,200
140 DATA 177,254,141,87,2,200,177,254,
    170,200,224,254,240,236,224,255
145 DATA 240,8,177,254,200,157,68,209,
    208,236,96,234,234,234,234,234
148 REM- GRAPHICS FIGURE TABLE
149 FOR P=0 TO 60: READ M: POKE 608 +P,
    M: NEXT P
160 DATA 155,209,1,2,3,167,4,157,5,161,8,167
165 DATA 32,165,33,161,34,161,35,161,37,155,38,
    176,39,161,40,161
170 DATA 64,166,65,161,67,161,68,161,69,128,70,
    161,71,161,72,161
175 DATA 96,176,97,224,98,225,99,226,102,226,
    104,226,255
180 DATA 131,209,0,165,3,161,4,156,5,165,7,2
182 DATA 32,161,33,161,34,178,35,155,36,161,37,
    161,38,161,39,161,40,167
184 DATA 64,161,65,161,66,161,67,128,68,161,69,
    161,70,161,71,161,72,168
186 DATA 96,226,98,226,101,226,102,224,103,225,
    104,178
187 DATA 255,0,0,0,0,0,0,0,0,0
199 RETURN
200 REM- EXPLOSION
210 GOSUB 300
220 Z=53711
230 FOR X=1 TO 6
240 FOR Y=1 TO 8
250 POKE Z+X*X(Y),42
260 NEXT Y
270 NEXT X
280 FOR T=0 TO 2500: NEXT
290 END
300 REM- EXPLOSION DATA
310 X(1)=-33: X(2)=-32: X(3)=-31
320 X(4)=-1: X(5)=1
330 X(6)=31: X(7)=32: X(8)=33
399 RETURN
```

©

SOFTWARE FOR OSI

Video Games 1 .... **$15**
Three games. Head-On is like the popular arcade game. Tank Battle is a Tank Game for two to four. Trap! is an enhanced blockade style game.

Video Games 2 .... **$15**
Three games. Grelin Hunt is an arcade-style game for one to three. Gunfight is a duel of mobile artillary. Indy is a race game for one or two.

Adventure: Marooned In Space **$12**
An adventure that runs in 8K! Save your ship and yourself from destruction.

Dungeon Chase **$10**
A real-time video game where you explore a twenty level dungeon.

Board Games 1 .... **$15**
Two games. Mini-gomoku is a machine language version of five stones gomoku. Cubic is a 3-D tic-tac-toe game. Both with graphics.

Super- Biorhythms **$15**
A sophisticated biorhythm program with many unique features.

C1 Shorthand **$12**
Use only two keys to enter any one of the BASIC commands or keywords. Saves much typing when entering programs. Written in machine language.

Disassembler **$12**
Use this to look at the ROMs in your machine to see what makes BASIC tick. Reconstruct the assembler source code of machine language programs to understand how they work. Our disassembler outputs unique suffixes which identify the addressing mode being used, no other program has this!

Cassette Data Base Manager **$15**
Here's a DBM that will work on non-disk systems. Most of the features of disk based DBMs are here. Does NOT use DATA statements as similar programs do.

For all BASIC-in-ROM systems. Selected programs available on disk. Color and sound on video games.

Send for free catalog listing many more programs.

ORION SOFTWARE ASSOCIATES

147 Main Street
Ossining, NY 10562

www.commodore.ca

Breakout for OSI1P

Charles Stewart

```
 6 POKE605,0:FORK=611TO625:POKEK,32:NEXT
K:POKEK,255
 10 POKE515,0:CLEAR
 12 FORX=1TO10:READI(X):NEXT
 15 DATA30,31,32,33,34,-30,-31,-32,-33,-
34
 20 REM SET UP QUICK CLEAR
 30 POKE11,34:POKE12,2:POKE574,96
 40 FORX=0TO27:Y=PEEK(65036+X):POKE546+X
,Y:NEXT
 41 X=USR(X):INPUT"INSTRUCTIONS";A$:IFAS
C(A$)=89THEN8600
 47 X=USR(X):PRINT"HIT ESCAPE TO START"
 48 IFPEEK(57088)=254THENR8=RND(1):GOTO4
8
 50 X=USR(X):TP=53445:BOT=54149
 51 INPUT"NAME PLEASE";A$:PRINT:PRINT
 52 INPUT"DIFFICULTY LEVEL";DI
 53 DI=DI*INT(10*RND(1)):X=USR(X)
 54 CH=0
 60 FORSC=TP-32TOTP-8:POKESC,96:NEXT
 65 FORSC=TPTOTP+217:POKESC,159:NEXT
 70 FORSC=53437TO54173STEP32:POKESC,143:
NEXT
 80 FORSC=53412TO54148STEP32:POKESC,136:
NEXT
 100 BA$="BALL":K=611
 102 IFPEEK(K)=255THEN110
 105 X1=PEEK(K):POKE54160+K-611,X1:K=K+1
:GOTO102
 110 BALL=54151:FORLE=1TOLEN(BA$):POKEBA
+LE,ASC(MID$(BA$,LE,1)):NEXT
 120 KEY=57088:POKE56900,1
 121 S$="SCORE=":S=53390
 122 FORW=1TOLEN(S$):POKES+W,ASC(MID$(S$
,W,1)):NEXT
 125 FORSC=54115TO54115+32:POKESC,131:NE
XT
 130 SYM=54060
 131 VI=53965+INT(RND(1)*10)
 132 B=1
 133 X=I(8)
 134 D$=STR$(B):D=54155
 135 POKESYM,155
 136 FORY=2TOLEN(D$):POKED+Y,ASC(MID$(D$
,Y,1)):NEXT:POKED+Y,32
 139 U$=STR$(CH+B*5):FORE=2TOLEN(U$):POK
EE+S+7,ASC(MID$(U$,E,1)):NEXT
 140 POKEKE,254:ST=255-PEEK(KE)
 150 IFST=5THENSYM=SYM-1:POKESYM+1,32
 155 IFST=3THENSYM=SYM+1:POKESYM-1,32
 160 IFPEEK(VI+1)=143THENX=-33:POKEVI,96
:VI=VI+X:POKEVI,226
 170 IFPEEK(VI-1)=136THENX=-30:GOTO300
 190 IFPEEK(VI+32)=155THEN2030
 210 IFPEEK(VI-32)=159THEN3000
 220 IFPEEK(VI+32)=131THENGOSUB6000:B=B+
1:GOTO133
 295 IFPEEK(VI-32)=96THEN8000
 300 POKEVI,32:VI=VI+X:POKEVI,226
 305 FORT=1TODI:NEXT:GOTO135
 2030 G=INT(RND(1)*10):IFG<7ORG>10THEN20
30
 2035 X=I(G):GOTO300
 3000 X=32:CH=CH+1:POKEVI-32,32:GOTO300
 6000 FORR9=169TO32STEP-1:POKEVI,R9:NEXT
R9
 6010 VI=53965+INT(RND(1)*10):RETURN
 8000 CH=CH+(B*5):X=USR(X):PRINT"IT TOOK
 YOU"CH"TO BREAKOUT ";A$
 8002 IFR2=0THENPOKE605,CH:GOTO9000
 8003 PRINT"THE LOW SCORE IS";PEEK(605):
PRINT:PRINT
 8004 IFCH<PEEK(605)THEN9000
 8005 FORA=1TO10:PRINT:NEXT
 8010 PRINT"CARE TO TRY AGAIN"
 8015 R2=R2+1
 8020 INPUTA1$:IFLEFT$(A1$,1)="Y"THEN50
 8500 END
 8600 X=USR(X):PRINT"THE OBJECT IS TO BR
EAK-"
 8610 PRINT"OUT..LOW SCORE WINS!!":PRINT
"SHIFTS CONTROL PADDLE
 8620 PRINT:PRINT"EACH BALL COUNTS 5 POI
NTS":PRINT
 8630 INPUT"READY TO START";A$:GOTO47
 9000 POKE605,CH
 9010 FORK=611TO611+LEN(A$)-1:POKEK,ASC(
MID$(A$,K-610,1)):NEXTK
 9020 POKEK,32:POKEK+1,32
 9030 A$=STR$(CH):FORG=2TOLEN(A$):POKEK+
G,ASC(MID$(A$,G,1)):NEXTG
 9040 POKEK+G,255
 9050 PRINT:PRINT:GOTO8010
OK
```

©

All About

OSI

BASIC-IN-ROM

Reference Manual

computeII.: *"...any of several sections of this very well presented manual are worth the purchase price"*

Aardvark Journal: *"It is the book you were hoping was packed with your computer at the factory"*

PEEK(65): *"in goes far enough...to hold the interest of advanced programers as well as novices."*

Complete, concise, accurate, detailed. USR(X). Bugs. Tapes: BASIC, autoload and homemade. Source code and variable tables above $0300. Memory maps: $00,01, 02,A000-BFFF. Line-by-line description of MONITOR in $FE,FF.

$8.95 from your dealer or postpaid from me.

Edward H. Carlson
3872 Raleigh Dr.
Okemos, MI 48864

www.commodore.ca

THE PET® GAZETTE

Basic CBM 8010 Modem Routines

Jim Butterfield, Toronto

The programs given on page 7 of the 8010 Modem Operator's Manual don't seem to do the job. In particular, the ASCII interface program often crashes; prints peculiar things if you are receiving parity characters; and drops line characters from time to time.

Here are a couple of replacement programs that should do the job better.

ASCII Interface

Set the modem switches to OR (Originate) and HD (Half Duplex). One exception: if you're working an "echoplex" type of system, the distant computer will repeat back everything you send; in this case, set the switch to FD (Full Duplex).

The program takes a few seconds to set up its translation arrays. You may start the program before telephone connection is established.

Special control characters can be set up, depending on your needs. Note, for example, that the delete character has been implemented in this program: PET's delete, decimal value 20, will be translated to ASCII backspace, decimal value 8, and vice versa; you can see the coding on line 210. You may implement your own to suit the needs of the computer or network. To enable Control-P, more accurately known as DLE (Data Link Escape) you might code: T(176) = 16. This would translate PET's shifted-zero character, a square-corner with bit value 176, to the ASCII DLE character, value 16.

PET-to-PET Interface

Both users should set their modem switches to HD (Half Duplex). One user should set OR (Originate), and the other AN (Answer); it doesn't matter which user sets what, so long as they are different. Communication is two-way in either case.

Cursor controls, reverse screen and graphics features are supported. A user can clear both screens with the CLR key.

The biggest operational problem is making sure you don't both try to talk at the same time. There's no flashing cursor to prompt you. You'll soon get used to waiting for a pause from the other PET before sending your own stuff.

General Comments

The business part of these programs - lines 300-320- are under severe time constraints. If you modify the programs, check carefully to make sure you don't start losing the occasional character incoming from the line.

These programs are quite simple; they convert your PET into a CRT terminal. That's not a cost-effective way to use a PET (terminals are cheaper) and eventually you should anticipate fitting more sophisticated programs which will allow you to send and receive programs and files.

For communications to an ASCII system:

```
100 REM 8010 INTERFACE  JIM BUTTERFIELD
110 REM FOR ASCII LINES
120 REMARK: SET SWITCH TO HD
200 DIM F(255),T(255)
210 FOR J = 32 TO 64 : T(J) = J : NEXT J : T(13) = 13 :
    T(20) = 8
220 FOR J = 65 TO 90 : K = J + 32 : T(J) = K : NEXT J
230 FOR J = 91 TO 95 : T(J) = J : NEXT J
240 FOR J = 193 TO 218 : K = J-128 : T(J) = K : NEXT J
250 REM ADD EXTRA FUNCTIONS HERE
260 FOR J = 0 TO 255 : K = T(J) : IF K THEN F(K) = J :
    F(K + 128) = J
270 NEXT J
280 POKE 1020,0 : POKE 59468,14
290 OPEN 5,5 : PRINT "ASCII I/0 READY"
300 GET A$ : IF A$<>"" THEN PRINT#5,CHR$
    (T(ASC(A$)));
310 GET#5, A$ : IF ST = 0 AND A$<>"" THEN
    PRINT CHR$(F(ASC(A$)));
320 GOTO 300
```

For communications to another PET:

```
100 REM 8010 INTERFACE  JIM BUTTERFIELD
110 REM  FOR PET INTERCOMMUNICATION
120 REMARK: SET SWITCH TO HD
280 POKE 1020,0 : POKE 59468,14 if text mode desired
290 OPEN 5,5 : PRINT "PET I/0 READY"
300 GET A$ : IF A$<>"" THEN PRINT#5,A$;
310 GET#5,A$ : IF ST = 0 THEN PRINT A$;
320 GOTO 300
```

©

www.commodore.ca

Increase your PET's IQ™ for $39.95.

Speed up your PET programming with The BASIC Programmer's Toolkit,™ now only $39.95.

Don't waste valuable programming time if there's an easier way to go. Here it is: The BASIC Programmer's Toolkit, created by Palo Alto ICs, a division of Nestar. The Toolkit is a set of super programming aids designed to enhance the writing, debugging and enhancing of BASIC programs for your PET.

The BASIC Programmer's Toolkit has two kilobytes of ROM firmware on a single chip. This extra ROM store lets you avoid loading tapes or giving up valuable RAM storage. It plugs into a socket inside your PET system, or is mounted on a circuit board attached on the side of your PET, depending on which model you own.

There are basically two versions of PET. To determine which Toolkit you need, just turn on your PET. If you see ***COMMODORE BASIC*** your PET uses the TK-80P Toolkit. If you see ###COMMODORE BASIC###, your PET uses the TK-160 Toolkit. Other versions of the BASIC Programmer's Toolkit are available for PET systems that have been upgraded with additional memory.

How Toolkit makes your programming easier:

FIND locates and displays the BASIC program lines that contain a specified string, variable or keyword. If you were to type *FIND A$,100–500,* your PET's screen would display all lines between line numbers 100 and 500 that contain *A$.*

RENUMBER renumbers the entire program currently in your PET. You can instantly change all line numbers and all references to those numbers. For instance, to start the line numbers with 500 instead of 100, just use *RENUMBER 500.*

HELP is used when your program stops due to an error. Type *HELP,* and the line on which the error occurs will be shown. The erroneous portion of the line will be indicated in reverse video on the screen.

These simple commands, and the other seven listed on the screen, take the drudgery out of program development work. And for a very low cost. The BASIC Programmer's Toolkit costs as little as $39.95, or at most, $59.95.

Get the BASIC Programmer's Toolkit and find out how quick and easy program development can be. See your local PET dealer or send this coupon in today.

PALO ALTO ICs
A Division of Nestar Systems Incorporated

The Toolkit is fully assembled. It is not a kit and requires no special tools to install.

PET™ is a trademark of Commodore Business Machines, Inc. The BASIC Programmer's Toolkit™ is a trademark of Palo Alto ICs, a division of Nestar Systems, Inc.

I want to save programming time and money.
Send me The BASIC Programmer's Toolkit that will give my PET 10 new and useful commands.
Fill in the appropriate line below:

Qty.________TK-160 Toolkit(s) @ $39.95 each

Qty.________TK-80P Toolkit(s) @ $59.95 each

Want to charge it? Call (415) 493-TOOL, or fill out the form below.

Enclosed is a ☐ money order ☐ check

(If charging): ☐ Bill VISA ☐ Bill Master Card.

Charge Card________________ Exp. Date________

Master Card Interbank Number________________

Signature________________

Please include the amount of the Toolkit, plus $2.50 for shipping and handling. Please allow 4-6 weeks for delivery. SATISFACTION GUARANTEED, OR SEND IT BACK WITHIN 10 DAYS OF RECEIPT AND PALO ALTO ICs WILL REFUND YOUR MONEY.

SEND TO:

Name________________

Address________________

City________________State________Zip________

Phone________________

MAIL TO: Palo Alto ICs
A Division of Nestar Systems, Inc.
430 Sherman Avenue
Palo Alto, CA 94306
(415) 493-TOOL

Dealer inquiries invited.

www.commodore.ca

Programmer's Notes for the CBM 8032

Roy Busdiecker

Several good articles describing major features of the CBM 8032, have already appeared (Butterfield Reports: The 8032, by Jim Butterfield, COMPUTE Issue 5, July/August 1980; and New Additions to the Commodore Line, by Robert W. Baker, Kilobaud Microcomputing, July 1980). There are quite a few features, however, which were not mentioned in those articles and will be of interest to those who own or are contemplating purchase of the new machine.

New Functions from Keyboard

My most recent (and most exciting) discovery is the fact that many of the new screen-editor functions (scroll down, delete line, insert line, etc.) can be activated directly from the keyboard, without the necessity of doing a PRINT CHR$ (XX) as described in the articles. The trick is simply to press the right combination of keys simultaneously. The combinations are shown in Figure 1. In some cases, it doesn't matter which key is pressed first; however, it's generally safer to press the key listed in the left column first.

Abbreviation	Meaning (Key)
DE	Delete
ES	Escape
LA	Left Arrow
LS	Shift Key on Left Side
OR	Off/Reverse
RS	Shift Key on Right Side
SH	Either Shift
TA	Tab
UA	Up Arrow
k	Key on Alpha-Numeric Keyboard
p	Key on Numeric Keypad

Function	Keys
Condensed graphics	LS RS 2k
Scroll down	LS ES K
	LS TA I
	LS 1k UA
Erase from beginning of line to cursor	LS LA 3p
	SH TA LA DE
	SH LA Q 4p
	SH LA A 6p
	SH LA Z 2p
	SH ES LA 5p
	SH OR LA 1p
Erase to end	LA Q 4p
	LA A 6p
	LA Z 2p
	ES LA 5p
	OR LA 1p
	TA LA DE
Delete line	ES OR K
	OR TA I
	OR 1k UA
	OR Q O
	OR A L
Insert line	SH ES OR K
	SH OR TA I
	SH OR 1k UA
	SH OR Q O
	SH OR A L
Set top left corner of window	Z A L
	Z ES K
	Z 1k UA
Set bottom right corner of window	SH Z A L
	SH Z ES K
	SH Z 1k UA

Figure 1. Keyboard Combinations for Special Screen Editor Functions

Calling the Monitor

Those who make heavy use of the built-in monitor can enter it with a SYS 54386. This mode of entry gives a "call" entry rather than the "break" entry you get with a SYS 1024. There are two observable differences between the two forms. A "call" entry gives a *C message on the screen, and does not change the value in the stack pointer (SP). A "break" entry gives a *B message, and decrements the value in the stack pointer by two. The "break" feature was not designed as the normal method for getting into the monitor, but rather as a tool for machine language programming. It's possible that if you went back and forth from BASIC to monitor many times using the SYS 1024 "break" entry, that you could run out of stack pointer space unnecessarily, although it's a rather unlikely occurence. Incidentally, for the older PET/CBM 2001-16 and -32, the "call" entry for the monitor is SYS 64785.

Automatic Program Adjustments

Many folks use location 50003 to allow a program to figure out what kind of PET/CBM computer that it's running on. PRINT PEEK (5003) gives a value of 0 on "old" PET's (version 1, BASIC 2.0), a value of 1 on "new" PET/CBM (version 2, BASIC 3.0), and now a value of 160 on the CBM 8032 (BASIC 4.0). Since many page zero locations in 8032 are the same as in the "new" PET/CBM's, some programs designed to run on either "old" or "new" versions can be adapted for the 8032 as shown in Figure 2.

Original program

```
10 PV = PEEK(50003)
20 REM:  =0 for OLD PETs,  =1 for NEW
```

Modified for 8032

```
10 PV  =  PEEK (50003)
15 IF PV = 160 THEN PV = 1:?"Program running on
CBM 8032"
20 REM:  =0 for OLD,  =1 for NEW,  =160 for 8032
```

Figure 2.

www.commodore.ca

We Saved a piece for you

(the Commodore Software/Hardware Dealer)

Computer cake enjoyed by those at initial meeting.

We're sorry you missed our initial meeting, but there is still time for you to participate.

DISC, an organization of international scope, has been formed to find, evaluate, develop, and distribute quality software worldwide for CBM systems.

Only $250 allows you to participate and benefit as a member if you act now! Effective January 1, 1981 the cost will be $500.

Don't delay. Send the coupon with your check today and save!

DISC
Attn: Alex Amor
President
50 Northwest 68 Avenue
Miami, FL 33126

☐ Yes, I want quality software. Put me down for membership. My check for $250 is enclosed.

☐ Sounds good. But I need more info.

☐ Send me information on how I can have my software/hardware listed and/or evaluated.

Name____________________

Address____________________

City, State, Zip____________________

DISC
DEALER INFORMATION SYSTEMS

www.commodore.ca

Of course, this modification will not adapt all programs for the 8032. I've seen very few programs for 40-column machines whose output looks "right" on the 80-column unit (those which do are the ones without sophisticated graphics or formatting). If the program uses built-in routines from the PET/CBM ROM, it will take more effort to find the routine in the 8032 and modify the program to use it.

Hidden Memory

As in previous machines, the screen memory appears to "use up" memory addresses from 32768 to 36863, although only the first 2000 of those are "real" screen memory addresses. Another 2000 are "image" addresses, due to the incomplete decoding of those addresses. Of particular interest are the 48 addresses from 34768 through 34815 which do not appear to be used for anything. That memory space could be used for short machine language routines, or data values that need to be tucked away where BASIC can't hurt them.

One bug I discovered in the 8032 is that a PRINT "[HOME]" often returns the cursor to the second line on the screen, rather than the first.

It was very frustrating to me to discover that many of the excellent machine language tools I've obtained via Jim Butterfield and Carl Moser do not work on the 8032. For those fortunate enough to have access to a 2040 disk drive, a 2001-32, and an 8032 all at the same time, it's possible to create a "host-target environment" or development system for the 8032.

Old Tools for New Programs

The 8032 and 2001-32 can both be connected to the 2040 using the IEEE-488 ports and the appropriate cables. A program "saved" to disk from one machine can be loaded into the other, and the transfer will work either way. You must be careful, however, not to have both computers trying to access the disk at the same time, or the system will get locked up. I've also experienced lockups when one of the computers is running certain machine language programs.

If you want to create an assembly language program for the 8032, you can use a good assembler (like the MAE from Eastern House Software) running on the 2001-32. After assembling the program in the 2001-32, use the built-in monitor to save the resulting machine language to disk. When the disk file is then loaded into the 8032, it will go into the memory locations corresponding to those from which it was saved.

Another thing I wanted to do was to look at the ROM in the 8032. Unfortunately, the only disassembler I had that would run on the 8032 was written in BASIC, and was exceedingly slow. On the other hand, I had several machine language disassemblers that were quite fast, but would not run on 8032. The solution was to copy a block of 8032 ROM, for example $B000 to $BFFF, into free RAM, say $1000 to $1FFF. This can be done in command mode with a statement like

```
FOR I = 0 TO 4095:POKE 4096 + I,PEEK
(45056 + I):NEXT
```

When this is finished, we use the 8032's monitor to save the copy ($1000 to $1FFFF), which can then be loaded into the 2001-32 for examination. The choice of locations, obviously, must be such that it will not interfere with any of the tools being used to examine the code.

ROM Features

The monitor in the 8032 is very similar to that in the 2001-32, except for having been relocated. This is both good and bad. It's good because the 2001-32 monitor is documented, which allows us to figure out some of the ROM routine locations in the 8032 which correspond to known routine locations in the 2001-32. It's bad because there are many improvements which should have been made. It's a shame to waste half the screen, when we could be seeing twice as many locations on the 80 column machine. It's also a shame to have such limited capabilities in a monitor, when so many good ones are available.

In the 8032, the operating system ROM starts at $B000 rather than $C000, which means there are only two free ROM sockets. Obviously, Word Pro 4 will take up at least one of those when it appears (this is being written in mid-August, and we've not been able to obtain a production copy yet).

Reader Feedback

We expect to be learning many more features of the 8032 in coming months, especially when we are able to get one of the new 8050 disk drives and test its interactions with old and new computers.

Any COMPUTE readers who would like to contribute their discoveries may forward them to me, and I'll incorporate them in a future article (giving credit to the first contributor of each item). I would be especially interested in keyboard combinations that cause a shift from graphics to business mode (upper and lower case letters) and the ones to cause the screen to scroll up (without having to cursor down to the bottom of the screen).

Send your contributions directly to me at Virginia Micro Systems, Inc., 14415 Jefferson Davis Highway, Woodbridge, VA 22191. ©

www.commodore.ca

DR. DALEY'S *BEST* Mailing List Is Now Better!

DR. DALEY has taken his best selling mailing list and made it even better! This version has been totally revised to increase the reliability of the files and make it even easier to operate. Several new features have been added:

- Goof-proof input routine. Eliminates the irritating results of accidentally pressing some cursor control keys. This is a machine code routine so it is as fast as you are! BONUS—Auto repeat on all keys!
- Interface to allow output of the entire mailing list or virtually *ANY* subset to WORDPRO III and WORDPRO IV format files so you can use these to generate personalized form letters. *YOU* can format the structure of this output!
- Routines to merge files and to minimize the number of duplicate entries in a file.
- More machine code routines to speed up processing.
- In addition you have the same powerful file formatting options where *YOU* can determine the structure of the files. *YOU* can format your label output with up to 11 lines per label and from 1 to 8 (yes EIGHT) labels per line.

This system is completely menu driven. It includes 100 pages of user documentation. This documentation is for the end user and is not padded with listings, flow charts, and other such extraneous material.

This program will be available for a short time at the introductory price of $159.95. It is available for the 32K PET and CBM 2000, 3000 and 8000 series computers. You can order through your dealer or directly from us. We will accept VISA or MASTERCARD or your check or money order. Overseas orders include 10% to cover shipping.

Charge to your MC/VISA

master charge THE INTERBANK CARD

VISA

DR. DALEY'S Software

425 Grove Avenue, Berrien Springs, MI 49103
Phone (616) 471-5514
Sunday - Thursday noon to 9 p.m., Eastern Time

www.commodore.ca

Keyprint

Charles Brannon

KEYPRINT is an easy solution to many hardcopy problems. For example, how would you copy the instructions from a computer game onto your printer? The obvious solution is to modify the program to direct its output to the printer. This is, however, time-consuming. Besides, what if -- horror of horrors -- you do not know how to make this modification?

So what does KEYPRINT do, anyway? Simple. You just touch a single key and the entire screen is copied onto the printer. This can happen at any time: while calculations are in progress, during a game of STARTREK, after a print-out of information to the screen, when you touch that certain key accidentally -- *anytime.* KEYPRINT totally interrupts everything PET is doing, dumps the screen onto the printer, and then returns control back to BASIC as though nothing had happened.

KEYPRINT's uses are multitudinous. No longer do you have to write special printer subroutines. It's just touch and go. Your software can even call the screen dump directly with an SYS command. If you have a Commodore 2022 printer, you can copy graphics verbatim. (Remember to set the lines-per-inch to eight first. A side-effect of this is that text looks crammed together; remember to reset the 1pi to six.)

So here's how to use KEYPRINT:

1. Enter the machine language monitor with an SYS 1024 command

2. If you've already typed in and saved KEYPRINT, enter:

   .L "KEYPRINT",01

   and hit 'RETURN'. Now type an 'X', hit 'RETURN' and go to step 6.

3. Otherwise, list the block of memory that KEYPRINT occupies with:

   .M 033A 03CB

4. Now, using the cursor, replace the "numbers" (Which often contain alphabetic characters, since they're hexadecimal) with the one shown in the listing. Type these bytes in EXACTLY as shown. (All machine language program instructions seem to stress that but it's really important as the program will CRASH if you don't type it in perfectly right.) Remember to hit 'RETURN' after each line.

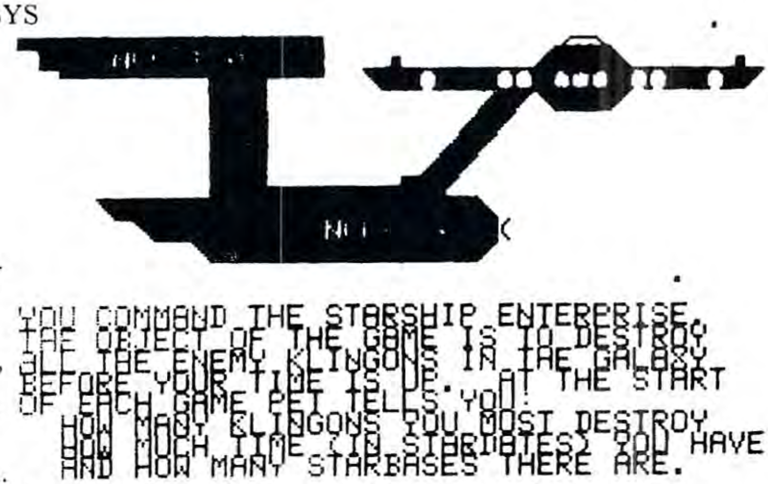
5. Save the program by entering:

   .S "KEYPRINT",01,033A,03CB (Afterwards, enter .X to exit to BASIC)

6. Now activate KEYPRINT with:

   SYS 826 (hit 'RETURN')

```
B*
     PC  IRQ  SR AC XR YR SP
.;  0401 0345 32 04 5E 00 F4
.M 033A 03CB
.:  033A 78 A9 03 85 91 A9 45 85
.:  0342 90 58 60 A5 97 C9 45 D0
.:  034A 03 20 51 03 4C 2E E6 A9
.:  0352 80 85 20 A9 00 85 1F A9
.:  035A 04 85 B0 85 D4 20 BA F0
.:  0362 20 2D F1 A9 19 85 22 A9
.:  036A 0D 85 21 20 D2 FF A9 11
.:  0372 AE 4C E8 E0 0C D0 02 A9
.:  037A 91 20 D2 FF A0 00 B1 1F
.:  0382 29 7F AA B1 1F 45 21 10
.:  038A 0B B1 1F 85 21 29 80 49
.:  0392 92 20 D2 FF 8A C9 20 B0
.:  039A 04 09 40 D0 0E C9 40 90
.:  03A2 0A C9 60 B0 04 09 80 D0
.:  03AA 02 49 C0 20 D2 FF C8 C0
.:  03B2 28 90 CB A5 1F 69 27 85
.:  03BA 1F 90 02 E6 20 C6 22 D0
.:  03C2 A6 A9 0D 20 D2 FF 4C CC
.:  03CA FF 72 21 61 3F 7F 76 57
.X
```



The cursor should re-appear almost instantly, blinking merrily under 'READY.'' If it doesn't, then your PET has CRASHED. Why? Either you typed in the program incorrectly, (even one tiny mistake) or you're using an old ROM PET. Shame on you! Go back and check over that program you typed in! (Aren't you glad you saved it first?)

www.commodore.ca

PET SOFTWARE

LAS VEGAS CASINO SERIES:

These four programs were developed both as a tutorial for those planning to visit a Casino and wanting to learn to play correctly, and as a means for the serious gambler to develop and thoroughly test a gambling 'system' under actual Casino conditions. All betting odds and options available in the Casino of the MGM Grand Hotel in Las Vegas have been incorporated into these programs. Full screen graphics have been used to show the cards being dealt, the Roulette Wheel spinning, and the Dice being thrown in order to increase realism and heighten enjoyment.

1. **Casino Blackjack:**
For 1 to 5 players plus the dealer. Keeps track of winnings or losses for each player plus number of hands won, lost, and tied and the number of times the dealer and each player has busted. Play with 1, 2, or 4 decks. Change decks or reshuffle at any time. Split pairs, Double Down, or place an Insurance bet. Full Casino rules and many other options .... $ 7.95

2. **Casino Roulette:**
Bet on one number, two numbers, odd, even, black, etc. Watch the numbers come up as the wheel spins. Twelve ways to bet.......... $ 7.95

3. **Casino Baccarat:**
James Bond's favorite game. Two games in on. Casino style and Blackjack style. Includes special features to help in developing a winning 'system' .... $ 7.95

4. **Casino Craps:**
Bet the Dice to Pass or Fall Off. Bet the Hardway or Press with Double Odds. Ten ways to bet.......... $ 7.95

5. **Casino Package:**
All four Casino Programs above $24.95

STRATEGIC GAMES:

6. **Backgammon:**
Play Backgammon against your PET. Excellent graphics and doubling option make for a fast and exciting game .... $ 9.95

7. **SP(*) INTRUDERS: (WITH SOUND)**
Written by COMMODORE-JAPAN. Performs exactly like the popular video arcade version being played all over the country. Machine language graphics and sound provide hours of entertainment.......... $ 9.95
Please specify which ROM set your PET has.

8. **Checkers:**
Play against the PET. PET plays a good, fast game according to International Checker Rules. Excellent graphics show the board and all checkers. Watch your PET move his man around the board. Clock shows elapsed time for each move .... $ 7.95

9. **Qubic-4:**
This is three dimensional Tic-Tac-Toe played on four, 4 x 4 boards. PET plays a fast exciting game choosing one of three strategies for each game.......... $ 7.95

10. **Go Moku:**
Ancient Chinese board game played on a 9 x 9 board. Get 5 men in a row before your PET. A different strategy for each game.......... $ 7.95

11. **Othello:**
English game known as Reversi. Try to capture the PET's men before he captures yours. Play against the PET or against your friends. Fast and fun.......... $ 7.95

12. **Cribbage:**
An Excellent version of this favorite card game. All cards are shown using PET's excellent graphics. The PET plays a cool logical game difficult to beat even for the best players .... $ 7.95

13. **Game Package:**
Any six of the above programs.......... $34.95

GUARANTEE: All programs are guaranteed to be free from errors and to load on any PET. Any defective tapes will be replaced free of charge.

master charge THE INTERBANK CARD

Order From: **CMS SOFTWARE,** 5115 Menefee Drive, Dallas, TX 75277

VISA®

www.commodore.ca

7. Hopefully, your cursor came back. That means that KEYPRINT is ready and rarin' to go. How do you make it work? Just press the "/" key. If you have a printer hooked up that responds to a secondary address of 4, then the entire screen will be printed onto your printer. For devices other than 4, POKE 858, SA where SA equals the secondary address of your printer. (If the abofe terminology seems confusing, don't worry. If you have a Commodore printer, everything will work fine. If not, then I can't guarantee flawless operation.)

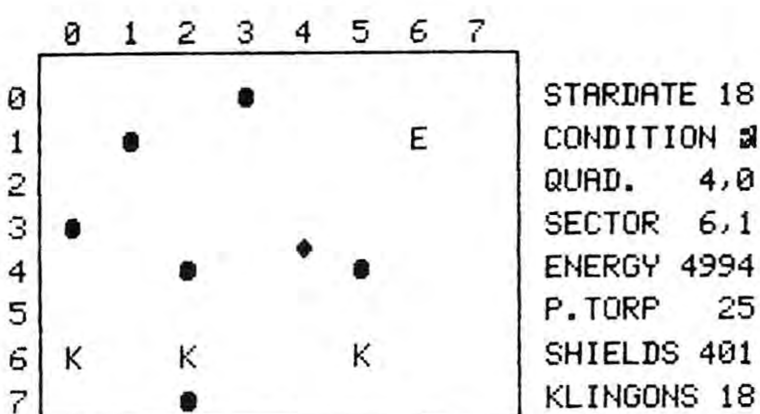
8. KEYPRINT remains in your machine until you turn it off or you otherwise interrupt its power supply (Like dropping the PET or setting it on fire). KEYPRINT can be de-activated, however, by a simple procedure: Hold down the shift key and press the RUN/STOP key or type in LOAD and hit 'RETURN'. Ignore any messages the PET says. Now press the RUN/STOP key again. The word BREAK is displayed and that is exactly what you did to KEYPRINT -- you broke it. It will work no longer. IMPORTANT NOTE: loading any program also "breaks" KEYPRINT. In either case, you can re-activate it with an SYS 826.

9. If you don't want to have to type a key to dump the screen, use a SYS 849 either in direct mode or within a program. It does not matter whether KEYPRINT is "activated" or not for the command to work.

So there you have it. I plan to use KEYPRINT quite a bit in the future. I think of it as a "Wedge" for the printer as DOS SUPPORT (Commodore) is for the 2040.

P.S. I want to make it clear which key is used to print the screen: it is the key to the right of the ampersand at the top of the keyboard, not the shift of "M". Also, beware that some programs use the second-cassette buffer (where KEYPRINT resides) for data storage or for their own machine language programs.

Resources:
Butterfield, Jim. "PET in Transition" **COMPUTE**, pp 68-70 (Fall, 1979)
Sheward, D. "Listing from Commodore's 'The Transactor'" **The PAPER**, p.39 (March/April 1980)



©

BLACKJACK ANALYZER

PUT PET TO WORK

PLAYS 100,000 games in 24 minutes.
Find the most advantageous method.

User defines player method by simple keyboard entry including when to hit and double on soft and hard hands, split pairs and take insurance. Then run simulator and see resulting gain or loss in 24 minutes run time.

Uses 4 Decks. Full casino rules. Countless permutations.
- OR -
Switch to visual play and watch your system in real time.
8K. Specify old or new ROM. 6502 ML
Cassette Tape $12.00

COUNTING TUTOR

Play Blackjack and win by keeping track of high and low cards. Program displays count on request if you lose track. A real time simulator.
8K. Specify old or new ROM. 6502 ML
Cassette tape $6.00

Malco
54 Hesketh Court
Toronto, CANADA
M4A 1M6

Oops! Our address was misspelled in the last issue. If your letter was returned, please try again.

commodore

Your Commodore Dealer in CANADA

We stock the full line of
PET and CBM products and accessories.

OFF THE SHELF DELIVERY

SOFTWARE — HARDWARE — SERVICE
Best Book Selection In Town
BEST BOOK SELECTION IN TOWN

Dust Covers: Pet Cover $11.95
Floppy Cover 9.95
Printer Cover (old) 9.95
Printer Cover (new) 9.95

Trade-in Your Old PET For The Latests Models

HOUSE oF COMPUTERS inc.
368 EGLINTON AVE. W. (at Avenue Rd.)
TORONTO, ONTARIO M5N 1A2
(416) 482-4336

www.commodore.ca

STOP!

WE WERE PROVEN WRONG BY OVER 50% OF OUR CUSTOMERS!

READ ON TO SEE WHY.

MISCALCULATION?

We predicted that our recent ad on the PLEXI-VUE Solar Screen would sell a lot of screens. It did! We also predicted that giving a FREE PLEXI-VUE Screen with our SOFTPAC-1 package would not find many takers, due to the fact that there wasn't a full description of it. Over 50% of our customers PROVED US WRONG! We also figured that allowing a FREE TRIAL would find us giving a few refunds on the Screens and software packages. The customers again PROVED US WRONG! We have not had 1 return! Our calculations may have been wrong, but in one thing we are right. And that is when we say, "IF THE QUANTITY DOESN'T GET YOU, THE QUALITY WILL!" Since so many have found this to be true, we are going to make the following offer for CHRISTMAS. Purchase a SOFTPAC-1 package and you will receive FREE a PLEXI-VUE Solar Screen of your choice . . . our CHRISTMAS gift to you!

NEW CONCEPT?

We can't put a price on our most valuable asset, the customer. Thats why we offer the customer an attractive package, good programs, Back-up copies, & Listings of the serious programs. How many other companies in the world offer this? We DARE you to COMPARE our software with other Manufacturers. The names of some of our programs may be the same as some you've seen, but the programs are different. Most software from other sources require an entry and then a RETURN. Ours only needs one keystroke for most entries or choices from a Menu. You are only required to press RETURN when entering DATA. Games from other sources usually have set parameters. We offer programmability in our games so the user can change the level of difficulty, and therefore not tire of the game quickly. We have extensively re-written our programs to preclude the user from falling out of the program with an incorrect entry. You receive a sheet describing how to ADD SOUND for less than $15! Some of the programs have the SOUND CAPABILITY built-in. Our Floppy Disc version is menu-driven using a sophisticated overlay technique! One keystroke LOADS and RUNS the program of your choice from the Animated Menu! We give Back-up copies on a SEPARATE Disc Or Cassette! We have your order in the Mail in less than 4 days! By now you should be able to understand why we haven't been receiving any returns.

PROGRAMS

If you like chess or checkers you will enjoy OTHELLO, which pits you against the Computer in a game of strategy. OTHELLO/2 allows you to play the same game with a friend, while the Computer calls you each by name, and constantly updates scores. You can checkout your BIORHYTHMS, and see how accurate the predictions are. In ELECTRICITY you can check the cost of operating an Appliance, Daily, Monthly or Yearly. It uses two figures from your Electric bill to give an accurate picture of actual costs. With the LOANS programs you can figure the cost of any purchase through a Credit Union or Bank. SAVINGS allows you to figure what interest will accumulate over a period of time using the variables you input. STRATEGY is virtually a programmable MASTERMIND, because you can control the level of difficulty. The QUIZ MASTER Demo tests your ability of recollection of States & Capitols. You can use it as a Multiple Choice or Fill-in quiz! The QUIZ MASTER PGM is blank and allows you to create quizzes for your own kids. The MPG program allows you to enter your odometer reading and gallons purchased to give you the miles per gallon at each fill-up or average over the long run. The MEMORY TEST gives a choice of tests. You can test (1) All Contiguous memory (2) From any Location you select (3) Screen Memory or (4) the Error Message. The Arcade type games some of which have sound, allow you and your family to enjoy them together. In CHASE you are trying to avoid being captured by Robots, while attempting to keep from running into the electrified posts or fence. You can change level of difficulty. In TANK a two player game, you try to destroy your opponent's TANKS at the same time he is trying to destroy yours! The difference is both players can play at the same time, and neither player can hold a key down to stop the other from moving. In TANKS-16k you have a choice of progressively harder playing fields. BOUNCE is a game of deflection. You must deflect a moving ball into targets. The computer keeps track of the time, and gives humorous remarks at the end of each game. You can change number of targets. BREAKOUT is the old favorite where you have a paddle which you use to deflect a ball into a wall of bricks. Each hit removes a brick, and you win when you have removed all the bricks. You have a limited number of balls to use! Game will run in an automatic mode for display. It isn't easy to win! As you can see there is something for the whole family, and with the price of gasoline on the rise, use your computer to entertain your family and friends.

FREE TRIAL OFFER

We urge you to try the SOFTPAC-1 Package now. Order one for our 30-day no obligation trial. See how much you enjoy the programs. See the difference the FREE PLEXI-VUE Solar Screen makes to your computer. We can make this offer because we have a QUALITY PRODUCT at a REASONABLE PRICE that we feel will meet with your approval, if you will GIVE IT A TRY!

Order a SOFTPAC-1 at the Reg. price of $34.95 + $2.00 Shipping and we'll GIVE you the PLEXI-VUE FREE! THE 30 DAY TRIAL OFFER APPLIES TO BOTH! (You must return BOTH for a refund)

SOFTPAC-1 contains 18 PGMS on DISC or 17 PGMS on TAPE (SPECIFY) in a Notebook with BACK-UP copies, printed instructions and program Listings! There are GAMES W/SOUND too!

Give the PLEXI-VUE Model Number you need, from the CHART:

PX1 = PETs/CBMs with METAL CRT Cases.
PX2 = PETs/CBMs with PLASTIC CRT Cases.
PX3 = NEW 80 Character CRT CBMs.

★ ★ CHRISTMAS BONUS! ★ ★

On all orders POSTMARKED before DECEMBER 16, 1980, we will include a Certificate that allows you to purchase a 1 YEAR Subscription to COMPUTE at 25% off the Regular Price of $16! Thats a Savings of $6 off the Newsstand Price. This offer is in addition to the FREE PLEXI-VUE if you meet the above postmark requirement.

IF FOR ANY REASON YOU ARE NOT SATISFIED WITH THE SOFTPAC-1 PACKAGE, RETURN IT WITH THE SCREEN AND ABOVE CERTIFICATE FOR A PROMPT AND COURTEOUS REFUND. YOU CAN'T LOSE!

We accept personal checks, money orders and MASTERCHARGE or VISA! Give ACCOUNT NUMBER, INTERBANK NUMBER, EXPIRATION DATE, AND SIGN your Order.

COMPETITIVE

SOFTWARE

21650 Maple Glen Drive
Edwardsburg, MI 49112

www.commodore.ca

PET 4.0 ROM Routines

Jim Butterfield, Toronto

The 40-character and 80-character machines are the same except for addresses $E000-$E7FF.

This map shows where various routines lie. The first address is not necessarily the proper entry point for the routine. Similarly, many routines require register setup or data preparation before calling.

	Description
B000-B065	Action addresses for primary keywords
B066-B093	Action addresses for functions
B094-B0B1	Hierarchy and action addresses for operators
B0B2-B20C	Table of Basic keywords
B20D-B321	Basic messages, mostly error messages
B322-B34F	Search the stack for FOR or GOSUB activity
B350-B392	Open up space in memory
B393-B39F	Test: stack too deep?
B3A0-B3CC	Check available memory
B3CD	Send canned error message, then:
B3FF-B41E	Warm start; wait for Basic command
B41F-B4B5	Handle new Basic line input
B4B6-B4E1	Rebuild chaining of Basic lines
B4E2-B4FA	Receive line from keyboard
B4FB-B5A2	Crunch keywords into Basic tokens
B5A3-B5D1	Search Basic for given line number
B5D2	Perform NEW, and;
B5EC-B621	Perform CLR
B622-B62F	Reset Basic execution to start
B630-B6DD	Perform LIST
B6DE-B784	Perform FOR
B785-B7B6	Execute Basic statement
B7B7-B7C5	Perform RESTORE
B7C6-B7ED	Perform STOP or END
B7EE-B807	Perform CONT
B808-B812	Perform RUN
B813-B82F	Perform GOSUB
B830-B85C	Perform GOTO
B85D	Perform RETURN, then:
B883-B890	Perform DATA: skip statement
B891	Scan for next Basic statement
B894-B8B2	Scan for next Basic line
B8B3	Perform IF, and perhaps:
B8C6-B8D5	Perform REM: skip line
B8D6-B8F5	Perform ON
B8F6-B92F	Accept fixed-point number
B930-BA87	Perform LET
BA88-BA8D	Perform PRINT#
BA8E-BAA1	Perform CMD
BAA2-BB1C	Perform PRINT
BB1D-BB39	Print string from memory
BB3A-BB4B	Print single format character
BB4C-BB79	Handle bad input data
BB7A-BBA3	Perform GET
BBA4-BBBD	Perform INPUT#
BBBE-BBF4	Perform INPUT
BBF5-BC01	Prompt and receive input
BC02-BCF6	Perform READ
BCF7-BD18	Canned Input error messages
BD19-BD71	Perform NEXT
BD72-BD97	Check type mismatch
BD98	Evaluate expression
BEE9	Evaluate expression within parentheses
BEEF	Check parenthesis, comma
BF00-BF0B	Syntax error exit
BF8C-C046	Variable name setup
C047-C085	Set up function references
C086-C0B5	Perform OR, AND
C0B6-C11D	Perform comparisons
C11E-C12A	Perform DIM
C12B-C1BF	Search for variable
C1C0-C2C7	Create new variable
C2C8-C2D8	Setup array pointer
C2D9-C2DC	32768 in floating binary
C2DD-C2FB	Evaluate integer expression
C2FC-C4A7	Find or make array
C4A8	Perform FRE, and:
C4BC-C4C8	Convert fixed-to-floating
C4C9-C4CE	Perform POS
C4CF-C4DB	Check not Direct
C4DC-C509	Perform DEF
C50A-C51C	Check FNx syntax
C51D-C58D	Evaluate FNx
C58E-C59D	Perform STR$
C59E-C5AF	Do string vector
C5B0-C61C	Scan, set up string
C61D-C669	Allocate space for string
C66A-C74E	Garbage collection
C74F-C78B	Concatenate
C78C-C7B4	Store string
C7B5-C810	Discard unwanted string
C811-C821	Clean descriptor stack
C822-C835	Perform CHR$
C836-C861	Perform LEFT$
C862-C86C	Perform RIGHT$
C86D-C896	Perform MID$
C897-C8B1	Pull string data
C8B2-C8B7	Perform LEN
C8B8-C8C0	Switch string to numeric
C8C1-C8D0	Perform ASC
C8D1-C8E2	Get byte parameter
C8E3-C920	Perform VAL
C921-C92C	Get two parameters for POKE or WAIT
C92D-C942	Convert floating-to-fixed
C943-C959	Perform PEEK
C95A-C962	Perform POKE
C963-C97E	Perform WAIT
C97F-C985	Add 0.5
C986	Perform subtraction
C998-CA7C	Perform addition
CA7D-CAB3	Complement accum#1
CAB4-CAB8	Overflow exit
CAB9-CAF1	Multiply-a-byte
CAF2-CB1F	Constants
CB20	Perform LOG
CB5E-CBC1	Perform multiplication
CBC2-CBEC	Unpack memory into accum#2
CBED-CC09	Test & adjust accumulators
CC0A-CC17	Handle overflow and underflow
CC18-CC2E	Multiply by 10
CC2F-CC33	10 in floating binary
CC34	Divide by 10
CC3D	Perform divide-by
CC45-CCD7	Perform divide-into
CCD8-CCFC	Unpack memory into accum#1
CCFD-CD31	Pack accum#1 into memory
CD32-CD41	Move accum#2 to #1
CD42-CD50	Move accum#1 to #2
CD51-CD60	Round accum#1
CD61-CD6E	Get accum#1 sign
CD6F-CD8D	Perform SGN
CD8E-CD90	Perform ABS
CD91-CDD0	Compare accum#1 to memory
CDD1-CE01	Floating-to-fixed
CE02-CE28	Perform INT
CE29-CEB3	Convert string to floating-point
CEB4-CEE8	Get new ASCII digit
CEE9-CEF8	Constants
CF78	Print IN, then:
CF7F-CF92	Print Basic line #
CF93-D0C6	Convert floating-point to ASCII
D0C7-D107	Constants
D108	Perform SQR
D112	Perform power function
D14B-D155	Perform negation
D156-D183	Constants
D184-D1D6	Perform EXP
D1D7-D220	Series evaluation

www.commodore.ca

Skyles Electric Works

BASIC Programmer's, Toolkit™, Disk-O-Pro™, Command-O™

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

Here are thirty-five commands you'll need, all on dual chips you can install in two minutes without tools, **on any PET or PET system.** 2KB or 4KB of ROM firmware on each chip with a collection of machine language programs available from the time you turn on your PET to the time you shut it off. No tape to load or to interfere with any running programs.

For PET/CBM 2001-8, -8N, -16N/B, -32N/B, 3016 and 3032

BASIC Programmers Toolkit™ commands

AUTO^{ed} DELETE^{ed} RENUMBER^{ed} HELP^{ed} TRACE^{ed} STEP^{ed} OFF^{ed} APPEND^{ed} DUMP^{ed} FIND^{ed}

BASIC Programmers Disk-O-Pro™

CONCAT^{B80} DOPEN^{B80} DCLOSE^{B80} RECORD^{B80} HEADER^{B80} COLLECT^{B80} BACKUP^{B80} COPY^{B80} APPEND^{B80} DSAVE^{B80} DLOAD^{B80} CATALOG^{B80} rename^{B80} SCRATCH^{B80} DIRECTORY^{B80} INITIALIZE^{BS} MERGE^{BS} EXECUTE^{BS} SCROLL^{ed} OUT^{ed} SET^{ed} KILL^{ed} EAT^{ed} PRINT USING^{BS} SEND^{BS} BEEP^{BS}

For CBM 8016 and 8032

BASIC Programmer Command-O™

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

ed — a program editing and debugging command
B80 — a BASIC command also available on Commodore CBM™ 8016 and 8032 computers.
BS — a Skyles Electric Works added value BASIC command.
BASIC Programmers Toolkit™ is a trademark of Palo Alto IC's.
BASIC Programmers Disk-O-Pro™, Command-O™ is a trademark of Skyles Electric Works.
PET™, CBM™ are trademarks of Commodore Business Machines.

Can be placed in main board socket or with precision-engineered interface

Phone or write for information. We'll be delighted to answer any questions and to send you the complete information package.

Skyles Electric Works

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

www.commodore.ca

```
D221-D228 RND constants
D229-D281 Perform RND
D282      Perform COS
D289-D2D1 Perform SIN
D2D2-D2FD Perform TAN
D2FE-D32B Constants
D32C-D35B Perform ATN
D35C-D398 Constants
D399-D3B5 CHRGET sub for zero page
D3B6-D471 Basic cold start
D472-D716 Machine Language Monitor
D717-D7AB MLM subroutines
D7AC-D802 Perform RECORD
D803-D837 Disk parameter checks
D838-D872 Dummy disk control messages
D873-D919 Perform CATALOG or DIRECTORY
D91A-D92E Output
D92F-D941 Find spare secondary address
D942-D976 Perform DOPEN
D977-D990 Perform APPEND
D991-D9D1 Get disk status
D9D2-DA06 Perform HEADER
DA07-DA30 Perform DCLOSE
DA31-DA64 Set up disk record
DA65-DA7D Perform COLLECT
DA7E-DAA6 Perform BACKUP
DAA7-DAC6 Perform COPY
DAC7-DAD3 Perform CONCAT
DAD4-DB0C Insert command string values
DB0D-DB39 Perform DSAVE
DB3A-DB65 Perform DLOAD
DB66-DB98 Perform SCRATCH
DB99-DB9D Check Direct command
DB9E-DBD6 Query ARE YOU SURE?
DBD7-DBE0 Print BAD DISK
DBE1-DBF9 Clear DS$ and ST
DBFA-DC67 Assemble disk command string
DC68-DE29 Parse Basic DOS command
DE2C-DE48 Get Device number
DE49-DE86 Get file name
DE87-DE9C Get small variable parameter
** Entry points only for E000-E7FF **
E000      Register/screen initialization
E0A7      Input from keyboard
E116      Input from screen
E202      Output character
E442      Main Interrupt entry
E455      Interrupt: clock, cursor, keyboard
E600      Exit from Interrupt
**                                  **
F000-F0D1 File messages
F0D2      Send 'Talk'
F0D5      Send 'Listen'
F0D7      Send IEEE command character
F109-F142 Send byte to IEEE
F143-F150 Send byte and clear ATN
F151-F16B Option: timeout or wait
F16C-F16F DEVICE NOT PRESENT
F170-F184 Timeout on read, clear control lines
F185-F192 Send canned file message
F193-F19D Send byte, clear control lines
F19E-F1AD Send normal (deferred) IEEE char
F1AE-F1BF Drop IEEE device
F1C0-F204 Input byte from IEEE
F205-F214 GET a byte
F215-F265 INPUT a byte
F266-F2A1 Output a byte
F2A2      Abort files
F2A6-F2C0 Restore default I/O devices
F2C1-F2DC Find/setup file data
F2DD-F334 Perform CLOSE
F335-F342 Test STOP key
F343-F348 Action STOP key
F349-F350 Send message if Direct mode
F351-F355 Test if Direct mode
F356-F400 Program load subroutine
F401-F448 Perform LOAD
F449-F46C Print SEARCHING
F46D-F47C Print LOADING or VERIFYING
F47D-F4A4 Get Load/Save parameters
F4A5-F4D2 Send name to IEEE
F4D3-F4F5 Find specific tape header
F4F6-F50C Perform VERIFY
F50D-F55F Get Open/Close parameters
F560-F5E4 Perform OPEN
F5E5-F618 Find any tape header
F619-F67A Write tape header
F67B-F694 Get start/end addrs from header
F695-F6AA Set buffer address
F6AB-F6C2 Set buffer start & end addrs
F6C3-F6CB Perform SYS
F6CC-F6DC Set tape write start & end
F6DD-F767 Perform SAVE
F768-F7AE Update clock
F7AF-F7FD Connect input device
F7FE-F84A Connect output device
F84B-F856 Bump tape buffer pointer
F857-F879 Wait for PLAY
F87A-F88B Test cassette switch
F88C-F899 Wait for RECORD
F89A      Initiate tape read
F8CB      Initiate tape write
F8E0-F92A Common tape I/O
F92B-F934 Test I/O complete
F935-F944 Test STOP key
F945-F975 Tape bit timing adjust
F976-FA9B Read tape bits
FA9C-FBBA Read tape characters
FBBB-FBC3 Reset tape read address
FBC4-FBC8 Flag error into ST
FBC9-FBD7 Reset counters for new byte
FBD8-FBF3 Write a bit to tape
FBF4-FC85 Tape write
FC86-FCBF Write tape leader
FCC0-FCDA Terminate tape; restore interrupt
FCDB-FCEA Set interrupt vector
FCEB-FCF8 Turn off tape motor
FCF9-FD0A Checksum calculation
FD0B-FD15 Advance load/save pointer
FD16-FD4B Power-on Reset
FD4C-FD5C Table of interrupt vectors
** Jump table:                   **
FF93-FF9E CONCAT,DOPEN,DCLOSE,RECORD
FF9F-FFAA HEADER,COLLECT,BACKUP,COPY
FFAB-FFB6 APPEND,DSAVE,DLOAD,CATALOG
FFB7-FFBC RENAME,SCRATCH
FFBD      Get disk status
FFC0      OPEN
FFC3      CLOSE
FFC6      Set input device
FFC9      Set output device
FFCC      Restore default I/O devices
FFCF      INPUT a byte
FFD2      Output a byte
FFD5      LOAD
FFD8      SAVE
FFDB      VERIFY
FFDE      SYS
FFE1      Test stop key
FFE4      GET byte
FFE7      Abort all files
FFEA      Update clock
FFFA-FFFF Hard vectors:  NMI, Reset, INT
```

©

www.commodore.ca

McTerm

MADISON COMPUTER

A Telecommunication Package

With the McTerm package your Commodore computer can now become a very intelligent terminal. It allows you to easily participate in the electronic mail revolution. McTerm can communicate simultaneously with another PET or mainframe. You can transfer entire disk files to other disk drives; send or receive program, WordPro, sequential, or relative files. This package works on Commodore computers with 3.0 or 4.0 Basic, 1.0 or 2.0 DOS, 40 or 80 column computers, and 2040 or 8050 disk drives (requires the use of TTL Level RS-232 modems).

Complete with Program, ROM, Cable and Manual.
Price: $195.00

Features include:

-Baud selection from 75 to 1200.
-Full and half duplex, also local echo.
-Supports odd, even and mark parity.
-CRC, error checking for PET to PET files.

1825 MONROE STREET, MADISON, WI 53711 (608) 255-5552
-Dealer Inquiries Are Welcome-

FINALLY!

PET DISK UTILITY PACK

MASSAGE ($60): EDIT BLOCKS, UPDATE BAM, SCAN FILES, MAP DISK, TRACE LINKS
EDIT ($60)1: EDIT AND APPEND SEQUENTIAL FILES IN-PLACE!
RECOVER ($30): RECOVER ACCIDENTALLY SCRATCHED FILES
CLEAN - FAST - FULLY DOCUMENTED - HARD COPY OPTIONS
COMPLETE UTILITY PACK - $100
LEDGER ($100): 8-24 COLUMNS, 9-39 BYTES/COLUMN, 3000 ROWS.
NUMBER - ALPHA - DATE - DOLLAR COLUMN SPECIFICATION.
NUMBER/DOLLAR COLUMNS DERIVEABLE VIA USER FUNCTIONS.

NOTEBOOK ($25): 1-5K OF NOTEBOOK SPACE CO-EXISTENT WITH BASIC.
RECALL - EDIT - TRANSFER WITH TWO KEYSTROKES.

CALL OR WRITE FOR BROCHURE
DEALER INQUIRIES INVITED

CALIFORNIA SOFTWARE ASSOCIATES
P.O. Box 969
Laguna Beach, CA. 92652
(714) 497-2000

www.commodore.ca

BASIC 4.0 MEMORY MAP

Compiled by
Jim Butterfield

There are some differences in usage between the 40- and 80-column machines.

Hex	Decimal	Description
0000-0002	0-2	USR jump
0003	3	Search character
0004	4	Scan-between-quotes flag
0005	5	Input buffer pointer; # of subscripts
0006	6	Default DIM flag
0007	7	Type: FF=string, 00=numeric
0008	8	Type: 80=integer, 00=floating point
0009	9	Flag: DATA scan; LIST quote; memory
000A	10	Subscript flag; FNX flag
000B	11	0=INPUT; $40=GET; $98=READ
000C	12	ATN sign/Comparison Evaluation flag
000D-000F	13-15	Disk status DS$ descriptor
0010	16	Current I/O device for prompt-suppress
0011-0012	17-18	Integer value (for SYS, GOTO etc)
0013-0015	19-21	Pointers for descriptor stack
0016-001E	22-30	Descriptor stack(temp strings)
001F-0022	31-34	Utility pointer area
0023-0027	35-39	Product area for multiplication
0028-0029	40-41	Pointer: Start-of-Basic
002A-002B	42-43	Pointer: Start-of-Variables
002C-002D	44-45	Pointer: Start-of-Arrays
002E-002F	46-47	Pointer: End-of-Arrays
0030-0031	48-49	Pointer: String-storage(moving down)
0032-0033	50-51	Utility string pointer
0034-0035	52-53	Pointer: Limit-of-memory
00B3	179	Logical Address temporary save
00B4	180	Tape buffer character; MLM command
00B5	181	File name pointer; MLM flag, counter
00B7	183	Serial bit count
00B9	185	Cycle counter
00BA	186	Tape writer countdown
00BB-00BC	187-188	Tape buffer pointers, #1 and #2
00BD	189	Write leader count; read pass1/2
00BE	190	Write new byte; read error flag
00BF	191	Write start bit; read bit seq error
00C0-00C1	192-193	Error log pointers, pass1/2
00C2	194	0=Scan/1-15=Count/$40=Load/$80=End
00C3	195	Write leader length; read checksum
00C4-00C5	196-197	Pointer to screen line
00C6	198	Position of cursor on above line
00C7-00C8	199-200	Utility pointer: tape, scroll
00C9-00CA	201-202	Tape end addrs/End of current program
00CB-00CC	203-204	Tape timing constants
00CD	205	0=direct cursor, else programmed
00CE	206	Tape read timer 1 enabled
00CF	207	EOT received from tape
00D0	208	Read character error
00D1	209	# characters in file name
00D2	210	Current file logical address
00D3	211	Current file secondary addrs
00D4	212	Current file device number

Why Is CURSOR So Good?

Maybe it's because we've always had high standards. Beginning with our first issue in July, 1978, we've published some 100 programs for the Commodore PET in our first 20 issues, plus 20 animated graphic "Front Cover" programs. Each program has been extensively edited by Glen Fisher, our Editorial Director. The result is obvious: **Cursor** programs reflect professional standards. We're proud of every program we publish.

But there's something else, too.

It's imagination. Our subscribers continue to be delighted with the new, fresh programming ideas that **Cursor** provides. Some of the best graphic animations for the Pet have appeared in **Cursor.** Teachers love us! They use **Cursor** as an example of what can be done on a PET, with some skill and imagination.

Finally, there's service. Orders for single issues are almost always shipped within 24 hours. New subscriptions are processed within five working days. Should you get one of our rare defective tapes, just return it for an immediate replacement. And of course you can cancel your subscription at any time and we'll gladly refund all remaining issues.

Cursor: Quality. Imagination. Service.

For only $4.95 you can buy a sample issue and judge for yourself. Or send $27 for a six-issue subscription. You'll get six C-30 cassettes, each with five programs and a Front Cover ready to LOAD and RUN on your PET. With each issue you also get our **Cursor Notes**, a lively commentary on the industry, as well as documentation for the programs.

☐ Sample issue of **Cursor** — $4.95 (CA. Res. add 6% tax)
☐ 6 issues for $27.00 (U.S. & Canada)

Published By: The Code Works
**Box 550
Goleta, CA. 93017
Phone 805-967-0905**

www.commodore.ca

0036-0037	54-55	Current Basic line number
0038-0039	56-57	Previous Basic line number
003A-003B	58-59	Pointer: Basic statement for CONT
003C-003D	60-61	Current DATA line number
003E-003F	62-63	Current DATA address
0040-0041	64-65	Input vector
0042-0043	66-67	Current variable name
0044-0045	68-69	Current variable address
0046-0047	70-71	Variable pointer for FOR/NEXT
0048-0049	72-73	Y-save; op-save; Basic pointer save
004A	74	Comparison symbol accumulator
004B-0050	75-80	Misc work area, pointers, etc
0051-0053	81-83	Jump vector for functions
0054-005D	84-93	Misc numeric work area
005E	94	Accum#1: Exponent
005F-0062	95-98	Accum#1: Mantissa
0063	99	Accum#1: Sign
0064	100	Series evaluation constant pointer
0065	101	Accum#1 hi-order (overflow)
0066-006B	102-107	Accum#2: Exponent, etc.
006C	108	Sign comparison, Acc#1 vs #2
006D	106	Accum#1 lo-order (rounding)
006E-006F	110-111	Cassette buff len/Series pointer
0070-0087	112-135	CHRGET subroutine; get Basic char
0077-0078	119-120	Basic pointer (within subrtn)
0088-008C	136-140	Random number seed.
008D-008F	141-143	Jiffy clock for TI and TI$
0090-0091	144-145	Hardware interrupt vector
0092-0093	146-147	BRK interrupt vector
0094-0095	148-149	NMI interrupt vector
0096	150	Status word ST
0097	151	Which key down; 255=no key
0098	152	Shift key: 1 if depressed
0099-009A	153-154	Correction clock
009B	155	Keyswitch PIA: STOP and RVS flags
009C	156	Timing constant for tape
009D	157	Load=0, Verify=1
009E	158	Number of characters in keybd buffer
009F	159	Screen reverse flag
00A0	160	IEEE output; 255=character pending
00A1	161	End-of-line-for-input pointer
00A3-00A4	163-164	Cursor log (row, column)
00A5	165	IEEE output buffer
00A6	166	Key image
00A7	167	0=flash cursor
00A8	168	Cursor timing countdown
00A9	169	Character under cursor
00AA	170	Cursor in blink phase
00AB	171	EOT received from tape
00AC	172	Input from screen/from keyboard
00AD	173	X save
00AE	174	How many open files
00AF	175	Input device, normally 0
00B0	176	Output CMD device, normally 3
00B1	177	Tape character parity
00B2	178	Byte received flag
00D5	213	Right-hand window or line margin
00D6-00D7	214-215	Pointer: Start of tape buffer
00D8	216	Line where cursor lives
00D9	217	Last key/checksum/misc.
00DA-00DB	218-219	File name pointer
00DC	220	Number of INSERTs outstanding
00DD	221	Write shift word/read character in
00DE	222	Tape blocks remaining to write/read
00DF	223	Serial word buffer
00E0-00F8	224-248	(40-column) Screen line wrap table
00E0-00E1	224-225	(80-column) Top, bottom of window
00E2	226	(80-column) Left window margin
00E3	227	(80-column) Limit of keybd buffer
00E4	228	(80-column) Key repeat flag
00E5	229	(80-column) Repeat countdown
00E6	230	(80-column) New key marker
00E7	231	(80-column) Chime time
00E8	232	(80-column) HOME count
00E9-00EA	233-234	(80-column) Input vector
00EB-00EC	235-236	(80-column) Output vector
00F9-00FA	249-250	Cassette status, #1 and #2
00FB-00FC	251-252	MLM pointer/Tape start address
00FD-00FE	253-254	MLM, DOS pointer, misc.
0100-010A	256-266	STR$ work area, MLM work
0100-013E	256-318	Tape read error log
0100-01FF	256-511	Processor stack
0200-0250	512-592	MLM work area; Input buffer
0251-025A	593-602	File logical address table
025B-0264	603-612	File device number table
0265-026E	613-622	File secondary adds table
026F-0278	623-632	Keyboard input buffer
027A-0339	634-825	Tape#1 input buffer
033A-03F9	826-1017	Tape#2 input buffer
033A	826	DOS character pointer
033B	827	DOS drive 1 flag
033C	828	DOS drive 2 flag
033D	829	DOS length/write flag
033E	830	DOS syntax flags
033F-0340	831-832	DOS disk ID
0341	833	DOS command string count
0342-0352	834-850	DOS file name buffer
0353-0380	851-896	DOS command string buffer
03EE-03F7	1006-1015	(80-column) Tab stop table
03FA-03FB	1018-1019	Monitor extension vector
03FC	1020	IEEE timeout defeat
0400-7FFF	1024-32767	Available RAM including expansion
8000-83FF	32768-33791	(40-column) Video RAM
8000-87FF	32768-34815	(80-column) Video RAM
9000-AFFF	36864-45055	Available ROM expansion area
B000-DFFF	45056-57343	Basic, DOS, Machine Lang Monitor
E000-E7FF	57344-59391	Screen, Keyboard, Interrupt programs
E810-E813	59408-59411	PIA 1 - Keyboard I/O
E820-E823	59424-59427	PIA 2 - IEEE-488 I/O
E840-E84F	59456-59471	VIA - I/O and timers
E880-E881	59520-59521	(80-column) CRT Controller
F000-FFFF	61440-65535	Reset, I/O handlers, Tape routines ©

www.commodore.ca

Algebraic Expression Input for the Pet, Version 2

Elizabeth Deal

Issue 4 of Compute contains an article by Harvey Davis about inputing expressions during program execution. Mr. Davis posed two questions at the end of his article: 1. Can writing on the screen be avoided, and 2. Can you recover from syntax error without losing the variables? If I understand his questions correctly, then the answer to both questions is yes.

The following program achieves the same results by a different method. The screen is not cleared between calculations and variables are preserved in case of error in the expression, unless the expression itself mangles the variables.

The program begins with a funciton F(X) = X. The desired substitute functions are INPUT by the subroutine in lines 570-680. The function string is decoded into either keyword tokens or ASCII values of non-token characters (lines 630-660) which are then poked into the DEFFN line beginning in the first position after an equal sign. M1 = J + 7 in line 580 points to that position. When the control returns to the main program, the value of the function is calculated and displayed and the program is ready for new values, new function or both. Recovery procedure from error in the function is described in the REM lines at the end of listing.

The poking routine depends on the DEFFN line being the first physical line of the program. I advise you to make this line as long as shown in the listing, but at least fifteen colons(:) long. The poking subroutine checks the available room between the equal sign and the beginning of next program line. Any expression longer than permitted will be rejected. This prevents self destruction of the program following the DEFFN line.

The expressions are INPUT beginning with the desired text that will go after the equal sign. Pet abbreviations are not allowed. PEEK, SQR, INT, etc. must be written as full words. Spaces are permitted any place after the equal sign.

Please observe other restrictions described in the REM lines. Save the program before running it for the first time, for if any typographical errors will lead to poke addresses being incorrect this program will self destruct.

I'd like to recommend that people using my BIG FILES (issue 4) substitute this sort of a routine for the inconvenient RUN and GOTO procedures used in changing the decision lines 1140-1150.

References:

1. Token list by Warren Swan, **Pet User Notes**, vol 1, #3, p.5
2. Davis, Algebraic Input for the Pet, **Compute**, vol 1, #4
3. Commodore Manual


```
100 DEFFNF(X)=X:::::::::::::::::::::::::
      ¬::::::::::::::::::::::::::::::::::
      ¬::::
110 REM
120 REM ^ DEFFNF(X) LINE MUST BE THE
130 REM FIRST LINE IN THE PROGRAM.
140 REM
150 REM 1. RIGHT AFTER = SIGN TYPE
160 REM 64 COLONS OR X FOLLOWED BY 63
170 REM COLONS;  63-64 IS BEST. YOU
180 REM MAY PUT AS FEW AS 15 IF VERY
190 REM SHORT FUNCTIONS ARE USED.
200 REM 2. IF DURING EXECUTION DEFFN
210 REM LINE BECOMES 3 LINES LONG -
220 REM DON'T WORRY & LEAVE IT ALONE
230 REM 3. MORE TO READ AT THE END
240 REM
250 GOSUB480
260 PRINT:PRINT"rBfOTH NEW rSfAME F,
      ¬V rFfUNCTION rVfALUES rQfUIT":
      ¬PRINT
270 GETP$:AN=-1*(P$="B")-2*(P$="S")-3*(P
      ¬$="F")-4*(P$="V")-5*(P$="Q")
280 ONAN+1GOTO270,290,400,390,290,430
290 INPUT"X=  1←←←";X :    REM INSERT
300 INPUT"A=  1←←←";A :    REM EDIT
310 INPUT"B=  1←←←";B :    REM ROUTINE
320 INPUT"C=  1←←←";C :    REM HERE
330 REM
340 REM ^ VARIABLES DEFAULT TO 1 IF
350 REM ONLY RETURN KEY IS PRESSED
360 REM INPUT PROMPT CONTAINS
370 REM SP SP 1 (3)CURSOR-LEFT
380 REM
390 IFAN<>4THENGOSUB570
400 IFF$=""THENF$="RESULT"
410 PRINT"X="X"A="A"B="B"C="C
420 Y=FNF(X):PRINTF$"="Y:GOTO260
430 END
440 REM ============================
450 REM INITIALIZE LIST OF KEYWORDS
460 REM AND THEIR TOKEN NUMBERS
470 REM ============================
480 T=24:DIMT$(T),T(T):FORJ=1TOT:
      ¬READT$(J),T(J):NEXT:RETURN
490 DATA NOT,168,+,170,-,171,*,172,/,
      ¬173,AND,175,OR,176,>,177,=,178,<,
      ¬179
500 DATA INT,181,ABS,182,SQR,186,RND,
      ¬187,LOG,188,EXP,189,COS,190,SIN,
      ¬191
510 DATA TAN,192,ATN,193,PEEK,194,USR,
      ¬183,^,174,_,255
520 REM LAST 2 KEYW. ARE UP-ARROW & PI
530 REM ============================
540 REM INPUT FUNCTION, FIND TOKENS
550 REM POKE INTO DEFFN EXPRESSION
560 REM ============================
```

www.commodore.ca

Skyles Electric Works

"If you could own only one peripheral for your PET, it should be this. It opens the whole world to your PET."

The Cat: Switches for mode selection and operation. LEDs display unit status. Acoustic self-test is standard. Compact powerpack plugs directly into wall socket.

Now $325.00* complete with membership in The Source*, Skyles six-foot cable/interface to the PET user port, together with Skyles cassette program in machine language and in BASIC.
(If bought separately: $180.00, modem; $80.00, cable/interface and program; $100.00, membership in The Source)*

The Cat and D-Cat have been specially prepared by Skyles for interfacing to the PET user port (*not* to the IEEE port) and with a special cassette program, allowing communication

...from modem to disk and disk to modem
...from modem to terminal; read on screen, save on disk
...from disk to printer through IEEE
...from disk to screen

The D-Cat: FCC-approved for handset jack connection with any modular phone, either single or multi-line. No need for adapters. Can fit under phone; installs in seconds. Mode switch to monitor voice or data transmission. Special "hold" function; complete self-test. Power pack plugs directly into wall socket.

Now $350.00* complete with membership in The Source*, Skyles six-foot cable/interface to the user port and Skyles cassette program in machine language and in BASIC.

(If bought separately: $210.00, the modem; $80.00, cable/interface and program; $100.00, membership in The Source)*

*What About the Source?

Sometimes called "The Information Utility," it's a telecomputing network that gives you, through your Cat or D-Cat modem, thousands of programs and data bases and allows you to communicate with other users interactively and through electronic mail.

**California residents: please add 6% or 6.5% sales tax as required*

VISA, MASTERCHARGE ORDERS CALL (800) 538-3083 (except California residents)
CALIFORNIA ORDERS PLEASE CALL (408) 257-9140

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

www.commodore.ca

```
570 M2=PEEK(1025)+256*PEEK(1026)-2:
    ¬FORJ=1029TOM2
580 IFPEEK(J)=150ANDPEEK(J+1)=165ANDPEEK
    ¬(J+6)=178THENM1=J+7:GOTO600
590 NEXTJ:GOTO680
600 MM=M2-M1:IF(MM>63)OR(MM<15)GOTO680
610 INPUT"F(X)=  X←←←";F$:IFLEN(F$)>MM+1
    ¬THENPRINTTAB(15)"TOO LONG":GOTO610
620 FORJ=M1TOM2:POKEJ,58:NEXT:N=-1:L=1:
    ¬PRINT"↑"TAB(7)"ɼ";
630 FORK=1TOT:M$=MID$(F$,L,LEN(T$(K)))
640 IFM$=T$(K)THENN=N+1:L=L+LEN(M$):
    ¬QF=T(K):GOTO660
650 NEXTK:M$=MID$(F$,L,1):QF=ASC(M$):
    ¬L=L+1:N=N+1
660 PRINTM$;:POKEM1+N,QF:IFL<=LEN(F$)GOT
    ¬O630
670 PRINT"f̂":PRINT:RETURN
680 PRINT"THINGS AREN'T RIGHT. READ REM ¬
    ¬LINES":END
690 REM ============================
700 REM ALGEBRAIC INPUT - VERSION 2
710 REM       ELIZABETH DEAL
720 REM 337 W.FIRST AVE,MALVERN,PA
730 REM           19355
740 REM ============================
750 REM 4.DO NOT RUN WITHOUT SAVING !
760 REM   ERRORS IN POKING ROUTINE
770 REM   WILL WIPE OUT THE WHOLE
780 REM   PROGRAM
790 REM 5.MAKE SURE THAT THE POKE AD-
800 REM   DRESSES ARE TYPED CORRECTLY
810 REM 6.MAKE SURE THAT 'DEFFNF(X)='
820 REM   CONTAINS NO SPACES,AND THAT
830 REM   F AND X SUBSTITUTES (IF ANY)
840 REM   ARE SINGLE LETTER NAMES
850 REM 7.UNLESS THE POKING ROUTINE
860 REM   IS MODIFIED, DEFFN LINE IS
870 REM   THE FIRST LINE OF PROGRAM
880 REM 8.TO SAVE VARIABLES IN CASE
890 REM   OF ANY ERROR IN
900 REM     Y=FNF(X):PRINT ... ETC
910 REM   GOTO LINE# THAT CONTAINS
920 REM     IFAN<>4THENGOSUBXXX
930 REM ============================ ©
```

Defining a Function whilst Running a Program

M. J. Winter
Math Dept.
Michigan State University East Lansing, MI 48824

The following program shows how to allow a user to redefine a function while the program is running. This is a handy technique to have available if one is writing utility programs - to plot graphs, evaluate functions, etc. - for class use. Many students will hesitate to retype an entire line; for all users being able to continue without interruption is a convenience.

The program also includes "protection" against RETURN being pressed with no input entered. Line 210 is

210 INPUT"WHAT'S X□□□□   ";X

where □ denotes a shifted space. (To escape from this, one needs to hold down shift and press STOP).

The program runs on an old-ROM 8K Pet. It should be typed in as it appears, with no extra spaces or quotation marks. What makes it work is that at the end of a Basic program, the machine will execute (up to 10) entries in the keyboard buffer (locations 526-). Location 525 is set to indicate there are two such entries; the 13 means RETURN.

Reference
Mike Lauder, "Dynamic Keyboard", **Pet Users Group Newsletter**, vol 0, No. 4, pp7-8

```
 11 C$="                          "
 12 PRINT"WOULD YOU LIKE TO CHANGE THE ¬
      ¬FUNCTION? Y OR N":PRINTC$:
      ¬PRINT"↑";
 13 GETB$:IFB$=""THEN13
 14 IFB$<>"Y"THEN100
 15 PRINT"ENTER YOUR FUNCTION":PRINT"↓F(X
      ¬)=";:INPUTB$
 20 J=1:GOTO60010
 100 DEF FNA(X)=SIN(X)
 200 PRINT:PRINT
 210 INPUT"WHAT'S X____←←←";X
 220 PRINTFNA(X)
 230 GOTO12
 60010 PRINT"ĥ↓↓
 60011 FORI=JTOJ+1:IFI>JTHEN60015
 60012 PRINT100"DEF FNA(X)="B$
 60013 NEXT
 60014 GOTO60010
 60015 PRINT"RUN": PRINT"h":POKE525,2:
      ¬POKE527,13
 60016 POKE528,13:PRINT"h":END
READY.
```

©

www.commodore.ca

JINSAM

DATA MANAGER

SAVE TIME. SAVE MONEY. Let JINSAM work for you.

★ CUSTOM DATA FILES
★ CUSTOM REPORTS/LABELS
★ KEYED RANDOM ACCESS
★ FAST/EASY/MENU DRIVEN
★ MULTIPLE SEARCH KEYS
★ PRIVACY ACCESS CODES
★ WILD CARD SEARCH

"JINSAM is the best Database Management System for the Commodore Computers !"

JINSAM data manager assists you by intellectually manipulating records.

No more will hundreds of valuable hours be spent searching for needed information. No more will hundreds of hours be spent entering and re-entering information for various reports.

With JINSAM you can truly transform your Commodore Computer into the "state of the art" data processing machine with sophisticated features and accessories found nowhere, even at 10 times the price.

There are three disk based JINSAM. JINSAM 1.0 allows fast and easy file handling, manipulation and report generation. JINSAM 4.0 was designed for the professional and contains features needed in the business environment, such as: JINSORT, a user accessible machine language sort; compaction/expansion of databases, merging databases and much much more. JINSAM 8.0 is our best. JINSAM 8.0 runs on the new Commodore 8032, 80 column display computer. JINSAM 8.0 has all the functions of 4.0 plus additional features found only on the most sophisticated and expensive database management systems.

JINSAM is a new breed of data processing software. Powerful, sophisticated and easy to use. JINSAM has been thoroughly field tested. JINSAM is now installed and saving its owners valuable time and money in educational institutions, research institutions and offices nationwide.

JINSAM was designed with the user in mind. It is a forgiving system with help commands, prompts and utilities for recovering the bulk of data even after power failure, security passwords for privacy, editing, reclaiming space, auto recall, restructuring, unlimited report formats, label printing and a choice of accessory modules all accomplished by a few keystrokes.

JINSAM has 5 accessory interfacing modules:

WORDPROPACK – Intelligent interface for WORDPRO 3 or WORDPRO 4 which creates variable block with data or up to 10 conditions based on database contents. Produce "dunning letters", form letters, report to parent, checks, invoices, etc.

MULTI-LABEL – Prints multiple labels per record with up to 2 lines for messages and consecutive numbering. Produce inventory, bulk mail labels, etc.

MATHPACK – global +, -, x, ÷, by another field or a constant, or zero a field. Sum fields in each record or running sum of single field in all records. Extract information or effect permanent change. Replace in same field or place in a waiting field.

DESCRIPTIVE STATPACK – Determine mean, median, mode, standard deviation, variance, range. Generate histogram and produces Z-Score report.

ADVANCED STATPACK – (you must also acquire DESCRIPTIVE STATPACK). Generates CROSSTABS (number of occurances); CHI SQUARE, LINEAR REGRESSION with graphic representation and prediction. LINEAR CORRELATION and SIMPLE ANALYSIS OF VARIANCE.

All JINSAM accessories are accessed thru the JINSAM menu and require a security password to gain entrance.

JINSAM gives the user **FREEDOM OF CHOICE.** Start with JINSAM 1.0 and upgrade at any time. Choose from the accessory modules available at any time. JINSAM Newsletter brings the latest updates, user input and keeps an eye on the future.

JINSAM alone is reason enough to own a computer. JINSAM can be found at Commodore dealers. Write for the dealer nearest you.

The many features of JINSAM 1.0 – 8.0

JINSAM 1.0 for 16K/32K CBM 2001. Requires CBM 2040 or COMPU/THINK disk — including oldest ROMs. Menu Driven, ISAM – Indexed Sequential access method ● Encripted PASSWORDS for privacy ● Unlimited fields ● unlimited search criteria ● 3 deep subsorts ● .5 – 3 sec retrevial ● editing ● Auto Recall ● Wild Card Capabilities; Reports: multiple headings ● paging ● page numbering ● item count. Labels: any size ● 1-5 across ● sheet or continuous. Utilities: Help commands ● Recover ● Key Dump ● Record Dump ● Descriptor Dump ● Restructure.

JINSAM 4.0 for 32K CBM 2001 with BASIC 4.0. Requires CBM 2040 with DOS 2.1. Has most of JINSAM 1.0 functions Plus + machine sort with user access instructions ● sort 1000 records in apx 10 secs ● Global Compaction/Expantion ● Create new database from existing database ● merge databases. Includes MULTI-LABEL ● 4 deep subsorts. (Available Jan. 13, 1981)

JINSAM 8.0 for Model 8032 with 80 Column screen. Requires 2040 or 8050 disk. Commercial Disk version for 80 Columns, JINSAM 4.0 functions Plus + Displays report formats to screen, 4 deep subsorts. (Available Jan. 1, 1981)

JINSAM is a trademark of JINI MICRO-SYSTEMS, Inc.
WordPro is a trademark of Professional Software Inc.
CBM is a trademark of Commodore Business Machines.

JINSAM Data Manager
for Commodore Computers

— Additional Information

— Jinsam Demo Disk ($10, plus tax)

— Users Guide 1.0 ($25 plus tax)

Please send to:

Name ____________

Position ____________

Company ____________

Address ____________

City, State, Zip ____________

Phone ( ) ____________

Computer, Disk ____________

JINI MICRO SYSTEMS, INC.
Box 274 ● Bronx, NY 10463

Dealer inquiry welcome

www.commodore.ca

Machine Language Addressing Modes

Jim Butterfield

You can find a formal description of addressing modes in any 6502 machine language text or reference. Beginners often find this difficult reading, however.

To help intuitive understanding, I'll describe addressing modes in a different way: in terms of their "reach".

1. Modes that don't reach into memory at all.

Implied addressing means that no extra information is wanted. The Op Code tells the microprocessor everything that needs to be done. If you wish to BRK (Break), which will usually take you to the machine language monitor; or DEY, which decrements the value held in the Y register; or use any other implied address command, there's no need for further information - just give the command in one byte.

Immediate addressing supplies the value you need right away in the location following the Op Code. No need for an extra trip to memory: the actual value follows the instruction. So you may load the X register with the value zero by giving LDX #0 (A2 00); or compare Y to the value five with CPY # (C0 05); or add ten to the A register with ADC #10 (69 0A) - it all works simply and fast. Immediate addressing instructions use two bytes, of course.

2. Modes that reach a single location in memory.

Absolute addressing takes you to any address you give. The address itself uses two bytes of memory and is stored in the usual backwards 6502 format -for example, hex address 1234 will be stored as 34 and 12. If you want to store the contents of the Y register into address hexadecimal 2300, you'd code STY $2300 (8C 00 23); or if you wanted to compare the A register with the contents of location hex 027A, you'd write CMP $027A (CD 7A 02). Absolute addressing instructions use three bytes: one for the Op Code and two for the address.

Zero page addressing takes you to any address from hex 0000 to 00FF. Absolute addressing can take you there too, of course; but zero page addressing will be faster and save you a byte of memory. Your address will be only one byte long: a value from 00 to FF to indicate the location in zero page you want to access.

Zero page locations are in short supply on the PET. Use them sparingly; if possible reserve zero page locations for indirect addressing, which will be dealt with later.

3. Modes that reach a range of 256 locations.

Absolute indexed address modes allow you to reach out from the location you specify. You can reach from that address to any higher address up to 255 locations above. You might like to think of it as a robot sitting on the location you have named, equipped with an adjustable arm. The arm can reach out in one direction only (towards higher addresses) and can't reach further than 255 locations. You can adjust the distance the arm reaches by setting the contents of the index register; sometimes this is register X and sometimes register Y.

A reach of 255 is plenty to pick through a line x of text, a cassette buffer, or a table of ASCII characters.

Zero page indexed allows you to reach out from a zero page location. This is similar to absolute indexed addressing, but with one important difference: you'll never leave zero page. If you try to reach beyond address hex 00FF, you'll wrap around and start reaching location 0000 and up.

This can be very useful, since it gives the effect of allowing a negative index value: you can reach locations below the ones you specify. If you're using zero page X indexing, for example, and set the value 255 (hex FF) into the X register you'll end up accessing the location below the one you named in the instruction.

Relative addressing is used for branch instructions only. This is where you do all your decision-making in a program. You may branch ahead up to 127 locations; you may branch back up to 128 locations. That's not very far in a big program, so in many cases you'll want to couple your branch instruction with a JMP (Jump) which uses absolute addressing and can take you anywhere in memory you want to go.

4. Modes that reach anywhere in memory.

Indirect addressing is a scheme which allows you to set up the address you want to use somewhere in memory; later, you tell the processor to go to that location and use the address that you've put there. The processor takes two shots at memory - the first to get the actual address, and the second to deal with the contents of that address.

Here's the powerful part: since you can set up and change the indirect address to anything you like, the instruction using indirect addressing can reach anywhere at all in memory. You could set up the in-

www.commodore.ca

Skyles Electric Works

Presenting the Skyles MacroTeA

Text Editor

To help you write your program, MacroTeA includes a powerful text editor with **34 command functions:**

AUTO	Numbers lines automatically.
NUMBER	Automatically renumbers lines.
FORMAT	Outputs text file in easy-to-read columns.
COPY	Copies a line or group of lines to a new location.
MOVE	Moves a line or group of lines to a new location.
DELETE	Deletes a line or group of lines.
CLEAR	Clears the text file.
PRINT	Prints a line or group of lines to the PET screen.
PUT	Saves a line or group of lines of text on the tape (or disc).
GET	Loads a previously saved line or group of lines of text from the tape (or disc).
DUPLICATE	Copies text file modules from one tape recorder to the other. Stops on specific modules to allow changes before it is duplicated. This command makes an unlimited length program (text file) practical.
HARD	Prints out text file on printer.
ASSEMBLE	Assembles text file with or without a listing. Assembly may be specified for the object code (program) to be recorded or placed in RAM memory.
PASS	Does second pass of assembly. Another command that makes unlimited length text files (source code) practical.
RUN	Runs (executes) a previously assembled program.
SYMBOLS	Prints out the symbol table (label file).
SET	Gives complete control of the size and location of the text file (source file), label file (symbol table) and relocatable buffer.
DISK	Gives complete access to the eleven DOS commands; PUT GET NEW INITIALIZE DIRECTORY COPY DUPLICATE SCRATCH VALIDATE RENAME ERROR REPORT
EDIT	Offers unbelievably powerful search and replace capability. Many large computer assemblers lack this sophistication.
FIND	Searches text file for defined strings. Optionally prints them and counts them; i.e., this command counts number of characters in text file.
MANUSCRIPT	Eliminates line numbers on PRINT and HARD command. Makes MacroTea a true and powerful Text Editor.
BREAK	Breaks to the Monitor portion of MacroTea. A return to Text Editor without loss of text is possible.
USER	Improves or tailors MacroTea's Text Editor to user's needs; "Do-it-yourself" command.

Fast...Fast Assembler

Briefly, the pseudo-ops are:

• BA	Commands the assembler to begin placing assembled code where indicated.
• CE	Commands the assembler to continue assembly unless certain serious errors occur. All errors are printed out.
• LS	Commands the assembler to start listing source (text file) from this point on.
• LC	Commands the assembler to stop list source (text file) from this point in the program.
• CT	Commands the assembler to continue that source program (text file) on tape.
• OS	Commands the assembler to store the object code in memory.
• OC	Commands the assembler to not store object code in memory.
• MC	Commands the assembler to store object code at location different from the location in which it is assembling object code.
• SE	Commands the assembler to store an external address.
• DS	Commands the assembler to set aside a block of storage.
• BY	Commands the assembler to store data.
• SI	Commands the assembler to store an internal address.
• DE	Commands the assembler to calculate an external label expression.
• DI	Commands the assembler to calculate an internal label expression.
• EN	Informs the assembler that this is the end of the program.
• EJ	Commands the assembler to eject to top of page on printer copy.
• SET	A directive not a pseudo-op, directs the assemblers to redefine the value of a label.

Macro Assembler

The macro pseudo-ops include:

MD	This is a macro beginning instruction definition.
ME	This is end of a macro instruction definition.
EC	Do not output macro-generated code in source listing.
ES	Do output macro-generated code in source listing.

Conditional Assembler

The conditional assembly pseudo-ops are:

IEQ	If the label expression is equal to zero, assemble this block of source code (text file).
INE	If the label expression is not equal to zero, assemble this block of source code (text file).
IPL	If the label expression is positive, assemble this block of source code.
IMI	If the label expression is negative, assemble this block of source code.
•••	This is the end of a block of source code.

Enhanced Monitor

. . . By having 16 powerful commands:

A	Automatic MacroTeA cold start from Monitor.
Z	Automatic MacroTeA warm start from Monitor.
F	Loads from tape object code program.
S	Saves to tape object code between locations specified.
D	Disassembles object code back to source listing.
M	Displays in memory object code starting at selected location. The normal PET screen edit may be used to change the object code.
R	Displays in register. Contents may be changed using PET screen edit capabilities.
H	Hunts memory for a particular group of object codes.
W	Allows you to walk through the program one step at a time.
B	Breakpoint to occur after specified number of passes past specified address.
Q	Start on specified address. Quit if STOP key or breakpoint occurs.
T	Transfers a program or part of a program from one memory area to another.
G	Go!! Runs machine language program starting at selected location.
X	Exits back to BASIC.
I	Display memory and decoded ASCII characters.
P	Pack (fill) memory with specified byte.

What are the other unique features of the MacroTeA?

- Labels up to 10 characters in length
- 50 different symbols to choose from for each character
- 10^{16} different labels possible
- Create executable object code in memory or store on tape
- Text editor may be used for composing letters, manuscripts, etc.
- Text may be loaded and stored from tape or disc
- Powerful two-cassette duplicator function
- String search capability
- Macros may be nested 32 deep
- 25 Assembler psuedo-ops
- 5 Conditional assembler psuedo-ops
- 40 Error codes to pinpoint problems
- 16 Error codes related to Macros
- Warm-start button
- Enhanced monitor with 16 commands

...a completely solid state firmware system ...all in ROM and RAM. No tapes to load. The system is available from the time you turn on your PET to the time you shut it off.

15 chips on a single high quality printed circuit board; interfaces with PET's parallel address and data bus or with Skyles Memory Adapter. A comprehensive 170 page manual is included

Truly, there is simply no other system of this magnitude at anywhere near this price. **$395.00 ***

**California residents: please add 6% or 6.5% sales tax as required*

VISA, MASTERCHARGE ORDERS CALL **(800) 538-3083** (except California residents)
CALIFORNIA ORDERS PLEASE CALL (408) 257-9140

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

www.commodore.ca

direct address to point at location 0000; as the program runs, keep adding to it until it finally points at hex location FFFF; and you will have reached every one of the 65536 locations in memory.

Jump Indirect is used mostly to link programs in ROM (Read Only Memory) to user routines. ROM programs are fixed and inflexible; but if you want to add your own code, it may be permitted with an indirect jump from the ROM. Here's how it works: The ROM coding first sets up its own indirect address into memory. Later, it will go to certain routines by jumping via this address. If you change this address, the ROM program will jump to wherever you say. PET uses this type of system to allow you to change such things as the interrupt routines: the hardward interrupt vector (hex 0219-021A on original ROMs, 0090-0091 on newer machines) is used as an indirect address, for example.

Indirect Indexed addressing is the most commonly used indirect addressing mode on the 6502. You may recall that indexed addressing allowed you to reach up to 255 locations above a fixed address. Indirect indexed is similar except that it allows you to reach up to 255 locations above a variable address - which is, of course, your indirect address. Register Y is always used for indexing in this case.

This is so powerful - and so handy - that it deserves further comment. You may set the indirect address to the start of a line of text, and then use the Y index to look through that line. You may set the indirect address to the start of either cassette buffer area, and use the Y index to look into that buffer.

Indirect Indexed addressing has one important aspect: the indirect address must go into zero page: hex addresses 00 to fF. You need to conserve zero pae locations so that you'll have room to stage all the indirect addresses you need.

Indexed Indirect isn't used too much on the PET. Its purpose is this: if you have a bunch of indirect addresses neatly arranged in zero page, you can use the X register to select which indirect address you wish to use.

It's hard to find space for a number of indirect addresses in zero page. Most programmers just set X to zero and use a single indirect address. In certain special cases, however, Indexed Indirect addressing can be very powerful indeed. Within the PET 2040 disk, for example, the X register selects one of fifteen buffer pointers; then, indexed indirect addressing allows a character to be written to or read from the appropriate place in the correct buffer.

A Note on Jumps and Branches

The beginning programmer will find that the instructions which transfers program control - the Jumps and Branches - seem to have very limited versatility. Apart from the indirect jump, which isn't used much in applications coding, the programmer seems to be limited to jumping and branching to fixed locations.

Surprisingly, that's often a good thing. It's similar to the deliberate restriction in Basic that forbids you to code GOTO X + 20. It can be argued that a program that is written with a fixed control structure will be healthier than one that might jump any place.

Even so, you can get around this constraint if you wish. You can construct the equivalent of an Indexed Indirect jump very neatly by a little clever manipulation of the stack.

But that's a subject for a future column.

Summary and Self-Test.

You should have a feeling for the various addressing modes and how they are used. Try your hand at this quiz, and see how your answers match those given.

1. You wish to test the status word ST, which is at location hex 96 on new ROMs. What address mode?

   Answer: Zero Page addressing.

2. As part of the test for an alphabetic character, you want to compare the A register to hex 41 to see if it is greater or equal. What address mode?

   Answer: Immediate addressing.

3. You want to look through the table of logical file numbers, stored in hex locations 0251-025A, to see if logical file number 4 is in use. What addres mode?

   Answer: Absolute Indexed addressing.

4. You want to get each byte of the floating point accumulator locations 005E-0063, to store them somewhere else. What mode?

   Answer: Zero Page Indexed addressing.

5. If the status register Z flag is set, you want to skip the next four instructions, which occupy 7 bytes. What mode?

   Answer: Relative addresing.

6. You want to clear a line on the screen; it's not always the same line. What addressing mode?

   Answer: Indirect Indexed addressing. Set the indirect address to the start of the line; then use the Y register to clear the individual positions.

7. You want to increase the contents of the Y register by one. What addressing mode?

   Answer: Implied addressing. The INY instruction will do the whole job. ©

www.commodore.ca

NEW LIFE for old Pets™?

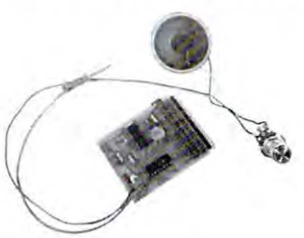
Is it a dream...is it fantasy?

Switch from old ROMs to new Retrofit ROMs or from the new 24 pin ROMs to the new Commodore® Basic 4.0 with the Basic Switch.

- Not sure about the ROM Retrofit Kit from Commodore? Now you can use all three sets of Commodore ROMs and others as well.
- The Basic Switch allows switch selection of either ROM set (your original set or your retrofit set) from Commodore. Plus, Models 15-A and 15-B include an additional zero insertion force socket allowing easy use of ROMs like the BASIC Programmer's Toolkit...concurrently.
- Models 15-A and 15-B The Basic Switch plus...includes expanded cable assembly and zero insertion force socket. Your 15th ROM simply plugs in...enabled while either ROM set is selected. Socket 15 may be readdressed by the user for additional flexibility.
- The Basic Switch is sold in assembled form only. All models are designed for easy attachment to your PET with a convenient cable assembly. No soldering or drilling is required. The Basic Switch mates with a cable assembly at your primary board, and does not use the physical connectors of any PET ports.
- Our prices and complete product specifications are available by contacting APPLIED MICRO SYSTEMS, Mishawaka, Indiana; or any Commodore Dealer.

Now available for Commodore® Basic 4.0

Other APPLIED MICRO SYSTEMS Products include:



KEYTONE: Audio Feedback for your PET

- Monitors PET® keyboard, emitting a short beep when any key is pressed.
- Easy, solderless installation.
- Works with old and new PETs.
- Completely self contained, with speaker and externally accessible volume control.

Price $49.95 FOB Mishawaka, IN

PET® to Centronics® Printer Interface

- No software drivers required.
- Retains IEEE port to facilitate use with other devices, i.e., disk systems and/or plotters, etc.
- No external power supplies required, negligible power needed is derived from the printer.
- PET/CBM software compatible.
- Attractive metal case.

Price $139.95 FOB Mishawaka, IN

MEMORADD: Memory Expansion for your PET

- Adds 16K bytes to your present 16K PET.
- Fast, easy, solderless installation.
- Small (6¼" x 1⁵⁄₁₆" x ⅜") or (16cm x 3.5cm x .9cm).
- Derives power from PET.
- Can be used only with 16K PETs that presently use 4116 RAMs.

Price $199.95 FOB Mishawaka, IN

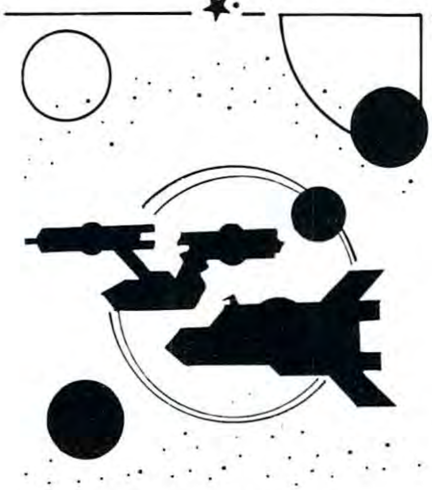
Dealer inquiries are encouraged. Free MACHINE LANGUAGE MONITOR COMMANDS List will be included with pricing and product specification requests.

APPLIED MICRO SYSTEMS

3502 Home Street, Mishawaka, Indiana 46544 • 1-219-259-3787 (Indiana) • 1-800-348-7208

www.commodore.ca

Get Fireworks From Your PET*!



TREK-X

TREK-X Command the Enterprise as you scour the quadrant for enemy warships. This package not only has superb graphics, but also includes programming for optional sound effects. A one-player game for the PET 8K. **Order No. 0032P $7.95.**

DUNGEON OF DEATH

DUNGEON OF DEATH Battle evil demons, cast magic spells, and accumulate great wealth as you search for the Holy Grail. You'll have to descend into the Dungeon of Death and grope through the suffocating darkness. If you survive, glory and treasure are yours. For the PET 8K. **Order No. 0064P $7.95.**

ARCADE I

ARCADE I This package combines an exciting outdoor sport with one of America's most popular indoor sports:

•**Kite Fight**—It's a national sport in India. After you and a friend have spent several hours maneuvering your kites across the screen of your PET, you'll know why!

•**Pinball**—By far the finest use of the PET's exceptional graphics capabilities we've ever seen, and a heck of a lot of fun to boot.

Requires an 8K PET. **Order No. 0074P $7.95.**

CODE NAME: CIPHER

CODE NAME:CIPHER

Enjoy that same feeling of intrigue and discovery with the Code Name: Cipher package. Included are:

•**Memory Game**—Would you like to match your memory against the computer's? You can with the Memory Game.

•**Codemaster**—One player types in a word, phrase, or sentence, and the PET translates that message into a cryptogram. The other player must break the code and solve the cryptogram in the shortest time possible.

•**Deceitful Mindmaster**—This isn't your ordinary Mastermind-type game. You must guess the five letters in the hidden code word.

•**Code Breaker**—Cracking this code won't be as easy as cracking walnuts. You'll need to flex your mental muscles to win this game.

If you want a mental challenge, then Code Name: Cipher is for you. For the 8K PET. **Order No. 0112P. $7.95.**

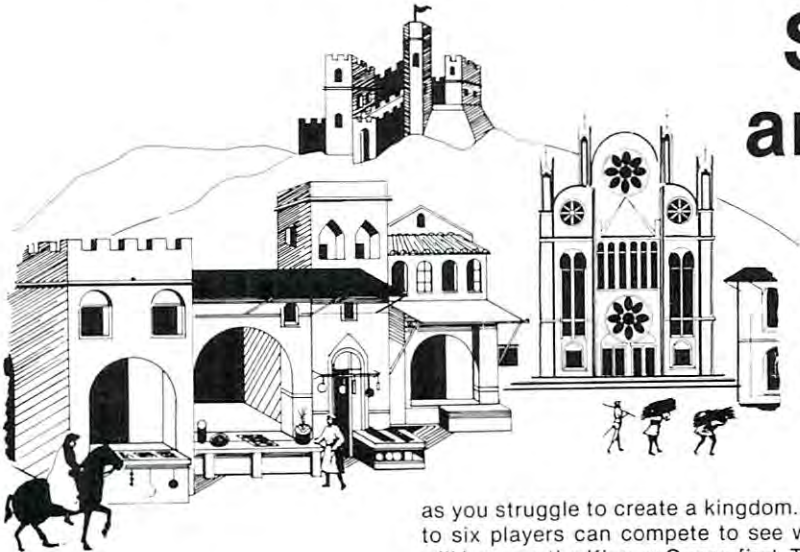
Instant Software™

*A trademark of Commodore Business Machines

PETERBOROUGH, N.H. 03458
603-924-7296

www.commodore.ca

Captivate Yourself.



Santa Paravia and Fiumaccio

The most captivating and engrossing program ever made for the PET*

SANTA PARAVIA AND FIUMACCIO Become the ruler of a medieval city-state as you struggle to create a kingdom. Up to six players can compete to see who will become the King or Queen first. This program requires a PET 16K. **Order No. 0175P.$9.95.**

It is the dawn of the 15th Century; you rule a tiny Italian city-state. Your goal: The Crown!

Up to six players can compete as rulers of neighboring cities. You control the grain harvest, feed your serfs, set tax rates, dispense justice and invest in public works.

The future of your realm will depend on your decisions. If they are wise, your city-state will grow and you will acquire loftier titles. If your rule is incompetent, your people will starve and you may be invaded by your neighbors.

How will *you* rule your kingdom? Will you be an enlightened leader—or an unscrupulous despot? Only you can answer that question—with Santa Paravia and Fiumaccio.

CHIMERA

CHIMERA If you think the legendary Chimera was hard to handle, wait until you try the Chimera package. Included are:

•**Reflex**—Round and round the little white ball rolls. Only fast reflexes can guide it into the center of the maze.

•**Dragon**—You'll have to shoot down those pesky, fire-breathing dragons with your cannon. If you succeed your castle will be safe, if not it will mean a call to your fire insurance company. For one player.

•**Dungeon**—A very punctual guard comes down to the dungeon every day to torture you. This means that you have only thirty seconds to dig your way under the castle to freedom. For one player.

•**Dragon Hunt**—You must go forth and slay a fire-breathing dragon. The only thing that will protect you from the flames is your shield, if you know when to use it. For one player.

•**Dropoff**—You must make your opponent's men "dropoff" the board by moving and firing your own men. For one or two players. **Order No. 0110P. $9.95.**

PET DEMO I

PET DEMO I You can give yourself, your family, and your friends hours of fun and excitement with this gem of a package.

•**Slot Machine**—You won't be able to resist the enticing messages from this computerized one-armed bandit.

•**Chase**—You must find the black piece as you search through the ever-changing maze.

•**Flying Pheasant**—Try to shoot the flying pheasant on the wing.

•**Sitting Ducks**—Try to get your archer to shoot as many ducks as possible for a high score.

•**Craps**—It's Snake Eyes, Little Joe, or Boxcars as you roll the dice and try to make your point.

•**Gran Prix 2001**—Drivers with experience ranging from novice to professional will enjoy this multi-leveled race game.

•**Fox and Hounds**—It's you against the computer as your four hounds try to capture the computer's fox.

For true excitement, you'll need a PET 8K. **Order No. 0035P $7.95.**

DOW JONE$

DOW JONES Up to six players can enjoy this exciting stock market game. You can buy and sell stock in response to changing market conditions. Get a taste of what playing the market is all about. Requires a PET with 8K. **Order No. 0026P $7.95.**

We Guarantee It!

Instant Software Guarantee

OUR PROGRAMS ARE GUARANTEED TO BE QUALITY PRODUCTS. IF NOT COMPLETELY SATISFIED YOU MAY RETURN THE PROGRAM WITHIN 60 DAYS. A CREDIT OR REPLACEMENT WILL BE WILLINGLY GIVEN FOR ANY REASON.

Prices subject to change without notice.

TO ORDER

SEE YOUR LOCAL INSTANT SOFTWARE DEALER

OR

call now Toll-Free 1-800-258-5473

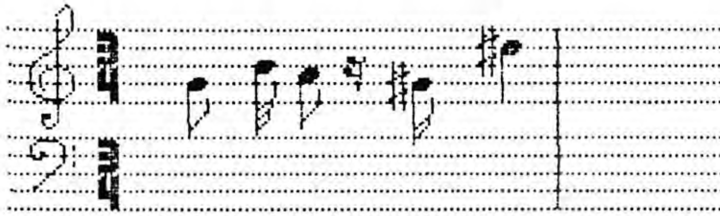
Instant Software™

*A trademark of Commodore Business Machines

PETERBOROUGH, N.H. 03458
603-924-7296

www.commodore.ca

Visible Memory Printer Dump

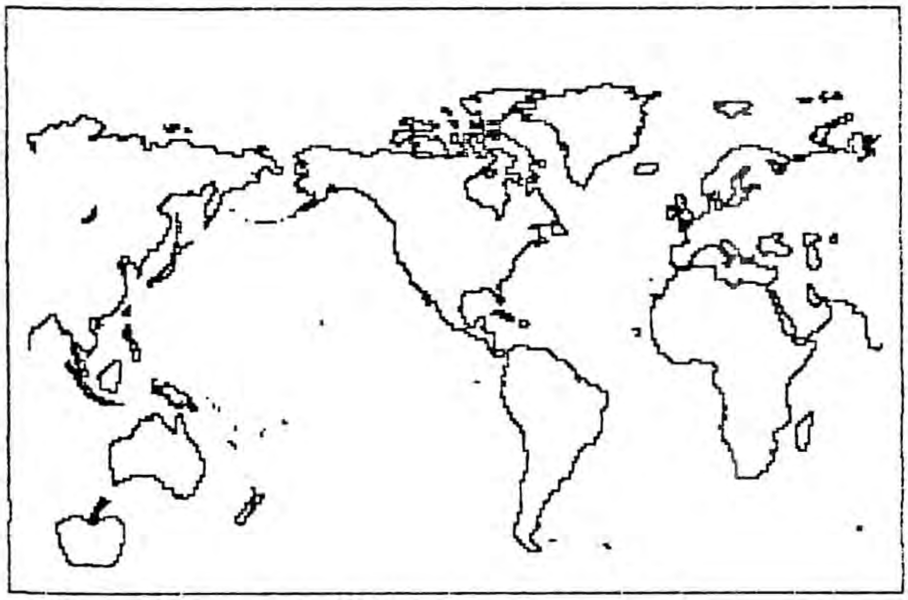


Dr. Frank Covitz

The MTU visible memory is 8K bytes of dynamic RAM which, during refresh (transparent to the 6502), generates a video image of itself. The display signal is standard composite video, and can be seen on a conventional monitor or converted TV set. With the MTU/PET interface, the PET screen itself can be used as the display. The 320 (horizontal) by 200 (vertical) pixel matrix 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).

The following 6502 program allows you to get a hard copy of this on the CBM 2022 (tractor feed) PRINTER. The first part is fairly self-documenting, and is used to open the special character channel to the printer and set the vertical spacing (not available on the CBM 2020 pressure feed printer). The VMDUMP machine code scans each 320H by 7V line to form the special character matrix. Skipping the proper number of spaces, the main program then prints this character.

This is a SLOW process, since the 2022 printer can handle only one special character per line, so as many as 53 prints to the same line may be required before the line is complete. Since 30 lines may be needed to complete the 8K scan, the whole process can take up to 30 minutes to finish!! In practice, however, sinces spaces are "weeded out", 5-10 minutes is usually sufficient to get a moderately dense print-out, and less for line-type graphs. Horizontal and vertical

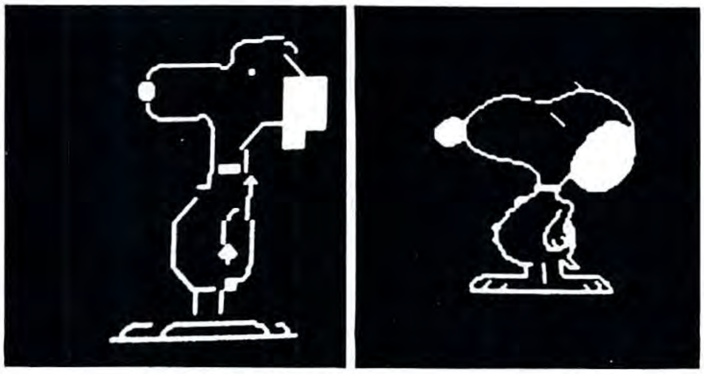


VM PRNTR DUMP
2

```
                ;
                ;*****************
                ;*               *
                ;* VM PRNTR DUMP *
                ;*               *
                ;* BY F.COVITZ   *
                ;*               *
                ;*****************
                ;
                ;
                ;THIS PROGRAM DUMPS AN 8K BLOCK
                ;TO THE CBM 2022 PRINTER AS A
                ;BIT MAPPED IMAGE.
                ;MAINLY INTENDED FOR USE WITH THE
                ;M.T.U. 8K VISIBLE MEMORY.
                ;FORMAT IS 320H*200V PIXELS
                ;
0280                    *=    $0280           ;ORIGIN (EXAMPLE ONLY)
                ;
                ;*** UPGRADE ROM Z-PAGE LOC'NS ***
                ;
0280                LENGTH  =   $D1             ;LENGTH OF FILE NAME
0280                LFN     =   $D2             ;LOGICAL FILE NO.
0280                SA      =   $D3             ;SECONDARY ADDRESS
0280                DN      =   $D4             ;DEVICE NO.
                ;
                ;*** CONSTANTS ***
0280                LF1     =   1               ;NORMAL PRINT CHANNEL
0280                SA1     =   0               ;NORMAL PRINT SEC.ADDR.
0280                LF2     =   2               ;VERT. SPACING CHANNEL
0280                SA6     =   6               ;VERT. SPACING SEC.ADDR.
0280                LF3     =   3               ;SPEC. CHAR. CHANNEL
0280                SA5     =   5               ;SPEC. CHAR. SEC.ADDR.
0280                PRNTR   =   4               ;PHYSICAL DEVICE # OF PRINTER
0280                SPACE   =   " "
0280                CR      =   $8D             ;CARRIAGE RETURN (WITHOUT LF)
0280                CRLF    =   $0D             ;CARRIAGE RETURN (WITH LF)
0280                MAX     =   54              ;MAX # OF SPACES (54*6=324)
0280                REV     =   $FE             ;'REVERSE' KEY
0280                SCHAR   =   $FE             ;SPEC.CHARACTER
```

www.commodore.ca

• Judge this character generator for yourself •



Regular PET Graphics — HAL GRAPHICS

Exciting games are now available for the HAL PCG 6500 ... Jupiter Lander, Night Driver, Super Slot and more ... $19.95 each.

Compare the two Snoopys and you'll begin to see why the new HAL PCG 6500 is attracting so much attention. It's the programmable character generator for your PET that gives sharp, detailed graphics.

Important features include:

- Storage for 64 new characters
- Built-in CB2 sound amplifier that produces all tones in the human hearing range
- PCG Manual and demo program
- Interface to PET/CBM with 24 pin character generator ROMS. (If your PET uses 28 pin ROMS, Conversion Kits are available for $45. To simplify ordering, please indicate memory RAM #2114 or #6550.)

The HAL PCG 6500 is available exclusively at Systems Formulate ...

Only $200 (plus $5 shipping & handling)

Add $45 for 28 pin Conversion Kit

MIPLOT: the right plotter at the right price

MIPLOT by Watanabe Corporation

Designed for straight forward interface to any microcomputer that outputs the ASC11 code, MIPLOT can even be used by operators with no plotter experience.

- Incorporates pre-programmed "intelligent" functions required for producing graphs and drawings
- Solid and broken line types can be specified
- Built in character generator for letters, numbers and symbols
- Characters can be enlarged and rotated to four orientations
- Special printer mode outputs character data as-is
- Uses commonly available hard fiber-tip pens
- Maximum plot speed approximately 2 inches per second
- Built in self-test mode

Only $1,200

at Systems Formulate Corporation

(plus shipping & handling)

ADCOM Light Pen (with sound) ... Only $34⁹⁵

(plus $2 shipping & handling)

Unlike many light pens, the ADCOM interfaces with PET through the second cassette port rather than the parallel port. The result? Sound!

- Ready to plug in, no assembly necessary.
- Faster than many other light pens designed for PET.
- Built-in sensitivity control.
- Machine language and demo program included for ease of programming.

Expand with the finest in micro peripherals, systems and software from Systems Formulate.

Call today for more information or to place your phone order: (415) 326-9100

We honor Master Charge, Visa, check or money order.

(California residents add 6.5% tax)

SYSTEMS FORMULATE CORPORATION

39 Town & Country Village • Palo Alto, California 94301

www.commodore.ca

registration are not perfect, given the limitations of the 2022's mechanism, and there is some distortion (vertical or horizontal, depending on your outlook). The accompanying examples should give you some idea of the results you can expect. Given the slow speed and imperfect registration, the fact still remains that the capability is there.

As written it is configured to go into both tape buffers (starts at $0280, 640dec.). It is then easily accessed by DISK-based systems. When you see something you like on the VM, just key in SYS640 and take a five minute break. If you have a tape-based system, you will need to relocate the code elsewhere. The assembly source should make this relatively easy to do.

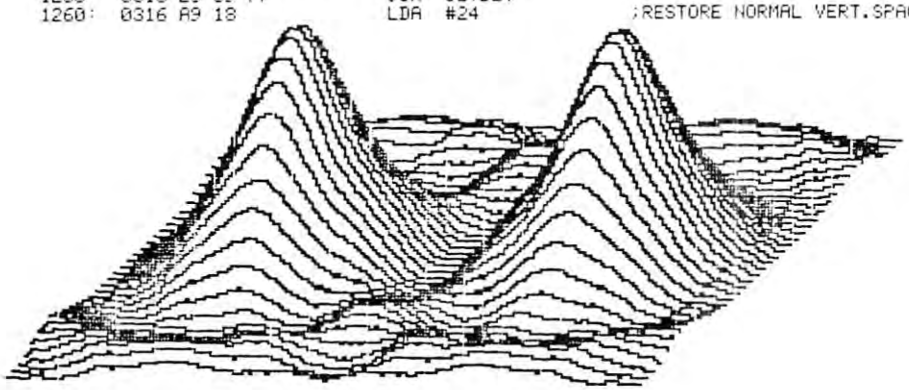
The only routine that is specific to upgrade ROM is the OPEN subroutine. I believe this is at $F52D in original ROM. Zero-page locations $D1, $D2, $D3 and $D4 are ROM dependent, and correspond to locations $EE, $EF, $F0, and $F1, respectively for original ROM. Zero page locations 1 and 2 are used as an indirect pointer. The last point to be aware of is the setting of VMORG, the origin page of the 8K block of memory. There is a single LDA #VMORG in the source code, so you must change this single byte if you want a dump of a different 8k block.

To be even more benign to the calling program, the original contents of zero page locations 1 and 2 (VM and VM + 1), as well as the registers, could be saved. Use the following code sequence:

```
MAIN SEI ;prevent interrupts
  PHP ;save status
  TXA ;save registers
  PHA
  TYA
  PHA
  LDA VM ;save loc'ns 1 and 2
  PHA
  LDA VM + 1
  PHA
  .
  .
  .
```

```
                  ;
                  ;*** SYSTEM LOCATIONS ***
                  ;***                  ***
                  ;*** ! = UPGRADE ROM  ***
                  ;
0280                  PIAK     =    $E812          ;KEYBOARD PORT
0280                  CLOSE    =    $FFE7          ;CLOSE IEEE DEVICES
0280                  OPEN     =    $F524          ;! OPEN IEEE DEVICE
0280                  SETDEV   =    $FFC9          ;SET OUTPUT DEVICE
0280                  OUTCHAR  =    $FFD2          ;SEND 1 CHAR.

0280 78               MAIN     SEI                 ;PREVENT INTERRUPTS
0281 A9 00                     LDA  #0             ;SET NULL NAME
0283 85 D1                     STA  LENGTH
0285 85 D3                     STA  SA             ;OPEN NORMAL PRINT CHANNEL
0287 A9 01                     LDA  #LF1
0289 85 D2                     STA  LFN
028B A9 04                     LDA  #PRNTR
028D 85 D4                     STA  DN
028F 20 24 F5                  JSR  OPEN
0292 A9 02                     LDA  #LF2           ;OPEN VERT.SPACING CHANNEL
0294 85 D2                     STA  LFN
0296 A9 06                     LDA  #SA6
0298 85 D3                     STA  SA
029A 20 24 F5                  JSR  OPEN
029D A9 03                     LDA  #LF3           ;OPEN SPEC.CHAR.CHANNEL
029F 85 D2                     STA  LFN
02A1 A9 05                     LDA  #SA5
02A3 85 D3                     STA  SA
02A5 20 24 F5                  JSR  OPEN
02A8 A2 01                     LDX  #LF1           ;SEND 'HOME'
02AA 20 C9 FF                  JSR  SETDEV
02AD A9 13                     LDA  #"ꟿ"
02AF 20 D2 FF                  JSR  OUTCHAR
02B2 A2 02                     LDX  #LF2           ;SEND VERTICAL SPACING
02B4 20 C9 FF                  JSR  SETDEV
02B7 A9 10                     LDA  #16
02B9 20 D2 FF                  JSR  OUTCHAR
02BC 20 20 03                  JSR  VMDUMP         ;INITIAL CALL
02BF AD 12 E8 MAIN1            LDA  PIAK           ;CHECK 'REVERSE' KEY
02C2 C9 FE                     CMP  #REV
02C4 F0 4B                     BEQ  MAIN7          ;ABORT ON REVERSE KEY
02C6 AD FE AF                  LDA  SPC
02C9 C9 36                     CMP  #MAX           ;CHECK IF LINE FEED NEEDED
02CB D0 04                     BNE  MAIN2
02CD A9 0D                     LDA  #CRLF          ;SEND A CRLF
02CF D0 35                     BNE  MAIN5
02D1 A2 03            MAIN2    LDX  #LF3           ;SPEC.CHAR.CHANNEL
02D3 20 C9 FF                  JSR  SETDEV
02D6 A0 06                     LDY  #6             ;6 BYTES TO SEND
02D8 B9 F5 AF MAIN3            LDA  MATRIX-1,Y     ;TRANSMIT TO PRINTER
02DB 20 D2 FF                  JSR  OUTCHAR
02DE 88                        DEY
02DF D0 F7                     BNE  MAIN3
02E1 38                        SEC                 ;HOW MANY SPACES TO TRANSMIT
02E2 A9 38                     LDA  #MAX+2         ;LEAVE A LITTLE ROOM
02E4 ED FE AF                  SBC  SPC
02E7 A8                        TAY                 ;USE AS INDEX
02E8 A2 01                     LDX  #LF1           ;SET FOR NORMAL PRINT CHANNEL
02EA 20 C9 FF                  JSR  SETDEV
02ED A9 20            MAIN4    LDA  #SPACE
02EF 20 D2 FF                  JSR  OUTCHAR        ;SEND SPACES TO PRINTER
02F2 88                        DEY
02F3 D0 F8                     BNE  MAIN4
02F5 A2 01                     LDX  #LF1
02F7 20 C9 FF                  JSR  SETDEV
02FA A9 FE                     LDA  #SCHAR         ;TRANSMIT SPECIAL CHAR.
02FC 20 D2 FF                  JSR  OUTCHAR
02FF A2 01                     LDX  #LF1
0301 20 C9 FF                  JSR  SETDEV
0304 A9 8D                     LDA  #CR            ;SEND CR (WITHOUT LF)
0306 20 D2 FF MAIN5            JSR  OUTCHAR
0309 20 77 03 MAIN6            JSR  VMDNEXT        ;RE-ENTER VMDUMP
030C AD FF AF                  LDA  LINES          ;CHECK FOR DONE
030F D0 AE                     BNE  MAIN1
0311 A2 02            MAIN7    LDX  #LF2           ;CLOSE ALL CHANNELS
0313 20 C9 FF                  JSR  SETDEV
0316 A9 18                     LDA  #24            ;RESTORE NORMAL VERT.SPACING
```



www.commodore.ca

. .ECX Co. .

Specializing in peripherals and software for the PET

We offer the following new products

*Full IEEE-488 Bus Acoustic MODEM . . . $395.00

*Full IEEE-488 Bus to Centronics or NEC Spinwriter Interface COM-PLICATIONS Model C101 . . . . $225.00

*Full IEEE-488 Bus to Watanabe Graphics Plotter Interface COM-PLICATIONS Model C102 $295.00

***Watanabe DIGIPLOT Intelligent Graphics Plotter Watanabe Model WX4671 . . . . $1200.00**

"CURVE Graphics Software Package: Runs on the 32K PET connected to the Watanabe DIGIPLOT via the C102 IEEE-488 Bus Interface Adapter. No computer skills needed to use, contains complete graphic softwear program and 16 subroutines . . . . $299.00"

*PET Computer Desk, designed for PET computer, disk and printer $395.00

*ALL COMMODORE COMPUTERS, PERIPHERALS AND SOFTWARE

We are announcing the following new products:

***C301** PET to DIABLO Printer Interface

***C302** PET to QUME Printer Interface

***C231** Full IEEE-488 Bus to RS232C Uni-directional Interface

***C232** Full IEEE-488 Bus to RS232C Bi-directional Interface

ECX Co. offers technical support and fast in-house maintenance and service for all the products we sell. We know how to repair Commodore equipment! EXC Co. is owned and operated by COM-PLICATIONS, INC., a design and development corporation. Call us, we talk technical! ALL COM-PLICATIONS INC. products are manufactured to industrial quality standards. If you need to know more about the IEEE-488 Bus read our just-published Osborne/McGraw Hill new book, **"The PET and The IEEE-488 (GPIB) Bus"**, authored by the president of COM-PLICATIONS, INC., available from ECX Co. for $18.00 (includes tax and shipping).

P.S. All of our IEEE-488 Bus interfaces meet ALL of the IEEE-488 Bus performance requirements!

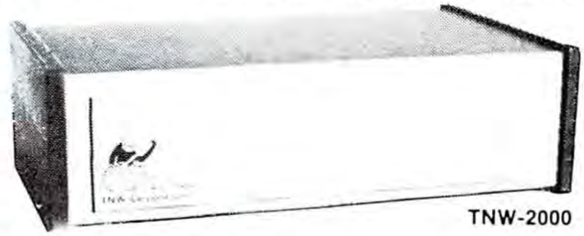
ECX Co.
2678 N. Main Street #6
Walnut Creek, CA 94596
(415) 944-9277

Northern California's Fastest-Growing Exclusive Commodore Computer Dealer
Dealer Inquiries Solicited

IEEE-488 BUS

SYSTEM BUILDING BLOCKS

For Commodore PET/CBM and other computers...



TNW-2000

TNW-1000 Serial Interface: $129
1 channel output only

TNW-2000 Serial Interface: $229
1 channel input and output

TNW-232D Dual Serial Interface: $369
2 channels input and output plus RS 232 control lines

TNW-103 Telephone Modem: $389
Auto answer, auto dial. Use with DAA

SOFTWARE

PTERM: A program that turns your PET into a terminal (Use with TNW-2000, TNW-232D, or TNW 103)

SWAP: Allows storage of up to 8 programs in PET memory at once. Run them in any order

PAN: A sophisticated electronic mail program (use with TNW-103)

PLUS Most popular computers, disks, printers, etc.

TNW CORPORATION

Write or call for information today:

TNW Corporation
3351 Hancock Street
San Diego CA 92110
(714) 225-1040

PET' MACHINE LANGUAGE GUIDE

Contents include sections on:

- Input and output routines.
- Fixed point, floating point, and Ascii number conversion.
- Clocks and timers.
- Built-in arithmetic functions.
- Programming hints and suggestions.
- Many sample programs.

While supply lasts:
Guides for Old ROMS
only $5.00 inc. postage
New ROMS order below

If you are interested in or are already into machine language programming on the PET, then this invaluable guide is for you. More than 30 of the PET's built-in routines are fully detailed so that the reader can immediately put them to good use.

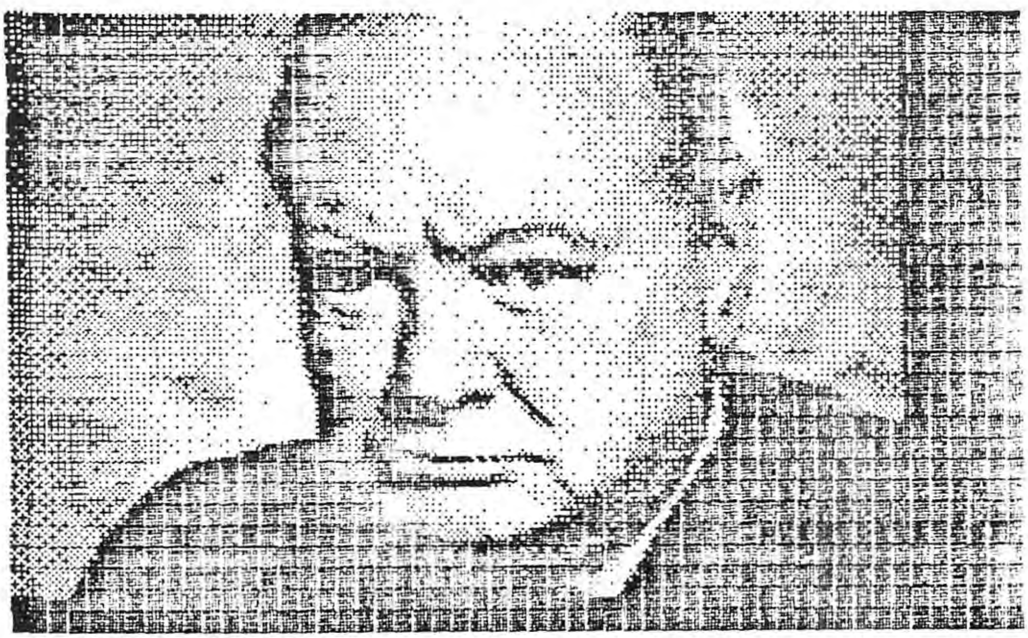
Available for $6.95 + .75 postage. Michigan residents please include 4% state sales tax. VISA and Mastercharge cards accepted - give card number and expiration date. Quantity discounts are available.

ABACUS SOFTWARE
P. O. Box 7211
Grand Rapids, Michigan 49510

www.commodore.ca

```
PLA restore loc'ns 1 and 2
STA VM + 1
PLA
STA VM
PLA ;restore registers
TAY
PLA
TAX
PLP ;restore status
CLI ;interrupts
RTS
```

```
1270  0318 20 D2 FF         JSR  OUTCHAR
1280  031B 20 E7 FF         JSR  CLOSE            ;CLOSE ALL CHANNELS
1290  031E 58               CLI                   ;ALLOW INTERRUPTS
1300  031F 60               RTS
                 ;
                 ;**********************
                 ;* VMDUMP SUBROUTINE *
                 ;**********************
                 ;
                 ; DOES THE TOUGH JOB OF FORMING
                 ;THE SPECIAL CHARACTER MATRIX.
                 ;IT SKIPS OVER SPACES (BUT KEEPS
                 ;TRACK OF THEM). THE ROUTINE
                 ;CLOBBERS A,X,AND Y
                 ;
                 ;!! VMORG SET = $90 !!
                 ;SO SET YOUR OWN IF NECESSARY.
                 ;
1450  0320           VMORG    =   $90               ;VIS.MEM.
1460  0320           VMEND    =   256*VMORG+$1FFF   ;LAST VM LOC'NS USED AS RAM
                 ;
                 ;ZERO PAGE LOC'NS 1 AND 2 USED
1490  0320           VM       =   1                 ;INDIRECT POINTER
                 ;
1510  0320           MATRIX   =   VMEND-9           ;THE CHARACTER MATRIX
1520  0320           BTPT     =   VMEND-3           ;BIT LOCATION
1530  0320           CNTR     =   VMEND-2           ;COUNTS 7 BITS PER BYTE
1540  0320           SPC      =   VMEND-1           ;COUNTS NO.OF SPACES
1550  0320           LINES    =   VMEND             ;COUNTS NO.OF LINES
                 ;
1570  0320 A9 00     VMDUMP   LDA  #0               ;ENTRY POINT
1580  0322 85 01              STA  VM               ;INITIALIZE THE POINTER
1590  0324 A9 90              LDA  #VMORG
1600  0326 85 02              STA  VM+1
1610  0328 A2 09              LDX  #9               ;FETCH INITIAL DATA
1620  032A BD 9E 03 VMD0      LDA  DATA,X
1630  032D 9D F6 AF           STA  MATRIX,X         ;AND STORE IT
1640  0330 CA                 DEX
1650  0331 10 F7              BPL  VMD0             ;ALL 10 BYTES
1660  0333 A2 05     VMD1     LDX  #5               ;6 BYTES TO FORM
1670  0335 AC FD AF  VMD2     LDY  CNTR             ;7 BITS TO ASSIGN
1680  0338 B9 97 03           LDA  OFSTAB,Y         ;OFFSET TO 8K BLOCK
1690  033B A8                 TAY
1700  033C B1 01              LDA  (VM),Y           ;FETCH A BYTE
1710  033E 2C FC AF           BIT  BTPT             ;RETURNS Z=0 IF BIT OFF
1720  0341 18                 CLC
1730  0342 F0 01              BEQ  VMD3
1740  0344 38                 SEC
1750  0345 3E F6 AF  VMD3     ROL  MATRIX,X         ;THE BYTE IS FORMED
1760  0348 CE FD AF           DEC  CNTR             ;DO 7 BITS
1770  034B 10 E8              BPL  VMD2
1780  034D A9 06              LDA  #6               ;RESTORE BIT COUNTER
1790  034F 8D FD AF           STA  CNTR
1800  0352 18                 CLC
1810  0353 6E FC AF           ROR  BTPT             ;ADVANCE 1 BIT RIGHT
1820  0356 D0 09              BNE  VMD4             ;BTPT NOT DONE
1830  0358 6E FC AF           ROR  BTPT             ;RESTORE BTPT =$80
1840  035B E6 01              INC  VM               ;NEXT ADDRESS
1850  035D D0 02              BNE  VMD4             ;BTPT NOT DONE
1860  035F E6 02              INC  VM+1
```



www.commodore.ca

```
1870:  0361 CA       VMD4     DEX                       
1880:  0362 10 D1             BPL  VMD2           ;DO THE 6 BYTES
1890:  0364 A2 05             LDX  #5             ;NOW CHECK FOR ALL ZERO
1900:  0366 18                CLC
1910:  0367 BD F6 AF VMD5     LDA  MATRIX,X
1920:  036A 29 7F             AND  #$7F           ;MASK BIT 7
1930:  036C 9D F6 AF          STA  MATRIX,X
1940:  036F F0 01             BEQ  VMD6           ;SKIP ON ZERO
1950:  0371 38                SEC                 ;A NON-ZERO WAS FOUND
1960:  0372 CA       VMD6     DEX                 ;SKIP ON ZERO
1970:  0373 10 F2             BPL  VMD5           ;A NON-ZERO WAS FOUND
1980:  0375 B0 1F             BCS  VMD8           ;RETURN ON A NON-ZERO BYTE
                     ;
                     ;RE-ENTRY POINT HERE
                     ;
2020:  0377 CE FE AF VMDNEXT  DEC  SPC            ;OTHERWISE IT IS A SPACE
2030:  037A D0 B7             BNE  VMD1           ;SO KEEP GOING
2040:  037C A9 36             LDA  #MAX           ;BUT NOT PAST 54 SPACES
2050:  037E 8D FE AF          STA  SPC            ;RESTORE SPACE COUNTER
2060:  0381 CE FF AF          DEC  LINES          ;DECREMENT LINES COUNTER
2070:  0384 F0 0B             BEQ  VMD7           ;RETURN IF LINES=0
2080:  0386 18                CLC
2090:  0387 A5 01             LDA  VM             ;OTHERWISE GO TO NEXT
2100:  0389 69 F1             ADC  #$F1           ;LINE OF 8K BLOCK
2110:  038B 85 01             STA  VM             ;(6*40+1) LOCATIONS AHEAD
2120:  038D 90 02             BCC  VMD7
2130:  038F E6 02             INC  VM+1
2140:  0391 A9 80    VMD7     LDA  #$80           ;RESTORE BTPT
2150:  0393 8D FC AF          STA  BTPT
2160:  0396 60       VMD8     RTS
2170:  0397 F0 C8 A0 OFSTAB   .BYTE $F0,$C8,$A0,$78 ;OFFSETS TO NEXT LINE
2180:  039B 50 28 00          .BYTE $50,$28,0
2190:  039E 00 00 00 DATA     .BYTE 0,0,0,0,0,0   ;INITIAL DATA
2200:  03A4 80 06 35          .BYTE $80,6,MAX-1,29
                     ;END
```

©

PROMINICO · PROMINICO · PROMINICO · PROMINICO · PROMINICO · PROMINICO · PROMINICO · PROMINICO · PROMINICO · PROMINICO · PROMINICO

TWO POWERFUL PROGRAMS IN ROM FOR YOUR PET/CBM*

XDOS ROM™–$97.50 Every user of Commodore's 2040 Disk has been waiting for XDOS. The maintenance, manipulation and organization of disk files is simplified since XDOS eliminates the repetitious drudgery of disk work. DISPLAY, COPY, SCRATCH, LOAD/RUN operate from a fast two-column Menu display. Multiple files can be selected with single key strokes and then copied or scratched as a batch without further input. Take advantage of our unique risk free refund policy. Once you use XDOS you'll never give it up.

DMENU - Display or print contents of data file.

CMENU - Copy any number of selected files as a batch.

SMENU - Scratch any number of selected files as a batch.

MENU - Load and run a program.

SCREEN PRINT - Special key sequence copies the screen image to your printer.

DIRECT PRINT - Put a "*" in front of any command and it will output to the printer automatically.

DOS SUPPORT - The complete DOS SUPPORT (WEDGE) is included in the XDOS ROM.

SAVE REPLACE - XDOS makes **SAVE** "@ ...and **SAVE** without a drive number completely safe and reliable.

UPGRADEABLE - All Prominico software is supplied in reprogrammable ROM which can be upgraded when required.

MANUAL INCLUDED - Comprehensive instruction manual included.

INDEPENDENT - XDOS uses no RAM, and BASIC is unaffected by its use.

COMPATIBILITY - XDOS is compatible with most other ROM products and can be ordered to fit any of the three available ROM sockets.

SORT ROM™–$97.50 Every serious program can benefit from the five utilities included:

SORT - This command takes a list of array names (string, real and integer in any order or mix) and sorts them based on the alphabetic or numeric order of the first array in the list. An example best illustrates the flexibility of this command: Suppose you wish to maintain an invoice list with the data held in the following arrays:

C%(N) = Customer Number, **I%(N)** = Invoice Number, **A(N)** = $ Amount, **D$(N)** = Date. It is now a simple matter to put this list into order of invoice date, customer number or amount owing. An Accelerated Headsort algorithm with K · N · Log(N) characteristics is used for extremely fast speed even on worst case data.

SORT TIME IN SECONDS

NO. OF RECORDS	1,000	3,000	5,000	10,000
INTEGER	2.6	8.9	15.6	33.0
REAL	4.9	16.7	29.3	–
STRING	3.8	13.3	–	–

READ STRING - This command is a much needed replacement for INPUT# with the following improvements. Maximum input string length increased from 80 to 254 characters. Embedded COMMAS, COLONS and QUOTES are now acceptable data. Null string is returned for empty records.

OPTIMIZED READ, OPTIMIZED WRITE - These two commands drastically simplify and improve data storage on disk. Numerical data is written in binary instead of ASCII, potentially increasing data density by 300%. Data is stored without the need for RETURNS between records thus allowing a string to contain any characters including RETURN, COLON, COMMA and QUOTE. In addition, a list of variable names need only be defined once and not in each read or write statement.

FIND SUBSTRING POSITION - **POS** is a very fast string search function which locates the position of one string within another.

Risk Free Warranty: one year replacement of defective ROM's, if you decide to return the ROM (undamaged) within 30 days we will refund the full purchase price including postage and you may keep the instruction manual with our compliments.

To order use Prominico Direct Mail Response Card in this issue or send: Cheque, Money Order, or Visa/Chargex (include expiry date and signature). Add $2.50 postage for each ROM ordered. Specify which ROM socket you want to fill.

*PET/CBM are registered trademarks of Commodore Business Machines.

PROMINCO LTD., 1921 BURRARD STREET, VANCOUVER, B.C. V6J 3H3. PHONE (604) 738-7811 FOR PROMPT SHIPMENT.

PROMINICO · PROMINICO · PROMINICO · PROMINICO · PROMINICO · PROMINICO · PROMINICO · PROMINICO · PROMINICO · PROMINICO · PROMINICO

www.commodore.ca

Disk Lister

A Disk Cataloging Program for the Commodore Pet and 2040 Disk

Baker Enterprises
15 Windsor Drive,
Atco, New Jersey 08004

Having finally copied all my programs from cassette onto floppy disks, I suddenly found it somewhat difficult to find out where anything was. With well over 300 programs scattered onto 20 or 30 disks, it just wasn't easy to quickly locate a particular program. In addition, I was starting to use Word Pro 3 quite heavily to write articles and various documents, saving them all on disk as well. Because of this, I decided to write a program to catalog all the disks and condense the information onto a single diskette.

The program shown here is the first step toward my final goal. It can catalog well over 100 diskettes with the current Commodore 2040 disk drive. It only has a few functions implemented, but it has proven be very handy. I have a "wish" list of other features I intend to add in the near future. All I need now is the time to do it!

The major flow of the program should be straight forward. I've sprinkled the program with REMarks to help document several operations and a few of the variables used. If you should copy the program, I would strongly recommend leaving out all REMarks and unnecessary spaces to help speed up program execution.

In it's present form, the program reads the directory of any disk placed in drive #1. It then writes a condensed directory as a data file on the master directory disk in drive #0. All of this is done automatically without any user input other than selecting the program function and verifying the correct disk was inserted. Once the data files are created, you can then display or print the directory of any disk that has been cataloged in the master directory. The directory will show the disk name, ID, and format. It will also show an alphabetized list of the files on the disk along with the file type and length (in blocks) of each file. While a directory is being listed, hitting "S" will stop the listing until another key is hit. Hitting "Q" at any time during the listing will terminate the list function. A sample directory printout is shown in Figure 1 to give you an idea of what is displayed.

The file names of the sequential data files created for the master directory consist of the two character disk ID followed by a period and the letters DIR. In its compacted form, the major disk information takes 25 bytes and each entry in the directory takes 20 bytes. Since the disk ID is used to create the data file name, be careful not to duplicate disk ID's. This precaution is also recommended when upgrading to DOS 2.0 since DOS uses the ID to recognize a disk has been changed in the drive. Another hint on using this program - reserve one disk as the master directory disk with nothing else stored on that disk except the directory data files. This will allow cataloging the maximum number of disks into your master directory.

If a cataloged disk is later updated or modified, simply re-catalog the disk to update the master directory. The old data file will be deleted and a new one created, all automatically. The program also provides a delete function, so you can delete a cataloged disk that no longer exists. This function simply deletes the appropriate data file for the specified disk ID. You could actually accomplish the same function by manually scratching the correct data file from the master directory disk.

Currently, when listing or deleting directories, you must enter the two character disk ID. This can be inconvenient at times, but it does make things easier. I intend to allow entering the ID -or- the disk name in the next version I'm working on. However, this will require maintaining some kind of cross-reference to correlate the disk ID's and disk names. When this feature is added, the delete function will always have to be used to remove a disk from the master directory. The added cross-reference will also be the basis for several other new features I intend to add:

- List all disk ID's currently used in alphabetical order; optionally display each disk's corresponding 16 character name. This will help avoid using duplicate disk IDs when creating new disks.
- List all disk names in alphabetical order and show each disk's corresponding 2-character disk ID.
- Ability to list all disks a particular file can be found on. This function should use character matching in case you can't remember the exact file name or want all files starting with a particular word, etc.

One other thing I would like to add is computation of the number of free blocks from the BAM. If this information were included in the data files for each disk, you could then list all disks with the number of free blocks displayed. This would allow quickly finding space on a disk to save a new program of known length.

Right now I'm not sure when I'll be able to get around to finishing this project. At least I've got something useful for now and it does help tremendously. If you have any ideas or suggestions as to other features you think might be useful, or if you're interested in how the final version turns out, let me know.

www.commodore.ca

ELECTRONIC TECHNOLOGY CORPORATION

BATTERY BACKUP FOR PET/CBM

BACKPACK™

Designed to install within the CPU Case, **BACKPACK** gives 6 to 10 minutes of full power to the computer (32K, CRT, Tape Drive) during times of power shortages and gliches in power lines thus allowing the user time enough to save data or programs to tape (or disk with **FLOPPY BACKPACK**). Batteries are recharged from CPU's own power supply and **BACKPACK** requires no wiring changes to install.

Single Quanity $225.00

FLOPPY BACKPACK™

FLOPPY BACKPACK is a battery backup system for the Commodore Dual Floppy Disk Drive. Used in conjunction with **BACKPACK** for the pet, the user can save data and programs during power shortages. It also eliminates possible disk crashes. Designed to install within the disk cabinet with no wiring changes, the batteries are recharged from the disk's own power supply.

Single Quanity $135.00

AIM-65 POWER SUPPLY (with battery backup) & ENCLOSURE

AIM POWER 4™

AIM POWER 4 is a total power supply with rechargeable battery backup for the Rockwell AIM 65. **AIM POWER 4** supplies full voltage output for the AIM under normal operating mode and with batteries can keep the AIM up and running for one hour. Batteries are recharged during normal operation. **AIM POWER 4** is on a single board and will mount inside the **AME 1** enclosure (from ETC), and Enclosure Group Products.

Single Quanity $250.00

AME 1™

AME 1 is a totally metal enclosure of lightweight aluminum construction. Designed to house the AIM 65 and a power supply, with easy access to switches and connectors. **AME 1** will have an integral reset button, easy paper feed and totally visible 20 character LED readout. Designed for industrial applications, but has professional appearance.

Single Quanity $100.00

FOR MORE INFORMATION OR TO ORDER BY PHONE: (919) 362-4200 OR (919) 362-5671
INCLUDE $5.00 FOR SHIPPING AND HANDLING • VISA AND MASTERCHARGE ACCEPTED
DEALER INQUIRIES INVITED

All Products are shipped fully assembled and Ready to Install • Instructions Included.

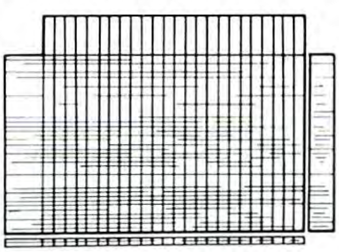
P.O. Box G, Apex, North Carolina 27502 (919) 362-4200

Etcetera International Software

COLUPAD™ THE ELECTRONIC COLUMNAR PAD

A Problem Solving Computer Software Program

COLUPAD™ is a unique computer tool which allows you to create, store, selectively retrieve and perform math operations upon data and then generate reports using that data. **COLUPAD™** can best be visualized as the typical columnar accounting pad found in most offices. Powerful, but simple to use, **COLUPAD™** has such features as:



- 24 columns and 40 rows for data manipulation
- Space for row and column descriptions
- Ability to reference decimal places for each column
- Ability to perform math functions on one column and store in another
- Ability to add, subtract, multiply, etc., one column to another and then place resulting data in a different column
- Print a whole page, selective columns or rows
- Save whole page to disk
  Retrieve whole pages, selective rows, or columns from disk
- Plus much, much more

One page of the pad, the worksheet, is in computer memory. The pad consists of worksheets stored on disk and linked by name. Pad size is only limited to diskette capacity.
If you find yourself spending hours (or days) copying, adding, subtracting, or whatever one column by another or just compiling data on a columnar pad, then **COLUPAD™** can save you time, and in your business, time is money. **COLUPAD™** requires a 32K PET, Disk, and printer. – $150.00

FOR MORE INFORMATION OR TO ORDER BY PHONE: (919) 362-4200 OR (919) 362-5671
INCLUDE $1.50 FOR SHIPPING AND HANDLING • VISA AND MASTERCHARGE ACCEPTED
DEALER INQUIRIES INVITED

P.O. BOX G, APEX, NORTH CAROLINA 27502 (919) 362-4200

www.commodore.ca

```
10 REM ******* DISK LISTER ************
20 REM
30 REM     BY: ROBERT W. BAKER
40 REM
50 REM 15 WINDSOR DRIVE, ATCO, NJ 08004
60 REM
70 REM ********************************
80 :
90 CLR:DIM D$(150),D(150):Q$=CHR$(34):
     ¬CR$=CHR$(13)
100 REM DISPLAY MENU & SELECT FUNCTION
110 PRINT"ĥ";SPC(9);"r̲D I S K   L I S T ¬
     ¬E R":GOSUB 1340
120 PRINT SPC(5);"0 - DONE↓
130 PRINT SPC(5);"1 - UPDATE MASTER ¬
     ¬DIRECTORY↓
140 PRINT SPC(5);"2 - DISPLAY SELECTED ¬
     ¬DIRECTORY↓
150 PRINT SPC(5);"3 - DELETE DISK ENTRY ¬
     ¬FROM MASTER
160 GOSUB 1340
170 PRINT"↓ENTER DESIRED FUNCTION:  ";
180 GOSUB 1320
190 IF C$="0" THEN PRINT"ĥ": END
200 C=VAL(C$):IF C<1 OR C>3 THEN 180
210 ON C GOTO 250,750,1050
220 REM ************************
230 REM UPDATE MASTER DIRECTORY
240 REM ************************
250 PRINT"ĥINSERT UPDATE DISK IN DRIVE ¬
     ¬#1
260 GOSUB 1310:GOSUB 1340:PRINT"OK
270 OPEN 15,8,15
280 PRINT#15,"I1"
290 OPEN 5,8,5,"$1,S,R":GOSUB 1260
300 Y=142:GOSUB 1200:REM *** SKIP BAM
310 Y=16:GOSUB 1180:DN$=S$:REM *** DISK ¬
     ¬NAME
320 Y=2:GOSUB 1200:REM *** SKIP SPACES
330 Y=2:GOSUB 1180:DI$=S$:REM *** DISK ¬
     ¬ID
340 PRINT"ĥr̲DISK NAME:r̂  ";DN$ :
     ¬ PRINT"↓r̲DISK  ID:r̂   "DI$:
     ¬GOSUB 1340
350 PRINT"CORRECT DISK INSERTED"; :
     ¬ GOSUB 1350:IF C$="N" THEN 710
360 GOSUB 1340:PRINT"READING DIRECTORY ¬
     ¬ENTRIES...
370 GOSUB 1250
380 Y=2:GOSUB 1180:DF$=S$:REM *** DISK ¬
     ¬FORMAT
390 Y=89:GOSUB 1200:NF=0:Z=0:REM *** ¬
     ¬SKIP TO FIRST DIRECTORY ENTRY
400 GOSUB 1220:FT=V:F$=C$:REM *** FILE ¬
     ¬TYPE (0=DELETED)
410 Y=2:GOSUB 1200:REM *** SKIP ¬
     ¬STARTING TRACK & SECTOR
420 Y=16:GOSUB 1180:REM *** FILE NAME
430 Y=9:GOSUB 1200:REM *** SKIP UNUSED ¬
     ¬INFO
440 GOSUB 1220:X=V:GOSUB 1220:X=X+(V*256
     ¬):REM *** #BLOCKS IN FILE
450 IF FT>0 THEN NF=NF+1:D$(NF)=F$+S$:
     ¬D(NF)=X:REM *** ADD FILE IF NOT ¬
     ¬DELETED
460 Z=Z+1:Z=Z-(INT(Z/8)*8):REM *** ¬
     ¬Z=ENTRY WITHIN THIS DISK BLOCK
470 IF Z>0 THEN Y=2:GOSUB 1200:REM ** ¬
     ¬SKIP 2 BYTES IF NOT LAST ENTRY IN ¬
     ¬BLOCK
480 IF SS=0 THEN 400:REM *** CONTINUE ¬
     ¬TILL END OF DIRECTORY
490 CLOSE 5:IF NF<2 THEN 600
500 GOSUB 1340
510 PRINT"SORTING DIRECTORY ENTRIES...
520 REM SORT DIRECTORY INTO
530 REM ALPHABETICAL ORDER
540 FOR X=1 TO NF:FOR Y=1 TO NF-1
550 IF D$(Y)<=D$(Y+1) THEN 570
560 C$=D$(Y):C=D(Y):D$(Y)=D$(Y+1):
     ¬D(Y)=D(Y+1):D$(Y+1)=C$:D(Y+1)=C
570 NEXT Y,X
580 REM DELETE OLD DIRECTORY
590 REM DATA FILE & SAVE NEW COPY
600 GOSUB 1340:PRINT"UPDATING MASTER ¬
     ¬DIRECTORY...
610 S$="0:"+DI$+".DIR"
620 PRINT#15,"S"+S$
630 OPEN 5,8,5,S$+",S,W":GOSUB 1260
640 PRINT#5,Q$;DN$;Q$;CR$;:GOSUB 1260
650 PRINT#5,DI$;CR$;:GOSUB 1260
660 PRINT#5,DF$;CR$;:GOSUB 1260
670 IF NF=0 THEN 710
680 FOR X=1 TO NF:FOR Y=1 TO 17:
     ¬PRINT#5,MID$(D$(X),Y,1);:GOSUB ¬
     ¬1260:NEXT Y
690 H=INT(D(X)/256):L=D(X)-(256*H)
700 PRINT#5,CHR$(L);CHR$(H);CR$;:
     ¬GOSUB 1260:NEXT X
710 CLOSE 5:CLOSE 15:GOTO 110
720 REM ******************************
730 REM DISPLAY SELECTED DISK DIRECTORY
740 REM ******************************
750 PRINT"ĥTO DISPLAY DISK DIRECTORY":
     ¬GOSUB 1140:OPEN 15,8,15
760 OPEN 5,8,5,S$+",S,R":GOSUB 1260
770 GOSUB 1340:PRINT"WANT PRINTED ¬
     ¬COPY";:GOSUB 1350:GOSUB 1340
780 PD=3:IF C$="Y" THEN PD=4
790 OPEN 4,PD:REM *** PD = PRINT DEVICE ¬
     ¬SELECTOR (3=DISPLAY, 4=PRINTER)
800 INPUT#5,DN$:GOSUB 1260
810 INPUT#5,DI$:GOSUB 1260
820 INPUT#5,DF$:GOSUB 1260
830 IF PD=3 THEN PRINT"ĥ";
840 PRINT#4,"r̲DISK NAME:r̂  ";DN$
850 PRINT#4
860 PRINT#4,"r̲DISK ID:r̂  ";DI$;SPC(10);"
     ¬r̲DISK FORMAT:r̂  ";DF$
870 PRINT#4 : REM *** DISK FORMAT WILL ¬
     ¬BE BLANK FOR DOS 1.0
880 PRINT#4,"CCCCCCCCCCCCCCCCCCCCCCCCCCC
     ¬CCCCCCCCCCC":PRINT#4
890 Y=17:GOSUB 1180:REM *** GET FILE ¬
     ¬NAME & TYPE
900 GOSUB 1220:Z=V:GOSUB 1220:Z=Z+(256*V
     ¬):REM *** GET #BLOCKS
910 GOSUB 1250:REM *** SKIP LAST CR
920 PRINT#4,RIGHT$("    "+STR$(Z),
     ¬4);"  ";
930 PRINT#4,MID$(S$,2,16);SPC(3);
940 V=ASC(LEFT$(S$,1)):REM *** DECODE ¬
     ¬FILE TYPE
950 IF V=129 THEN PRINT#4,"SEQ";
```

www.commodore.ca

Computer House Div.

Programs for Commodore & Apple

"Legal Accounting" $1200.00
"Political Party Mailing List" 150.00

ENGINEERING & MACHINE SHOP

"Machine Part Quoting" $280.00
"Trig & Circle Tangent" 70.00
"Bolt Circle" 25.00
"Spur Gears" 35.00
"Beams; Stress & Deflection" 145.00
"Tank Thickness"
For Filament Winding 85.00

All 6 for only $495.00

"SCRUNCH" — $36.00

For Apple II or Apple II Plus. Compacts Basic Programs up to 20%.

Dealer inquiries invited

COMPUTER HOUSE DIV.
1407 Clinton Road
Jackson, Michigan 49202
Phone: (517) 782-2132

Computer House Div.

Programs for Commodore Computers

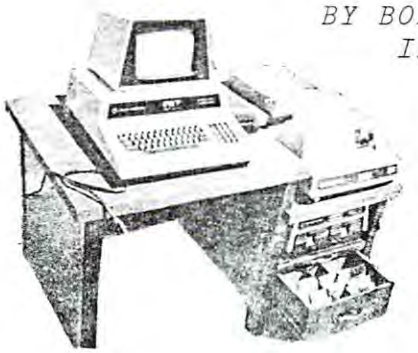
1 - F.E.T.-Recover; File Editing Tools, Adds 11 commands to assist disk recovery $65.00
2 - SOF-BKUP; copy disks faster including random files. Displays error messages for bad blocks. $40.00
3 - SUPER-RAM; checks every ram address against every other ram address, 2001 series $20.00
4 - VARI-PRINT; prints listing of all variables with every line number where each occurs $25.00
5 - DOCU-PRINT; prints CRT to printer, use in your own program, in basic $20.00
6 - SCREEN DUMP/REPEAT; similar to DOCU-PRINT except in machine language. Use with anyones program, REPEAT similar to Model 8032 ...... $35.00
7 - TRACE-PRINT; prints listing of all line numbers in order of execution $25.00
8 - SCRUNCH-PLUS; packs a copy of program to save up to 25% of memory space. $25.00
9- -SORTER; sorts a one dimensional array alphabetically (in machine language). $35.00

PACKAGE PRICE $170.00 FOR ALL 9 PROGRAMS
SAVE $120.00

Dealer inquiries invited

COMPUTER HOUSE DIV.
1407 Clinton Road
Jackson, Michigan 49202
Phone: (517) 782-2132

BY BOLIVAR INDUSTRIES!



the original Computer Systems Desk for The Commodore "Pet".

Some features are:

* Independently switched outlets for easier on/off control of equipment.
* Hide away paper supply compartment.
* File drawer holds up to 200 5 inch floppy diskettes plus 17 cassettes.
* Available in various sizes to fit your space and need.
* Matching office desks and file cabinets available.
* Available in Walnut or Fruitwood finishes.

We also sell:

Business software:

Mailing List Package--Read and write lists to the 2040 disk.

Check File Package--Read and write to 2040 disk drive your check entries. Categorizes deductable entries.

Inventory Package--Maintains an inventory file on items in stock.

Business forms:

In quantities as low as 50, invoices and statements available. These are continuous pin feed forms.

DEALER INQUIRIES ARE INVITED!

For your free broshures, write:

BOLIVAR INDUSTRIES
861 S. Drew st.
MESA, ARIZONA 85202
Ph. (602) 969-9689

www.commodore.ca

```
960 IF V=130 THEN PRINT#4,"PGM";
970 IF V=131 THEN PRINT#4,"USR";
980 PRINT#4:GET C$:IF C$="S" THEN GOSUB ¬
    ¬1320:REM *** ALLOW START/STOP OF ¬
    ¬LIST
990 IF C$<>"Q" AND SS=0 THEN 890
1000 CLOSE 4:CLOSE 5:CLOSE 15:IF PD=3 ¬
    ¬THEN GOTO 1300
1010 GOTO 110
1020 REM ********************************
1030 REM DELETE DISK DIRECTORY DATA FILE
1040 REM ********************************
1050 PRINT"ĥTO DELETE DISK FROM MASTER ¬
    ¬DIRECTORY":GOSUB 1140:OPEN 15,8,15
1060 PRINT#15,"S"+S$:CLOSE 15:GOTO 110
1070 :
1080 REM ***********************
1090 REM ***** SUBROUTINES *****
1100 REM ***********************
1110 :
1120 REM *** GET DISK ID
1130 REM *** & MAKE DATA FILE NAME
1140 INPUT"↓ENTER DISK ID   _<<<";DI$
1150 IF DI$="_" THEN 110
1160 S$="0:"+LEFT$(DI$,2)+".DIR":RETURN
1170 REM *** READ STRING FROM DISK,
    ¬ Y-BYTES LONG
1180 S$="":FOR X=1 TO Y:GOSUB 1250:
    ¬S$=S$+C$:NEXT X:RETURN
1190 REM *** SKIP Y-BYTES OF DISK FILE
1200 FOR X=1 TO Y:GOSUB 1250:NEXT X:
    ¬RETURN
1210 REM *** READ BYTE & RETURN ASC ¬
    ¬VALUE
1220 V=0:GOSUB 1250:IF C$<>"" THEN ¬
    ¬V=ASC(C$)
1230 RETURN
1240 REM *** GET BYTE & CHK FOR DISK ¬
    ¬ERROR
1250 GET#5,C$:SS=ST
1260 INPUT#15,EN,EM$,ET,ES:IF EN=0 THEN ¬
    ¬RETURN
1270 PRINT"ĥrDISK ERROR!↓
1280 PRINT EN;EM$;ET;ES:CLOSE 4:CLOSE 5:
    ¬CLOSE 15
1290 REM *** MISC ROUTINES ***
1300 GOSUB 1340:GOTO 110
1310 PRINT"↓DEPRESS ANY KEY TO CONTINUE
1320 GET C$:IF C$="" THEN 1320
1330 RETURN
1340 PRINT"↓@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    ¬@@@@@@@@@@@@":RETURN
1350 PRINT" (Y/N) ?  ";
1360 GOSUB 1320:IF C$<>"Y" AND C$<>"N" ¬
    ¬THEN 1360
1370 PRINT C$ : RETURN
```

©

MICRO COMPUTER INDUSTRIES, LTD.

INVENTORY CONTROL WITH POINT OF SALE FOR CBM AND PET 32K

DISK VERSION INCLUDES: Storage of 2500 items per diskette, Accounts Receivable, Writes Purchase Orders, Invoices, Summaries, Post Income and will Sort by 10 fields.

$100.00

INVENTORY CONTROL 8K

CASSETTE VERSION INCLUDES: Purchase Order program and Printing functions.

$ 39.00

GENERAL LEDGER

DISK VERSION INCLUDES: The total functions of the Inventory Control programs plus; Accounts Payable, Notes Payable, Purchases, Expenses, it also issues complete Reports, Statements and Summaries.

$350.00

Instruction Manual $10.00 refundable with purchase.

MICRO COMPUTER INDUSTRIES, LTD.

1520 East Mulberry Suite 240
Fort Collins, Colorado 80524
1-303-221-1955

COPYRIGHT 1980

www.commodore.ca

Commodore Dealers Form Cooperative

Joretta Klepfer
COMPUTE Staff

Realizing that software availability is frequently a critical factor in selling computers to businesses, a number of Commodore dealers have formed an association which will seek (or develop), evaluate, and distribute high quality business software. The Dealer Information Systems Cooperative (DISC) is the result of several months of planning by a group of CBM dealers in the Southeast. The organization will provide members with a source for business software and various computer peripherals as well as an excellent channel for communication.

A meeting was held September 12 in Apex, NC to organize formally, i.e. elect officers and board members, establish working committees, determine membership criteria, and decide on the type of organization. Alex Amor of Creative Equipment, Miami, FL was elected president. The organizers accepted the offer extended by Commodore Business Machines, Decatur, GA, (under the direction of Jerry Zeigler) to provide temporary office space and staff support for DISC. While DISC projects its own staff of programmers for the purpose of developing CBM business software, the organization expects to rely heavily on individuals and small companies who wish to market their products through the association. Although major emphasis will be on business software, new peripheral hardware will be solicited for evaluation and possible distribution by DISC. Other services to members will include a data bank of information about software currently available, information exchange through written newsletters and electronic mail (either public or private), seminars, and exhibitions.

Attendees at the meeting on Sept. 12 had an opportunity to review two software packages being offered to the association for evaluation and distribution. The Business Information System from Small Business Development Corporation is a comprehensive management system which functions interactively on a CBM 2001. This single-input program will automatically make multiple entries and then transfer information to a temporary transaction disk as security against data loss. At the end of the day's processing, permanent disk files are updated from the temporary disk. The program components include cash and sales receipts, expenses & cash disbursements, accounts receivable (multiple categories), accounts payable, inventory, payroll, loans, equipment, general ledger, and financial reports. A second demonstration involved the use of a PET as a cash register. The CE-1000 package from Creative Equipment includes a bar code reader, cash drawer and the software to allow a PET to become an efficient point of sale device which will provide complete inventory management and detailed receipts for the customer. The demonstration was quite impressive - the bar code reader would operate accurately in any direction. These two software packages alone could provide an excellent beginning for the new organization.

Although the charter members come from the Southeast, DISC was organized with the goal of being national and international in scope with each of the geographic regions of Commodore, including Canada and Commodore International possibly represented on the Board of Directors. Memberships are available to authorized dealers, software companies, and manufacturers for $250.00 through the end of this year. On January 1, 1981 the memberhsip fee increases to $500.00. Information about Dealers Information Systems Cooperative may be obtained from Membership Committee chairman Bob West, c/o Bob West Publications, P.O. Box 170, Sapphire, NC 28774. ©

Programmers:

Send your short and simple BASIC utility programs with a brief descriptive note to:

COMPUTE!
P.O. Box 5406
Greensboro, NC 27403 USA
Attn: Basically Useful BASIC

www.commodore.ca

Nuts and Volts

Gene Zumchak

In earlier columns I discussed the timing required when a processor talks to devices on its own intimate bus. A number of readers have asked that I say a few words about timing between a computer and peripherals, or between two computer systems. In particular, some readers have more than one computer system, and would like to pass information between them. One computer might be freed of a time consuming task by quickly downloading data to another computer, and letting it perform the time consuming task. Another application would be to put together a system for a classroom where the main computer and its disk and printer is shared by a number of users, each having a low-cost terminal. Such "time-sharing" techniques have been used for some time on large computer systems.

There are two basic methods for communicating between two devices, serial and parallel. Each type has advantages and disadvantages, and both should be considered for any particular application. Within each type are a variety of formats, both synchronous and asynchronous. In this column we will begin by discussing parallel communication.

Parallel Techniques

The primary advantage of parallel communication is speed. Clearly, sending data eight bits at a time will be faster than sending data a bit at a time. The price paid for the speed is the number of wires that must be used, for eight bits, at least nine wires plus a ground return. If the separation between devices is more than a few feet, separate return wires may be required for each data wire for reliable data transfer. When the cost of connectors is considered, high speed serial communication using a single coax may be more economical.

In the discussions that follow, we will assume that the parallel data words being transmitted are eight bits wide. The same techniques apply to data words of any width. Communication can be uni-directional or bi-directional. For the latter, separate sets of wires may be used for communication in each direction, or a single, bi-directional set used. The newer programmable port chips, available in most families, are quite conveniently used for bi-directional communication over the same data lines. Since so many 6502 users have boards with 6522 port chips, we will use the 6522 in the examples. If you need uni-directional communication, and you are putting the hardware together from scratch, TTL latches will be more economical for driving lines, and tri-state gates more economical for receiving than using port chips. On the other hand, the extra control lines associated with the ports may save additional chips and make the port chip still attractive.

Although parallel communication could be synchronous or asynchronous it is generally neither. Synchronous communication would require a clock line to clock data into the receiving device. An asynchronous technique would provide self-clocking parallel data with no feedback from the receiving device. Synchronous transmission imposes severe requirements on the receiving hardware. Asynchronous transmission sends data out at a fixed rate. Oftentime the receiving device takes more time to process some data characters than others, so an asynchronous method has to accomodate the worst case response and will in some cases slow down the computer unnecessarily. Consequently, a quasi-synchronous technique called "handshaking" is generally used. When the computer sends data to a peripheral, "write" handshaking is used. When the computer receives data from a peripheral, "read" handshaking is used. There really is no fundamental difference between the two types. Clearly what is "write" handshaking for the computer, is "read" handshaking for the peripheral.

Write Handshaking

Let's consider an example of "write" handshaking. A common peripheral is a printer with a parallel interface. Ordinarily, the printer can take characters as fast as it can print them, however, a carriage return may take a considerably longer time. Handshaking solves the problem of differing response times. The computer first presents the data to the port. It then generates a handshake signal called variously DATA STROBE, DATA AVAILABLE, or DATA READY. If the reading device needs a strobe type signal, this signal may be a pulse, otherwise it may be a level. The reading device (printer) responds by raising a flag called BUSY or in other cases, DATA TAKEN. If the DA (DATA AVAILABLE) signal is a level, the computer may respond to the BUSY by removing the DATA AVAILABLE. The printer will keep the BUSY line high for as long as it takes to process the character. If the computer is not otherwise busy, it will continue to poll the BUSY flag until it goes false. Then the computer will be free to write another character to the port and generate another DA signal. The computer must return from the print character routine in order to go back to the program that is generating the characters. It could be a hundred microseconds or more before the new character is ready. A more time efficient approach would be to exit the routine as soon as BUSY goes true and generate the character while the printer is busy. The print routine must then do its waiting at the beginning. It first checks to see that BUSY is false before sending a character. If the computer has a sophisticated operating system, it may be able to perform other tasks while the printer is processing

www.commodore.ca

Low Cost Graphics Are Now Here

Add Printable High Resolution (320 x 200) Graphics To Your AIM-65 Microcomputer

Visible Memory Board It's a reality. The Visible Memory and graphic print software from MTU are now available for the Rockwell AIM-65. The Visible Memory gives a high resolution 320 wide by 200 high bit mapped pixel display matrix. Each dot is individually addressable for maximum utilization and speed. Thus characters, image shapes, and graphs can all be displayed separately or simultaneously if needed — maximum flexibility for you.

The Visible Memory is just that, an 8K byte RAM board that contains 2 access ports to the memory matrix. The microprocessor bus uses one port and the display refresh circuitry uses the second port. The contents of the memory bit-for-bit is precisely what is displayed. If you need 8K of RAM for a non-display application, use it! It makes no difference to the board what its contents are; program (seen in its binary pattern form) or a human recognizable display pattern. The display refresh occurs at times when the processor never goes to memory. Therefore there is no snow on the display and no wait states for the processor.

Hardcopy Too In addition MTU has engineered a software package to drive the AIM-65 printer in new ways. Three new forms of printing are possible. QUICKPRINT gives a matrix 200 across by 320 up the 2¼" wide paper. QUALITYPRINT gives two prints, each 100 x 320 which gives a higher quality (4½" wide) printed area when placed side by side. TEXTPRINT allows you to print the AIM text buffer area of memory as 10 rows of characters printed "up" the paper strip. You may specify up to 127 characters per row for the row length. The QUICK and QUALITY print modes are designed to give you fast, easy hardcopy of the Visible Memory contents. Thus you now have a graphic computer with hardcopy capability.

Graphic/Text Software Drivers To allow you to easily use this graphic display and print power, MTU has also designed the K-1008-5C software package which gives you point plotting, line drawing, character generation and a host of other subroutines. Written in assembly language, these routines may be executed from BASIC or assembly language — your choice. Text output from BASIC or the AIM monitor may also be shown on the Visible Memory display as up to 22 lines by 53 characters per line significantly enhacing the use of the AIM-65 as a computer with a CRT display.

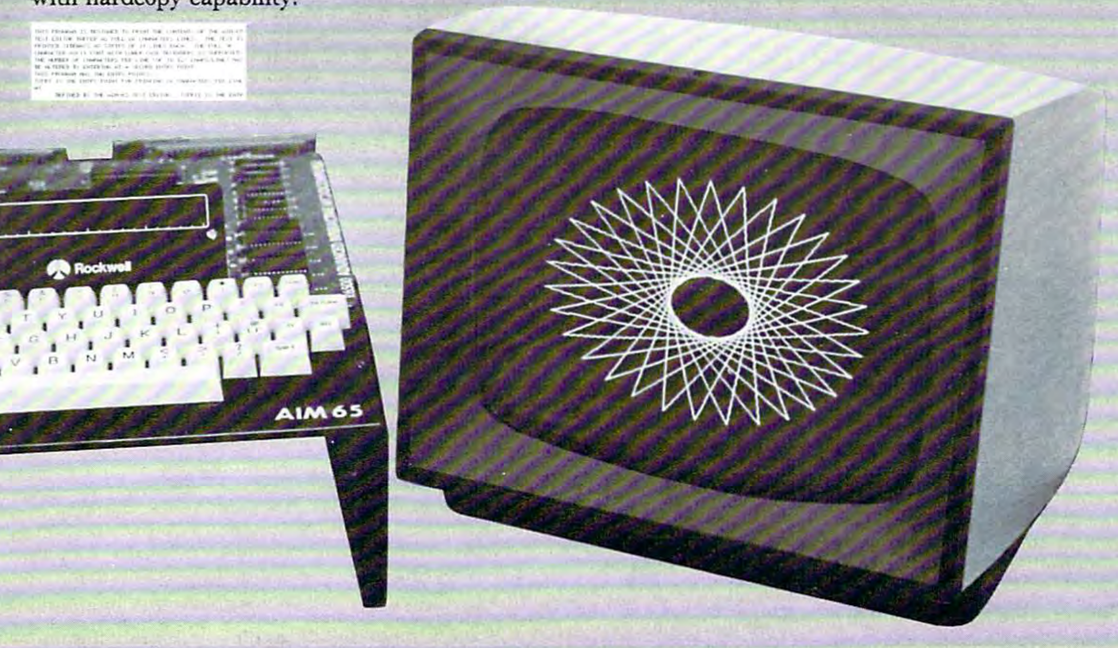
Call Us Now Many educators have been waiting for this type of price/performance to set up courses. MTU will be pleased to quote quantity purchases — call us direct — now. Demand is high and Fall is just around the corner.

K-1008 Visible Memory	$240
K-1008-5C Graphic/text software	$25
K-1009-1C AIM Print software	$25
K-1000-5 AIM-65 Power Supply	$65
K-1005-A AIM-65 Card File	$85
Many others not listed	

Call or write for our catalog listing all our AIM-65 (also our PET, KIM-1, and SYM-1) products.

As of June 1, 1980 place orders at: Micro Technology Unlimited, P.O. Box 12106, 2806 Hillsborough Street, Raleigh, North Carolina 27605

MTU
Micro Technology Unlimited
P.O. Box 12106
2806 Hillsborough Street
Raleigh, N.C. 27605
(919) 833-1458



www.commodore.ca

the character. In that case, the computer could use the falling edge of the BUSY flag to generate an interrupt. There are few microcomputer systems, however, which can support this type of response. For example, if the printer is being used to print out an assembly listing, the computer must wait for the printer before continuing the assembly process.

Figure 1. shows one method of connecting a 6522 port using write handshaking with a parallel type printer. Figure 2., taken from a 6522 spec sheet, shows write handshaking timing for either port. We must now resolve the characteristics of the port with requirements for the printer. First note that the functions of the control pins CA1, CA2, CB1, and CB2 are various, and primarily defined by bit groups in the PCR (peripheral control register). Figure 3., taken from the spec sheet, defines the functions of CA1 and CA2 as a function of PCR0 - PCR3. Let us now look at the timing in detail. The DA signal from CA2 is low going. That will satisfy the low going DATA STROBE of the printer. When CA2 is programmed for output handshake, it will automatically go low when the data is written to Port A. That's good, right? Not necessarily. It is very likely that the printer uses the low going edge of the DATA STROBE to latch the data. If the printer has a data setup time of 300 nanoseconds or more, we are in trouble. That is, the data should be sitting for awhile when the strobe occurs. We can resolve this difficulty with the following sequence:

1. Program CA2 for no handshake. (PCR = XXXX000X)
2. Write Data to Port A.
3. Program CA2 for handshake. (PCR = XXXX1001)
4. Read Port A. (Note that handshake is generated with read or write of port.)

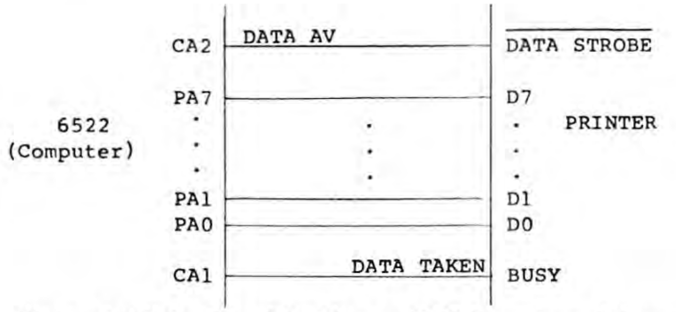
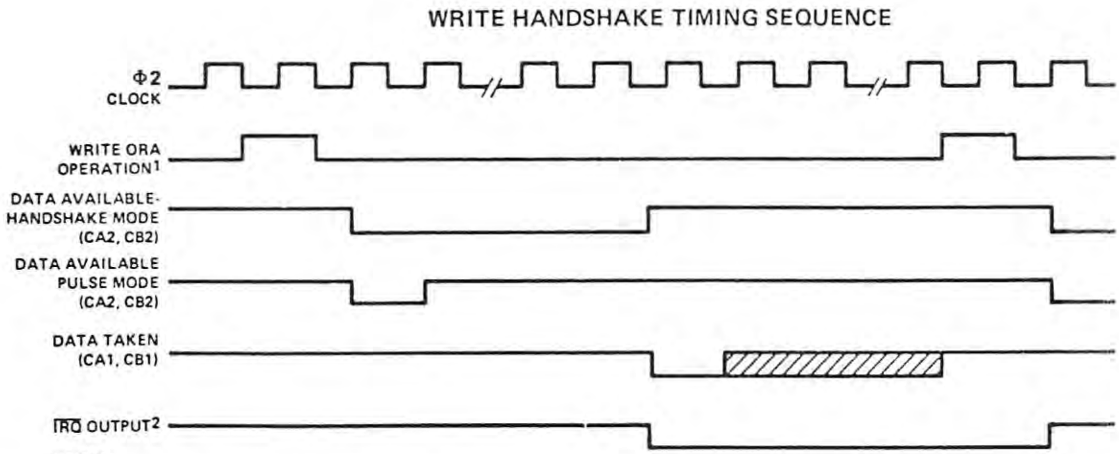


Figure 1. Write Handshaking - 6522 Port A to Printer

The printer responds to the DA signal with BUSY. Note that in Figure 2., the Data Taken Signal is a low going signal, while BUSY is high going. Do we need to invert BUSY? If we read the entry in Figure 3. which describes programming CA2 for handshaking we will note that CA2 is reset high "with an active transistion on CA1". Thus we must program PCR0 to be a one to respond to a positive transition on CA1. This was indicated in line 3 of the procedure above. Are we done? Not yet. We now must wait until BUSY goes false before we can send another character. The "write handshaking" of Figure 2. makes no allowance for this. We must poll BUSY, but the level of CA1 cannot be read. Note, however, that a transition on CA1 sets a flag in the IFR (Interrupt Flag Register), and we may poll this bit in this register. The following sequence should work, assuming that we have just detected the leading edge of BUSY.

1. Read Port A to reset CA1 flag in register.
2. Program CA1 for negative transition. (PCR = XXXX0000)
3. Poll CA1 flag in IFR. (Bit 1)
4. Read Port A to clear flag.



NOTES:
1. R/W = 0, $\overline{CS2}$ = 0, CS1 = 1, RS3 = 0, RS2 = 0, RS1 = 0, RS0 = 1.
2. Signals "data taken" to the system processor.

Figure 2. 6522 Write Handshaking Timing

www.commodore.ca

EXCERT, INCORPORATED

*** AIM-65 ***

P/N		QTY 1-9
A65-1	AIM-65 w/1K RAM	$375
A65-4	AIM-65 w/4K RAM	$420
A65-A	Assembler ROM	$ 85
A65-B	BASIC ROM	$100
A65PL	PL/165 ROMS	$125

ACCESSORIES

P/N — **QTY 1-9**

Power Supplies (Fully AIM-65 Compatible, Industrial Quality Open Frame)

2PRS3	+5V at 3A, +24V at 1A w/mtg hardware, cord, etc.	$ 65
PRS4	+5V at 2A, +24V at .5A w/mtg hardware, cord, etc.	$ 50
PRS5	+5V at 2A, +24V ±15% at .5A ±12V to ± 15V at .4A	$ 75

From The Enclosure Group

ENC1	AIM-65 case	$ 45
ENC1A	AIM-65 case w/space for one expansion board	$ 49

Cases with Power Supplies

ENC3	ENC1 w/PRS3 mounted inside	$115
ENC3A	ENC1A w/PRS3 mounted inside	$119
ENC4	ENC1 w/PRS4 mounted inside	$100
ENC4A	ENC1A w/PRS4 mounted inside	$104
ENC5	ENC1 W/PRS5 mounted inside	$125
ENC5A	ENC1A w/PRS5 mounted inside	$129

From The Computerist, Inc.

MCP1	Mother Plus™ Dual 44 pin mother card & card cage takes MEB4, VIB1, PTC1, fully buffered, 5 expansion slots underneath the AIM	$100
MEB4	DRAM Plus™16K RAM, 16K Prom sockets, 2-6522 I/O chip and programmer for 5V EPROMS	**16K RAM $295** **32K RAM $395**
PTC1	Proto Plus™Prototype card same size as KIM-1, MEB4, BIB1	$45
VIB1	Video Plus™board with 128 char, 128 user char, up to 4K display RAM, light pen and ASCII keyboard interface	$245
CABLE	For MEB4, VIB1, PTC1	$ 15

From Optimal Technology

ADC1	A/D eight channels, D/A 2 channels. Requires ±512V to 8'15V at 100MA & 2-I/O Ports from AIM-6522	$115
	Cable	$ 25

SPECIAL

P/N		QTY 1-9
A65-4AB	AIM-65 w/4K RAM Assembler and BASIC	$595
A65-4B	AIM-65 w/4K RAM, BASIC	$310
S/A65-1	AIM-65 w/1K RAM, w/o PRINTER, Display, Keybd, User 6522	$199

PIN — **QTY 1-9**

From Seawell Marketing, Inc.

MCP2	Little Buffered Mother™Single 44 pin (KIM-4 style) mother card takes MEB2,PGR2, PTC2 and PI02. Has on board 5V regulator for AIM-65, 4 expansion slots. Routes A&E signals to duplicates on sides with 4K RAM	$199
MEB2	SEA 16™16K static RAM board takes 2114L with regulators and address switches	$280
PGR2	Programmer for 5V EPROMS with ROM firmwave, regulators, low force sockets, up to 8 EPROMS simultanously, can execute after programming	$299
P102	Parallel I/O board with 4-6522's	$260
PTC2	Proto/Blank™Prototype card that fits MCP2	$ 49
PTC2A	Proto/Pop™with regulator, decoders, switches	$ 99

From MTU

DAC3	8 bit DAC Board	$ 49
FDC3	Floppy disk controller bd. & DOS, up to four 5¼" or 8" drives, double sided, double density	$595
MCP3	Card file w/4 slot expansion mother bd. w/keybd. brackets	$ 85
MEB3A	16K DRAM Board, low power	$298
PIO3	24K PROM, 4-8 bit I/O ports w/RS-232 port to 4800 bps. PROM Programmer	$295
VIB3	8K DRAM Bd. low power w/composite video out in 200 lines 320 dot/line format	$240

All MTU Software Available for these Products.

Miscellaneous

TPT2	Approved Thermal Paper Tape 5/165' rolls	$ 10
MEM6	6/2114 RAM Chips	$ 45
CAS1	Available Audio Cassette Recorder	$ 40
2716	16K5V EPROM	$ 20
A65-P	Printer	$ 75
A65-DM	Display Module	$ 30

All AIM-65 Spare Parts Are Available.

ASSEMBLED & TESTED SYSTEMS

We specialize in assembled and tested systems made from the above items. Normally, the price will be the total of the items, plus $5 for handling, shipping is extra on all C.O.D.'s or invoiced orders. Warranty on all systems.

Higher quantities quoted upon request.
COD's accepted, shipping will be added.
Add $5.00 for shipping, insurance and handling on prepaid orders.
Minnesota residents add 4% sales tax.
Prices subject to change without notice.

Mail Check or Money Order To:
EXCERT, INC.
Educational Computer Division
P.O. BOX 8600
WHITE BEAR LAKE, MN. 55110
612-426-4114

www.commodore.ca

Note that if we poll for BUSY false before sending a character, we may wait forever, since for the first character, BUSY is already false, and no transition will occur to set the CA1 flag. We could preset the CA1 flat in the IFR before the first character, but now we're making our print routine complicated. Another solution would be to run BUSY to an input port bit as well so that we can poll its level, but this just about defeats the advantage of having the control bits on the port.

If you have a KIM, or have ports without the control bits, there are other solutions. For the printer example in particular, only seven bits of output are generally required, since most printers only support a character set of 96 characters or less. Thus the low seven bits of a port can be used for data, and the high bit used to generate a DATA STROBE. First the data is sent with the high bit high (to provide setup time), then the data is sent with the high bit low to generate a strobe. Finally the data is sent with the high bit high again to remove the strobe. A bit on another port is required to poll the BUSY flag from the printer. The software for this approach is actually less complicated than that for the fancy port with the handshaking.

Read Handshaking

Once the mechanics of handshaking are understood, except for the point of view, there is really little difference between read and write handshaking. In read handshaking, the computer is reading, rather than providing, the data. The peripheral device informs the computer that data is available. The computer may either poll the DA line, or use it to generate an interrupt. The computer then reads the data and then generates a DATA TAKEN signal so that the peripheral will know that it is time to send another data word. The ports of the 6522 can be used for read handshaking. The CA2 or CB2 line is used to acknowledge DATA TAKEN. CA1 or CA2 is used to input DATA AVAILABLE. Only Port A, however, can perform automatic read handshaking. That is, Port B will generate the CB2 handshake or pulsed response when so programmed, only for a read of Port B. Clearly, the CB2 line may be manually set or cleared to generate a DATA TAKEN. The timing for automatic read handshaking for Port A is show in Figure 4.

Applications

The most common applications of parallel data communication are communicating with peripherals like printers, floppy disk controllers, tape drive controllers, etc. However, the handshaking methods just described are also a very speedy way of communicating between two independent computer systems. All that is necessary is to connect a port of one into a port of the other, together with handshaking signals. Although the ports of a 6522 can be used bidirectionally, it may be simpler to dedicate a pair of ports for each direction. A relatively small program is required to send or receive data. The data may be prefixed by a two byte address indicating where the data is to be stored, or the receiving program may be provided with an independent address parameter. In any case, it should not take more than about 50 microseconds to respond to a DATA AVAILABLE, read the data, store it in memory, increment the address pointer, and respond with DATA TAKEN. In fact, if the worst case process time is known, the DATA TAKEN response can be eliminated. That is, if it takes 45 microseconds worst case to respond to a DA and be ready for the next DA, then the sending computer can send data along with a DA every 50 microseconds with complete confidence that no data will be lost. This simplifies both the send and receive routines at a negligible cost in speed. That would give a transfer rate of 20K bytes per second. Data blocks of 4K or 8K would be transferred before you even removed your finger from the "GO" button (Or Carriage Return).

With virtually instantaneous transfer of large blocks of data from one computer to another, the possibilities begin to suggest themselves. For example, after finishing some chore, the results of which need to be stored on tape, you might download the data to another computer which will generate the tape. While the second computer is generating the tape, you can go on to do the next job. Presently, I am using this technique to save and load programs on my SYM for a single-board computer from a different processor family which does not have its own tape interface or software. Writing the simple send/receive programs for the back to back ports was certainly simpler than writing tape programs for the other computer.

We will continue this discussion in the next column with serial techniques.

COMPUTE! Needs You!

Address articles, programming notes and comments to:

The Editor
COMPUTE!
P.O. Box 5406
Greensboro, NC 27403

www.commodore.ca

Peripheral Control Register

The Peripheral Control Register is organized as follows:

Bit #	7	6	5	4	3	2	1	0
Function		CB2 Control		CB1 Control		CA2 Control		CA1 Control

Each of these functions is discussed in detail below.

1. CA1 Control

Bit 0 of the Peripheral Control Register selects the active transition of the input signal applied to the CA1 interrupt input pin. If this bit is a logic 0, the CA1 interrupt flag will be set by a negative transition (high to low) of the signal on the CA1 pin. If PCR0 is a logic 1, the CA1 interrupt flag will be set by a positive transition (low to high) of this signal.

2. CA2 Control

The CA2 pin can be programmed to act as an interrupt input or as a peripheral control output. As an input, CA2 operates in two modes, differing primarily in the methods available for resetting the interrupt flag. Each of these two input modes can operate with either a positive or a negative active transition as described above for CA1. In the output mode, the CA2 pin combines the operations performed on the CA2 and CB2 pins of the SY6522. This added flexibility allows processor to perform a normal "write" handshaking in a system which uses CB1 and CB2 for the serial operations described above. The CA2 operating modes are selected as follows:

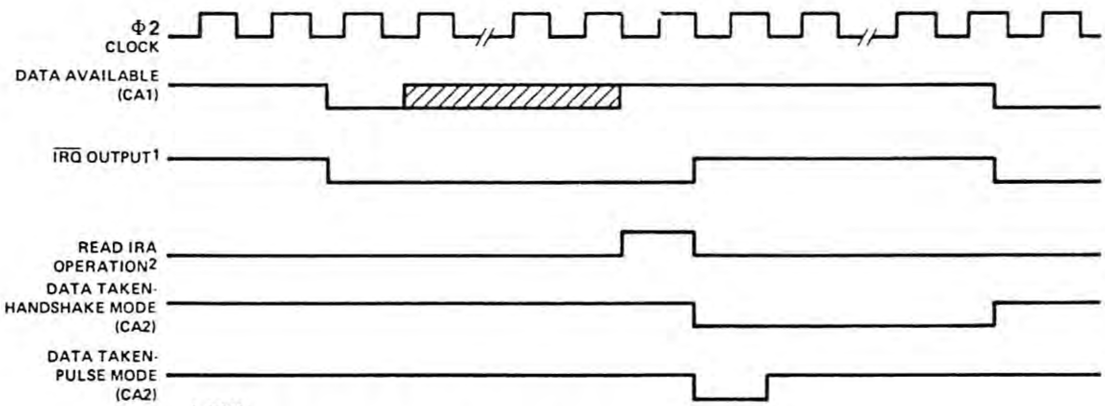
PCR3	PCR2	PCR1	Mode
0	0	0	Input mode–Set CA2 interrupt flag (IFR0) on a negative transition of the input signal. Clear IFR0 on a read or write of the Peripheral A Output Register.
0	0	1	Independent interrupt input mode–Set IFR0 on a negative transition of the CA2 input signal. Reading or writing ORA does not clear the CA2 Interrupt flag.
0	1	0	Input mode–Set CA2 interrupt flag on a positive transition of the CA2 input signal. Clear IFR0 with a read or write of the Peripheral A Output Register.
0	1	1	Independent Interrupt input mode–Set IFR0 on a positive transition of the CA2 input signal. Reading or writing ORA does not clear the CA2 interrupt flag.
1	0	0	Handshake output mode–Set CA2 output low on a read or write of the Peripheral A Output Register. Reset CA2 high with an active transition on CA1.
1	0	1	Pulse Output mode–CA2 goes low for one cycle following a read or write of the Peripheral A Output Register.
1	1	0	Manual output mode–The CA2 output is held low in this mode.
1	1	1	Manual output mode–The CA2 output is held high in this mode.

In the independent input mode, writing or reading the ORA register has no effect on the CA2 interrupt flag. This flag must be cleared by writing a logic 1 into the appropriate IFR bit. This mode allows the processor to handle interrupts which are independent of any operations taking place on the peripheral I/O ports.

The handshake and pulse output modes have been described previously. Note that the timing of the output signal varies slightly depending on whether the operation is initiated by a read or a write.

Figure 3. Port A PCR Bits

READ HANDSHAKE TIMING SEQUENCE



NOTES:

1. Signals "data available" to the system processor.
2. R/W = 1, $\overline{CS2}$ = 0, CS1 = 1, RS2 = 0, RS3 = 0, RS0 = 1.

Figures reprinted by permission, Synertek Systems Corp. ©

www.commodore.ca

Interfacing The Am9511 Arithmetic Processing Unit

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

Introduction

If you are interested in a hardware solution to the problem of addition, subtraction, multiplication, division, and functions such as sine, cosine, tangent, square root, exponential, logarithm and their inverse functions, then the Am9511 integrated circuit will be of interest to you. The Am9511 Arithmetic Processing Unit is a product of Advanced Micro Devices Inc., 901 Thompson Place, Sunnyvale, CA 94086. It performs signed multiplication, addition, subtraction and division with either 16-bit integers or 32-bit integers, in twos complement form. It also does these operations and evaluates a variety of functions (mentioned above) in a 32-bit floating point form. In the floating point form, the mantissa of the number is represented by 24 bits (equivalent to approximately seven significant decimal digits). The exponent is represented by six bits and a sign bit, giving a range of numbers that can be represented from roughly 10^{-19} to 10^{+19}. The one bit not accounted for so far is the sign of the mantissa. Thus, the Am9511 should satisfy most of the calculating needs of microcomputer users. It is important to point out that the Am9511 is a *binary* device as opposed to a *BCD* device. If you intend to use it like a calculator, then appropriate BCD-to-binary and binary-to-BCD routines will be needed to input and output numbers.

Timing of the various control pins on the Am9511 is one of the most important considerations in constructing an interface between it at the microprocessor. The timing requirements seem to be more relaxed in the most recent specification sheets, but my original specifications were quite complex. Perhaps it would be easy to interface the Am9511 somewhere in the address space, using address lines and control lines to operate it. However, given the complexities of the original timing diagrams, we used an interface adapter (the 6522, although any of the other popular interface adapters such as the 6530 can also be used with our programs). One port is used for data transfers, while several pins of the other port on the interface adapter is used to control the Am9511. These techniques produce an extremely simple interface at the expense of some overhead in software.

Before proceeding to the details of the circuit and the driver programs it should be pointed out that if you are interested in building and using this or some other circuit that uses the Am9511, you will want to get complete specification sheets, a publication called "Algorithm Details for the Am9511 Arithmetic Processing Unit," and a card-type Am9511 reference card. All three of these publications are available from Advanced Micro Devices. The Am9511 itself costs about $200, a number which may cause you to turn to the next article. A few mail order houses such as Advanced Computer Products are beginning to list the chip in their advertisements. Be sure to request all the literature mentioned above because you will need it to know how to use the chip. Space does not permit us to write a complete description of all the features of the chip.

The Am9511 Interface Circuit

The interface circuit is given in Figure 1. It is very simple because the complexity is absorbed in the software that must accompany this circuit. As noted, any 6502 system such as the SUPERKIM, KIM-1, AIM 65, etc., may be used, and any two-port interface adapter can be used. Be sure to include the 0.01 microfarad bypass capacitors, keep the leads between the Am9511 and the microcomputer short, and tie the unused control inputs ($\overline{\text{EACK}}$ and $\overline{\text{SVACK}}$) to logic one as shown in Figure 1. I will not reveal how many hours of grief the failure to follow these standard procedures cost me. Keep it simple, neat, and don't try any shortcuts. Also follow the usual procedures in handling integrated circuits that are susceptible to damage by static discharge. This is not your typical El Cheapo IC: $200 makes it irreplaceable. Avoid any Benjamin Franklin type experiments.

The Driver Subroutines

Listing 1 gives five subroutines that work with the interface circuit in Figure 1 to operate the Am9511. The subroutines are:

1. **RESET** - A subroutine that is used to reset the Am9511 either after power is applied or to clear the Am9511 to a known condition. This subroutine must be called after power-up and before using the Am9511.
2. **WRITE** - This subroutine transfers a byte of data in the accumulator of the 6502 to the stack of the Am9511.
3. **COMMAND** - A subroutine that transfers an eight-bit command word from the accumulator

www.commodore.ca

FACTORY PRICING

IN STOCK! IMMEDIATE DELIVERY!

ALL MOS TECHNOLOGY MPS 6500 ARRAYS---

PLUS

- MPS 6550 RAM for PET
- MPS 6530-002, -003 for KIM-1
- MANUALS
- KIM-1 MICROCOMPUTER
- KIM-3 8K STATIC RAM MEMORY BOARD
- KIM-4 MOTHERBOARD
- KIM PROMMER
  KIM-1 & 4 Compatable Eprom Programmer
- KIMATH
  Chips with Listing
- KIMEX-1 EXPANSION BOARD
  KIM-1 Plugable PROM, Ram and I/O Board
- RS-232 ADAPTER
  For KIM-1
- POWER SUPPLIES

STANDARD MICROSYSTEMS

★UART's ★FLOPPY DISC DATA HANDLER
★BAUD RATE GENERATORS ★CRT CONTROLLERS

FALK-BAKER ASSOCIATES

382 FRANKLIN AVE • NUTLEY, NEW JERSEY 07110
(201) 661-2430

WRITE, CALL, OR RETURN OUR COUPON FOR CATALOGUE AND PRICE LISTS.

www.commodore.ca

of the 6502 to the command register of the Am9511.

4. **READ** - Subroutine READ takes one byte of data (part of the answer) from the stack of the Am9511 and returns it to the X - register in the 6502.
5. **STATUS** - This subroutine reads the status register of the Am9511 and transfers its contents to the X - register in the 6502.

The comments in the various subroutines should be studied in connection with the Am9511 specification sheets to understand the functions of the various instructions. We only note here that each of the access subroutines, WRITE, COMMAND, READ, and STATUS, wait for the Am9511 to signal that an operation is complete when its $\overline{\text{PAUSE}}$ pin returns to logic one.

We will describe a few operations with the Am9511 to illustrate how the subroutines work. Refer to the literature mentioned previously for more details on the stack operation. The Am9511 stack may be regarded either as an eight-level, 16-bit wide stack, or as a four-level, 32-bit wide stack. Writing once to the Am9511 places an 8-bit word on the stack. However, since all of the "words" operated on by the Am9511 are either 16 bits or 32 bits wide, you must write at least 16 data bits (two bytes) to fill a 16-bit stack location. You must write four bytes to fill a 32-bit stack location. The *last* level filled (either 16 bits or 32 bits wide) is called TOS (acronym for top of stack). The level filled *previously* is referred to as NOS (next on stack).

An example will clarify the operation of the stack. Suppose we wish to add two 16-bit integers (they must be in twos complement form). Using the WRITE subroutine, we write the least-significant byte of one of the numbers to the Am9511 stack. Call this byte B1. Next we write B2, the most-significant byte of the same integer, to the Am9511. This puts a 16-bit integer onto TOS, the top level of the stack. The other addend, call it A1 and A2 for the least-significant and most-significant bytes respectively, is placed on the TOS by calling subroutine WRITE two more times. Now number B (B1 and B2) is in NOS and A (A1 and A2) is in TOS. The command code for a 16-bit addition, $6C, is now placed in the 6502 accumulator and subroutine COMMAND is called. The Am9511 adds TOS to NOS and puts the result into TOS. The result R, consisting of the most-significant byte R1 and the least-significant byte R2 of the 16-bit answer, is obtained by calling subroutine READ. The first call of READ retrieves the most-significant byte R2, and the second call of READ retrieves the least-significant byte of the result R. The status register can be read to see if the addition produced a carry or an overflow.

Subtraction follows exactly the same pattern. The minuend M is loaded on the stack, followed by the subtrahend S to obtain the difference D where D = M - S. After M and S are loaded on the stack, the subtraction command ($2D for a 32-bit word) will result in the difference D in TOS. Calling subroutine READ (twice for a 16-bit integer, four times for a 32-bit integer) gives the answer in the order from most-significant byte to least-significant byte. In division, the dividend is loaded on the stack followed by the divisor, and the quotient is read after the operation is completed. Some of you will recognize that the Am9511 uses RPN.

A program to illustrate these 16-bit operations is given in Listing 2. Suppose we wish to subtract $32FC from $FF5B. We would load $5B into location $0004, $FF into location $0003, $FC into location $0002, and $32 would be loaded into location $0001. The 16-bit subtraction command for the Am9511, $6D, would be loaded into location $0000. The program in Listing 2 will call the appropriate subroutines and place the answer in locations $00FF (most-significant byte) and $00FE (least-significant byte). This program can be used to test many of the operations of the Am9511, including sine, cosine, etc., by loading a 32-bit number (fixed or floating-point representation) on the stack, and then placing a command on the stack. It is a nice simple test program, but remember that many of the Am9511 functions require that the argument is in floating point form, so to find the square root of four requires that you convert four to a floating-point number. The Am9511 will do this if you either cannot or will not.

A word about execution time may be useful at this point. Instructions take from 16 clock cycles for a 16-bit integer addition to several thousand clock cycles for functions like sine, cosine, etc. We operated our Am9511 at 1MHz, but it can be operated at 2MHz and other versions go as high as 4MHz. Clearly the subroutines in Listing 1 require a significant amount of overhead for the simple integer operations, but become insignificant in terms of time overhead when the complex functions are called. Perhaps some reader will design an interface where instructions like STA DATA, STA COMMAND, LDA DATA, and LDA STATUS can be used instead of the subroutines. The difficulty is in working out the necessary timing requirements for the READ and WRITE operations of the 6502. The Am9511 timing seems to be more closely related to 8080A systems than either 6502 systems or 6800 systems.

Our final illustrative program is one that was designed to generate a sine table consisting of one cycle of a sine wave residing in one page of memory. The amplitude of the sine wave is $7F00, in other words, we found $7F00*Sin[Y*(Pi/128] where Y is a number that varied from $00 to $FF (0 to 255). This result was converted to a 16-bit fixed point format, and the most-significant byte was stored in a table in page $0E, while the least-significant byte was stored in a table in page $0F. Note that the result will be in twos complement form, so at location $0E80 in the

www.commodore.ca

HDE inc.

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

HUDSON DIGITAL ELECTRONICS INC.

THE TASK* MASTERS

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

OMNIDISK 65/8 and 65/5
Single and dual drive 8" and 5¼" disk systems. Complete, ready to plug in, bootstrap and run. Include HDE's proprietary operating system, FODS (File Oriented Disk System). From $795.00.

DM816-M8A
An 8K static RAM board tested for a minimum of 100 hours and warranted for a full 6 months. $195.00

DM816-UB1
A prototyping card with on-board 5V regulator and address selection. You add the application. $49.50

DM816-P8
A 4/8K EPROM card for 2708 or 2716 circuits. On board regulation of all required voltages. Supplied without EPROMS. $165.00

DM816-CC15
A 15 position motherboard mounted in a 19" RETMA standard card cage, with power supply. KIM, AIM and SYM versions. $545.00

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

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

HDE ADVANCED INTERACTIVE DISASSEMBLER (AID)
Two pass disassembler assigns labels and constructs source files for any object program. Saves multiple files to disk. TIM, AIM, SYM, KIM versions. $95.00

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

HDE TEXT OUTPUT PROCESSING SYSTEM (TOPS)
A comprehensive text processor with over 30 commands to format and output letters, documents, manuscripts. KIM, TIM and KIM cassette versions. $135.00 ($142.50 cassette)

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

HDE COMPREHENSIVE MEMORY TEST (CMT)
Eight separate diagnostic routines for both static and dynamic memory. TIM, AIM, SYM, KIM and KIM cassette versions. $65.00 ($68.50 cassette)

ORDER FROM THESE FINE DEALERS:

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

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

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

A-B Computers
115-B E. Stump Road
Montgomeryville, PA 18936
(215) 699-5826

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

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

Perry Peripherals
P.O. Box 924
Miller Place, NY 11764
(516) 744-6462

www.commodore.ca

table when we are exactly half-way through the sine wave, you will find $00, but at location $0E81 you will find the first negative value of the sine wave and it is $FC, the one in the most-significant bit of the 16-bit result indicating a minus number.

What do you do with a sine wave table? You could read it out to a D/A converter at various rates and play a tune, or you could add a series of sine waves to make a more complex sound. My purpose was to test the AM9511 and in the future I will use the sine wave table as part of a fast-Fourier transform program (I hope). Instead of synthesizing music I would really like to synthesize $20 bills. Let me know if you succeed.

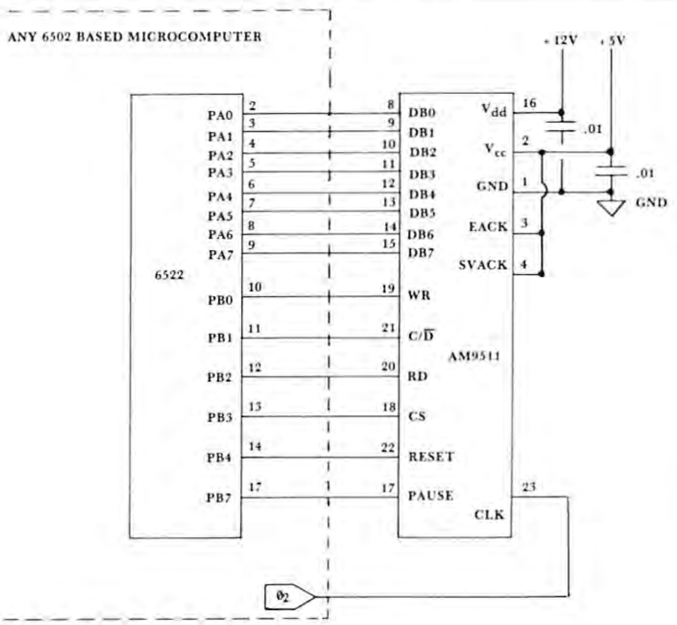


Figure 1.
Interfacing the AM9511 Arithmetic Processing Unit to a 6522 VIA Chip. Other interface adapters that may be used include the 6520, the 6530 and the 6532. No special handshaking pins are used.

Listing 1 Subroutines to drive the AM9511

```
0300 A9 1F     RESET     LDA $1F      Make PB0 - PB4
0302 8D 02 A0            STA PBDD     output pins to con-
                                      trol the AM9511.
0305 A9 0F               LDA $0F      RESET pin to
0307 8D 00 A0            STA PBD      logic zero.
030A A9 1F               LDA $1F      Hold RESET high
030C 8D 00 A0            STA PBD      for at least five
030F EA                  NOP          clock cycles.
0310 EA                  NOP
0311 A9 0F               LDA $0F      Bring RESET pin
0313 8D 00 A0            STA PBD      to logic zero to
                                      run the AM9511.
0316 60                  RTS          Return to the call-
                                      ing program.
               **************
0320 8D 01 A0  WRITE     STA PAD      A contains the
0323 A9 04               LDA $04      byte to be written
0325 8D 00 A0            STA PBD      to the AM9511
                                      (A = accumula-
                                      tor) CS low, C/D
                                      low, WR low.
0328 AD 00 A0  WAIT      LDA PBD      Read PBD to see
                                      if PAUSE pin is at
032B 10 FB               BPL WAIT     logic zero (no data
                                      transfer allowed).
032D A9 FF               LDA $FF      If PAUSE is high,
032F 8D 03 A0            STA PADD     make PAD an
                                      output port to
                                      transfer data to
                                      the AM9511.
0332 EE 00 A0            INC PBD      Bring WR high to
                                      complete data
                                      transfer.
0335 A9 0F               LDA $0F      Next bring CS,
                                      C/D high.
0337 8D 00 A0            STA PBD
033A A9 00               LDA $00      Now make Port A
033C 8D 03 A0            STA PADD     (PAD) an input
                                      port again.
033F 60                  RTS          Return to the
                                      calling program.
               **************
0340 8D 01 A0  COMMAND   STA PAD      A contains the
                                      command for the
                                      AM9511.
0343 A9 06               LDA $06      CS low, C/D
0345 8D 00 A0            STA PBD      high, WR low.
0348 AD 00 A0  LOAF      LDA PBD      Is PAUSE low?
034B 10 FB               BPL LOAF     Yes, then wait
                                      until it goes high.
034D A9 FF               LDA $FF      Make Port A an
034F 8D 03 A0            STA PADD     output port.
0352 EE 00 A0            INC PBD      Bring WR high.
0355 A9 0F               LDA $0F      Bring other con-
0357 8D 00 A0            STA PBD      trol pins high.
035A A9 00               LDA $00      Return Port A to
035C 8D 03 A0            STA PADD     input status.
035F 60                  RTS
0360 A9 01     READ      LDA $01      CS low, C/D low,
0362 8D 00 A0            STA PBD      RD low.
0365 AD 00 A0  LOITER    LDA PBD      Read PBD to see
                                      if PAUSE is low.
0368 10 FB               BPL LOITER   If it is, then wait
036A AE 01
A0                       LDX PAD      until it goes high.
                                      Am9511 output
                                      to X register.
036D A9 0F               LDA $0F      Bring control pins
036F 8D 00 A0            STA PBD      high.
0372 60                  RTS          Return to calling
                                      program with out-
                                      put in X.
               **************
0380 A9 03     STATUS    LDA $03      CS low, C/D
0382 8D 00 A0            STA PBD      high, RD low.
0385 AD 00 A0  DELAY     LDA PBD      Is PAUSE low?
0388 10 FB               BPL DELAY    Yes, then wait
                                      until it goes high.
038A AE 01 A0            LDX PAD      Read status regis-
038D A9 0F               LDA $0F      ter of AM9511
                                      and keep it in the
                                      X register.
038F 8D 00 A0            STA PBD      Bring control pins
                                      high.
0392 60                  RTS          Status is in X
                                      upon return.
```

www.commodore.ca

Listing 2 Program that loads four bytes (32 bits) and a command into the Am9511

```
0400 20 00 03  START  JSR RESET     Reset the AM9511
                                    to start using it.
0403 A2 03            LDX #03       Initialize X to
                                    count four bytes.
0405 B5 01     LOOP   LDA DATA,X    Get byte from the
                                    data table.
0407 20 20 03         JSR WRITE     Write the byte in-
                                    to
                                    the Am9511.
040A CA               DEX           Decrement byte
                                    counter.
040B 10 F8            BPL LOOP      Loop until four
                                    bytes are written.
040D A5 00            LDA CMND      Get command
                                    byte from location
                                    $0000.
040F 20 40 03         JSR           Write command
                      COMMAND       to the AM9511.
0412 20 60 03         JSR READ      Get MSB of 16-
                                    bit answer.
0415 86 FF            STX MSB       Put most-signifi-
                                    cant byte here.
0417 20 60 03         JSR READ      Get LSB of 16-
                                    bit answer.
041A 86 FE            STX LSB       Put least-signifi-
                                    cant byte in
                                    $00FE.
041C 00               BRK           End sample pro-
                                    gram here.
```

Listing 3. Sine table generator.

```
0500 20 00 03  SINE   JSR RESET     Reset the
                                    Am9511.
0503 A9 1A            LDA $1A       Push Pi
0505 20 40 03         JSR           (3.14159...) on
                      COMMAND       TOS by writing
                                    $1A to
                                    Am9511.
0508 A9 80            LDA $80       Load 128 =
050A 20 20 03         JSR WRITE     $0080 on TOS,
050D A9 00            LDA $00       Pi is pushed
050F 20 20 03         JSR WRITE     down to NOS.
0512 A9 1D            LDA $1D       Convert 128 =
0514 20 40 03         JSR           $0080 from
                      COMMAND       fixed point to
                                    to floating
                                    point form.
0517 A9 13            LDA $13       Divide NOS by
0519 20 40 03         JSR           TOS (Pi/128),
                      COMMAND       result onto
                                    TOS.
051C A0 00            LDY $00       Y serves as
                                    counter for 256
                                    points.
051E A9 37     REPEAT LDA $37       Duplicate NOS
0520 20 40 03         JSR           with TOS.
                      COMMAND       Pi/128 is now
                                    in TOS and
                                    NOS.
0523 98               TYA           Duplicate Y in
                                    accumulator.
0524 20 20 03         JSR WRITE     Push down
                                    TOS.
0527 A9 00            LDA $00       stack, Y into
0529 20 20 03         JSR WRITE     TOS.
052C A9 1D            LDA $1D       Change Y into
052E 20 40 03         JSR           floating point
                      COMMAND       form.
0531 A9 12            LDA $12       Multiply to get
0533 20 40 03         JSR           Y*(Pi/128).
                      COMMAND       Result to NOS.
                                    Pop stack up.
0536 A9 02            LDA $02       Take SIN[Y*
0538 20 40 03         JSR           (Pi/128)], result
                      COMMAND       to TOS.
053B A9 00            LDA $00       Push $7F00 on
053D 20 20 03         JSR WRITE     stack.
0540 A9 7F            LDA $7F
0542 20 20 03         JSR WRITE
0545 A9 1D            LDA $1D       Convert $7F00
0547 20 40 03         JSR            = 32512 to
                      COMMAND       floating point
                                    form.
054A A9 12            LDA $12       Find 32512*
054C 20 40 03         JSR           SIN[Y*(Pi/
                      COMMAND       128)], result to
                                    NOS, pop
                                    stack up.
054F A9 1F            LDA $1F       Convert that
0551 20 40 03         JSR           number to
                      COMMAND       fixed point
                                    format.
0554 20 60 03         JSR READ      Get MSB of
0557 8A               TXA           16-bit result in
                                    X register.
0558 99 00 0E         STA MSB,Y     Store it in a
                                    table in page
                                    $0E.
055B 20 60 03         JSR READ      Get LSB of 16-
055E 8A               TXA           bit result.
055F 99 00 0F         STA LSB,Y     Store it in a
                                    table in page
                                    $0F.
0562 C8               INY           Increment Y
                                    counter.
0563 D0 B9            BNE REPEAT    Repeat until
                                    table is filled.
0565 00               BRK           Break to the
                                    monitor.
```

©

www.commodore.ca

Interfacing KIM/SYM/AIM/OSI with Basic

Jim Butterfield, Toronto

Basic is a convenient and flexible language; but it isn't too fast. Machine language is fast, but rigorous to write. You can get the best of both worlds if you can make the two languages work together.

A hybrid program of this type invariably starts its run in Basic. Basic prints out the program title, and prompts the user for the detail of the job to be done. When it reaches a part where time is important, it will zip into machine language.

Getting there

Basic enters machine language by means of the USR function. The machine language coding will be written as a subroutine whose final command is RTS, signaling a return to Basic.

USR is a function: it's similar to SQR for square roots, RND for random numbers, etc. This means you can't start a Basic statement with USR: it must be part of an expression such as X = USR(0), or PRINT USR(99).

USR takes an argument: USR(6) passes a value of 6 to the machine language program. It returns a value: USR (6) might give back a value of say 12 to Basic. You don't need to use either of these. The machine language program can ignore the argument, and the Basic program can decide not to use the returned value. They are there if you need them.

Single routine

If you want one machine language subroutine and no more, it's quite easy. Poke the USR vector with the address of the subroutine. After that, the USR function will zip to that address every time it's used. The USR vector may be found at the following locations:

KIM: 0004 and 0005
SYM: 000B and 000C
AIM: 0004 and 0005
OSI: 000B and 000C

Check your Basic manual, if possible, to confirm that these are the locations that apply to your Basic package.

The address goes in low order first, as usual. Don't forget that Basic uses decimal numbers rather than hexadecimal. An example: to set up the address of the subroutine at OF22 on the KIM or AIM, you would code in Basic: POKE 4,34 : POKE 5,15. This needs to be done only once. After that, any USR reference takes you to OF22. For SYM or OSI, you'd code: POKE 11,34 : POKE 12,15.

Multiple machine language routines

There are several ways you can handle this.

You could repeat the pokes to the USR vector before each call. This is easy to code, but not lightning fast - POKEs from Basic are much slower than machine language.

If your routines come up in a certain order, you could have each machine language subroutine set up the next. A POKE in Basic is roughly equivalent to a STA machine language instruction. Each routine could set up the vector for the appropriate next USR entry.

Finally, you could keep a single entry point and have your machine language program decide which way to go on the basis of information supplied by Basic. This is discussed in the next section.

Single Entry Fanout

There are several ways that Basic could signal the type of job it wants done. It could POKE a location with a value that machine language could read and act upon.

A more complex method is to pass the information in the USR argument. USR(1) would mean, do job 1; USR(2), do job 2; and so on. This is a little trickier, since the argument is held in floating point. The next section will give more details on how to interpret it.

Passing parameters via the argument

When the function USR)6) is given, the argument-in this case, 6 - is placed in the floating-point accumulator. Later, when you return from machine language, the value in the floating-point accumulator is accepted by Basic as the value of the USR function. If you leave the floating-point accumulator alone, the value that went in comes back out. It's handy to keep in mind that you can use an expression as the argument: USR (X + Y*3 -2) is quite acceptable.

The floating point accumulator is at the following locations:

KIM - 00AE to 00B3
SYM - 00B1 to 00B6
AIM - 00A9 to 00AE
OSI - 00AC to 00B0

Note that the OSI floating point accumulator is one byte shorter than that of the other machines.

The first location is both zero flag and exponent. If it's zero, the whole number is zero and you don't need to look any further. If it's non-zero, it holds a binary exponent offset by $80. That means if it contains hex 80 or less, the number is a fraction less than 1. If it contains hex 81 or more, the number is greater or equal to 1. Don't worry about the details unless you have a mathematical leaning. It's useful to know, however, that you can double a number by adding one to the exponent, and halve it by subtracting one.

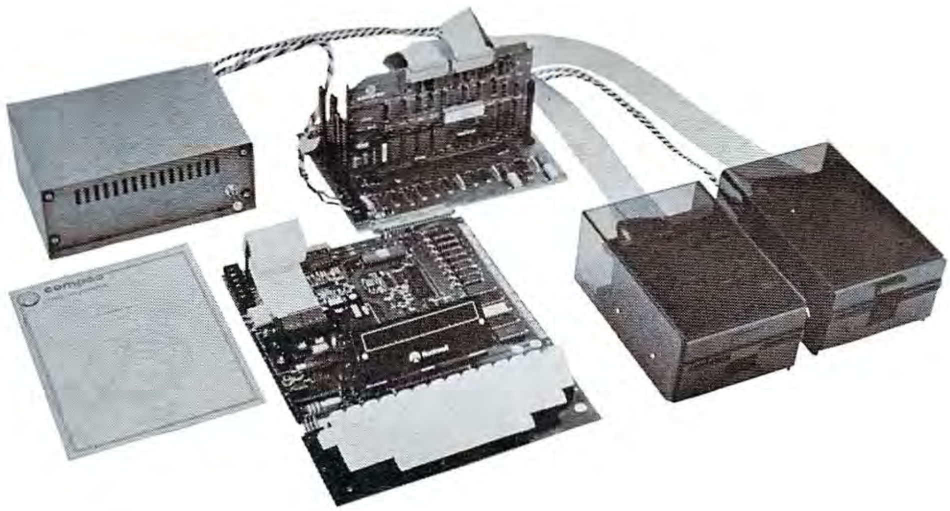
The next four locations are called the *mantissa* and hold the number itself. The number is always normally arranged so that its first 1-bit is in the high-order bit position of the mantissa. So numbers like 3, binary 11, and 6, binary 110 will have exactly the

www.commodore.ca

compas microsystems

P.O. Box 687
224 S.E. 16th Street
Ames, Iowa 50010
TWX 910-520-1166

DAIM



DAIM is a complete disk operating system for the ROCKWELL INTERNATIONAL AIM 65. The DAIM system includes a controller board (with 4K operating system in EPROM) which plugs into the ROCKWELL expansion motherboard, packaged power supply capable of driving two 5 1/4 inch floppy drives and one or two disk drives mounted in a unique, smoked plastic enclosure. DAIM is completely compatible in both disk format and operating system functions with the SYSTEM 65. Commands are provided to load/save source and object files, initialize a disk, list a file, list a disk directory, rename files, delete and recover files and compress a disk to recover unused space. Everything is complete — plug it in and you're ready to go! DAIM provides the ideal way to turn your AIM 65 into a complete 6500 development system. Also available are CSB 20 (EPROM/RAM) and CSB 10 (EPROM programmer) which may be used in conjunction with the DAIM to provide enhanced functional capability. Base price of $850 includes controller board with all software in EPROM, power supply and one disk drive. Now you know why we say —

There is nothing like a

DAIM

Phone 515-232-8187

www.commodore.ca

same mantissa: 11000000 ... How do we tell them apart? By using the exponent byte - the first location, remember?

The final byte contains the sign of the number. Only the first bit counts. If the first bit is zero, the number is positive; if it's one, the number is negative.

Floating point numbers are nice in Basic, but they can be difficult to handle in machine language. You'll probably want to use the built-in subroutines to covert them to and from the more familiar fixed-point numbers. See the Basic manual for this.

You can do the job yourself, if you prefer. Here's the general method. Assuming that your number is not zero (check the first byte) you can re-arrange it along the following lines. If you add one to the exponent, you will have multiplied the number by two; and if you shift the mantissa right, you will have divided it by two. If you do both, the number will have the same value. It will no longer be a normal floating-point number, since the high-order bit of the mantissa will now be zero, but the value will be the same. If you repeat this procedure until the exponent reaches a value of hexadecimal 90, the integer part of your number will be found in the first two bytes of the mantissa. It works: try it out with pencil and paper.

To go the other way (fixed to floating) you must "normalize" the number so that the high-order bit of the mantissa is 1; this takes left-shifting of the mantissa and decrementing the exponent.

Parameters: easier ways

Floating point is messy, and you may want to pass more than one value to or from machine language. There are other ways of doing the job.

The most obvious way is to have Basic POKE the values it wants to give into memory, and have the machine language program pick them up there. In the other direction, Basic can PEEK the results. If your values go above 255, you'll need to use more than one memory location for each value. Use the standard multiply or divide by 256 techniques to separate or recombine the parts.

A better way - but not quite so easy - is to have your machine language program go after the Basic variables in the locations they are stored in memory.

Variables: Ground Rules

Machine language can of course go after any data anywhere in memory. There are a few things you can do, however, to make it much easier to interchange data.

First rule: wherever possible, use Basic integer variables. These are the ones with the percent sign tacked on: J% or D%, for example.

The advantage of integer variables is that they are not stored in floating point notation. Machine language can use them, or change them, in a straightforward manner.

Second rule: arrange for Basic to use these variables at the very beginning of your program. If you want to pass six values (called A%, B%, X%, T1%, T2%, and S%) to machine language, have the first line of your Basic program define them with a line of code like:

```
100 A/ =0 : B/ =0 : X/ =0: S/ =0: T1/ =0 :
T2/ =0
```

This will place the values early in the variable table, where they are easy to access.

Variables: how they are stored

KIM, SYM, and AIM use seven locations for each variable; OSI uses six. The first two locations are the variable name, in ASCII. Fixed-point variables will have the high-order bit set over each byte of the name.

The next two locations of a fixed-point variable contain the binary value - high order first. The remaining two or three locations are not used.

Floating-point variables are also stored in seven (or six for OSI) locations. The format is slightly different from that of the floating-point buffer; a little experimentation should unlock secrets. You will find it generally simpler to use fixed-point format, except on the OSI Basic, which doesn't appear to have this option.

A couple of examples should make fixed-point formats easy to understand. If variable B5% has a value of 22, you'll see it stored as: C2 B5 00 16 00 00 00. C2 is an Ascii letter B with the high bit set; B5 is the Ascii character 5 with the high bit set - together they give the variable name. 00 16 is the value 22 in hexadecimal; and the remaining three locations are not used. If variable C% has a value of 300, you'll see: C3 80 01 2C 00 00 00. Can you figure it out?

Where to find the variables

The variables are normally stored above your Basic program. Since your program could be any size, the variables might start almost anywhere. You'll find out where by looking at your start-of-variables pointer. This is stored - low order first - at the following locations:

KIM - 007A and 007B
SYM - 007D and 007E
AIM - 0075 and 0076
OSI - 007B and 007C

So if your AIM contains the values B3 and 07 in 0075 and 0076, you'll know that your first variable is contained in location 07B3 to 07B9 inclusive. If it's a fixed-point variable, the value will be contained in 07B5 (high-order) and 07B6 (low-order).

You can look through the variable table, jumping seven locations at a time, to find the variable with the name you want. It's easier, as suggested before, to force the variables into the start of the table - that way they will be fast to find.

Here's a handy coding hint. The start-of-

www.commodore.ca

variables pointer can be used as an indirect address - after all, it's in zero page. So: if you wanted to get the low-order byte of the x first SYM variable, you could code: LDY #3; LDA ($7D),Y and you've got it. Count carefully; be sure that the variable is defined first in your Basic program; and the job becomes almost routine. You can reach over thirty variables this way, which is plenty for most applications.

If you want to pass values through an array, that's not hard to do. The format is similar to that of variables. Look around and you'll get the idea. One important caution: arrays can move during program execution. Always reference them through the start-of-arrays pointer, which is located directly after the start-of-variables pointer.

Conclusion

Your single-board machine is equipped with very powerful monitor facilities that allow you to look around and see how Basic does things. Use them: you'll find out a lot about how to get Basic and machine language to work harmoniously.

Basic and machine language can be married to give powerful and flexible programs. This brief article won't give you all the marriage counseling you need, but will at least perform the introductions. ©

6502 SOFTWARE

MACHINE LANGUAGE PROGRAMS FOR: *AIM *SYM *KIM *PET *APPLE *OSI & ANY 6502 SYSTEM

■ **FOLIO** A FILE ORIENTED LANGUAGE • A PSEUDO-CODE INTERPRETER • SUPPORTS 27 FUNCTIONS WHICH SIMPLIFY DATA HANDLING • SUPPORTS FILES WITH UP TO 254 RECORDS WITH UP TO 254 FIELDS WITH UP TO 80 CHARACTERS • FIELD SIZE IS VARIABLE FROM ONE RECORD TO ANOTHER • BUILT-IN KEY SORT FUNCTION • FOLIO INTERPRETER OCCUPIES ONLY 1.25K RAM OR ROM • TYPICAL FOLIO PROGRAMS TAKE LESS THAN 1K • DOES NOT REQUIRE BASIC, CP/M*, CBASIC*, 48K OR DISKS • WITH USER'S MANUAL, PROGRAMMER'S MANUAL & COMMENTED SOURCE LISTING — $25.00

FOLIO PROGRAMS:

■ **FILE CABINET** A MULTI-PURPOSE DATA STORAGE AND RETRIEVAL PROGRAM ONLY TAKES 320 BYTES • MENU DRIVEN • SELECT RECORDS KEYED BY ANY FIELD • SORT • CHOICE OF REPORT FORMATS • USE FOR MAILING LIST, INVENTORY, EMPLOYEE DATA, ETC •

■ **BUDGET** FOR HOUSEHOLD BUDGET PLANNING • PRINTS A 12 MONTH ACCOUNTING WITH UP TO 26 EXPENSE OR INCOME ITEMS WITH MONTHLY TOTALS AND BALANCE • MENU DRIVEN • EASY TO CHANGE ITEM NAMES OR AMOUNTS • REQUIRES 64 OR 72 CHARACTER LINE TERMINAL

■ **CHECKBOOK** ENTER CHECKS AS YOU WOULD IN YOUR CHECK REGISTER PLUS A CODE NUMBER FOR UP TO 256 CATEGORIES SUCH AS "FOOD","MEDICAL", ETC. • PRINTS THE STATEMENT WITH RUNNING BALANCE FOR SELECTED DATES OR IN NUMERICAL ORDER • PRINTS AND TOTALS CHECKS WITH SELECTED CODE # - FOR SELECTED DATES • PRINTS CODE TOTALS

■ EACH FOLIO APPLICATION PROGRAM COMES WITH THE FOLIO USER'S MANUAL, APPLICATION PROGRAM USER'S MANUAL, AND A HEX DUMP OF FOLIO AND THE PROGRAM — $10.00

OTHER 6502 PROGRAMS:

■ **TEA** TINY EDITOR/ASSEMBLER • CHARACTER ORIENTED EDITOR • SINGLE PASS ASSEMBLER • MAY BE USED SEPARATELY TO CONSERVE RAM • 1K EACH! • CLOSELY USES MOS MNEMONICS • COMES WITH USER'S MANUAL AND COMMENTED SOURCE LISTING — $20.00

■ **DISASSEMBLER** DISPLAYS SYMBOLIC LABELS AND OPERANDS FROM SYMBOL TABLE GENERATED BY TEA (ABOVE) • OR DISPLAYS ADDRESS VALUE • GREAT FOR DEBUGGING MACHINE CODE • WITH USER'S MANUAL AND COMMENTED SOURCE LISTING — $10.00

■ **ROBOT** INTERACTIVE PROGRAMMING LANGUAGE TO CONTROL ROBOT, PLOTTER OR CRT CURSOR • USER DEFINED COMMANDS & COMMAND SUBROUTINES • COMES WITH CRT ROUTINES WITH USER'S MANUAL AND COMMENTED SOURCE LISTING (1.5K) — $ 5.00

■ **MUSIC** INTERACTIVE PROGRAMMING LANGUAGE FOR THE CREATION OF PATTERNS OF SOUND "MUSIC" • A COMPOSITION TOOL • NOT A NOTE TABLE COMPILER OR "PIANO ROLL" • COMPLEX HIERARCHIES OF USER DEFINED FUNCTIONS - STRINGS OF MUSICAL EVENTS, WHICH CAN BE CALLED LIKE SUBROUTINES, ALLOW THE PROGRAMMING OF SURPRISINGLY INTRICATE COMPOSITIONS • WITH USER'S MANUAL & COMMENTED SOURCE LISTING — $10.00

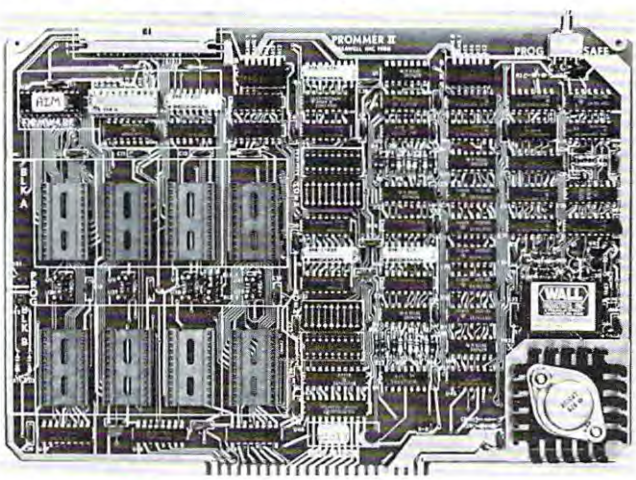
■ ALL PROGRAMS WORK IN A 4K SYSTEM • MANUALS CONTAIN INSTRUCTIONS FOR MODIFICATION USER EXTENSION, RELOCATION, AND INPUT & OUTPUT INTERFACING FOR 6502 PROGRAMMERS

■ CASSETTE TAPE (KIM-1 "HYPERTAPE") — $ 3.00

■ ORDER FROM: MICHAEL ALLEN: 6025 KIMBARK: CHICAGO, ILLINOIS 60637

*TM

SEAWELL PROMMER II -- There's Nothing Like It!



- Two independent blocks of 4 EPROM sockets
- KIM, SYM or AIM programming firmware
- Programs 1, 2 or 4K 5-Volt EPROMS: TMS2508, 2516, 2532 and 2758, 2716, 2732
- Read-Only/Deselected/Read-Program for each socket
- Program-protect toggle switch for whole board
- Provision for remoting 4 sockets
- On-board generation of programming voltage.

The Seawell PROMMER II is a general purpose EPROM tool designed for use in a development/production environment. Connects to a KIM, SYM or AIM with a Seawell LITTLE BUFFERED MOTHER motherboard, or to a SEA-1 single-board computer. The PROMMER II is all you need to read, program and execute 1, 2 or 4K 5-Volt EPROMs.

The PROMMER II allows you to put as little as 1K or as much as 32K of EPROM on the bus. You never have to give up address space to empty sockets. Each socket can be enabled to READ only, READ and PROGRAM, or can be DESELECTED entirely. Simply moving a shunt sets one block of 4 sockets to 1, 2, or 4K. The other block can be set independently.

Addresses are selected by piano-type switches on the top edge of the board. The whole board can be program-protected by a toggle switch on the top right corner of the board. A separate one-page ROM containing relocatable firmware for KIM, SYM or AIM is provided which can be set to any page in memory in either of two banks or deselected entirely. A satellite board with four sockets and program-protect switch will be available soon.

PROMMER II EPROM PROGRAMMER -- $299

OTHER SEAWELL PRODUCTS:

SEA-1	SINGLE BOARD DEVELOPMENT SYSTEM	$595
SEA-16	16K RAM BOARD	$280
SEA-FDC8	DOUBLE DENSITY 2-SIDED DISK CONTROLLER	$425
SEA-DEBUG	HARDWARE BREAKPOINTS AND TRIGGERS	$310
SEA-PROMMER II	EPROM PROGRAMMER	$299
SEA-PROTO	COMPLETELY DECODED PROTOTYPING BOARD	$ 99
SEA-CMOS	8K CMOS RAM, 16K EPROM, DAY/DATE CLOCK	$395
SEA-PIOB	4 FULLY-BUFFERED 6522S	$260
SEA-ISDC	8 SERIAL PORTS WITH FOCAL PROCESSOR & DUAL PORT RAM	$595
SEA-LBM	LITTLE BUFFERED MOTHER FOR KIM, SYM, AIM, SEA-1	$199
SEA-MOTHERS	4-SLOT MOTHERBOARD	$70.00
	10-SLOT MOTHERBOARD	$135

ALL PRODUCTS ASSEMBLED.
NO KITS

MASTERCHARGE/VISA ACCEPTED

SEAWELL MICROSYSTEMS
P.O. Box 30505,
Seattle, Washington 98103, U.S.A.,
(206) 782-9480

www.commodore.ca

Review

SYBEX, Inc.
2344 Sixth St.
Berkeley, CA 94710

"6502 Games" by Rodnay Zaks

by Harvey B. Herman

Most of us won't admit it but one of the reasons we have computers is to play games. The game may be traditional, like TIC-TAC-TOE, or intellectual, like solving a problem by writing and debugging a computer program. One of the "games" scientists are now playing is learning how to interface scientific instruments to computers. One of the stated purposes of the above book and accompanying hardware is to teach us how to play the game of computer interaction with the outside world (e.g., instruments). The instruction is accomplished by discussion of simple games (programmed in machine language) and the games board (lights, switches, and speaker). The following review gives my opinion on how well SYBEX has accomplished its goals.

The book, "6502 Games", and accompanying hardware, optional Games Board, is attractively packaged. The book's cover, although cute, has almost nothing to do with the book's contents. The hardware is trivial to connect to the SYM - just plug on the two edge connectors (A and AA). If your cassette is already connected to the terminal connector, nothing else is required except to press on letters on the keys of the Game Board. Otherwise your cassette will need to be reattached to either the new application's connector or the terminal connector.

The items reviewed here can be used in two ways. The simplist way is to load a game from the SYBEX tape, read the instructions in the book, and play the game with the Game Board. Games available range from Mindbender, a version of MasterMind, to Slot Machine, a simulation of a slot machine. Ten games in all (see list below) are available. They all interact with the keyboard and lights on the Games Board. Some, like Music Player, play tunes on the speaker of the Games Board. Others, like ECHO use the speaker to give audio clues to the player and signal when he wins or loses. These

SYBEX Games Board (for connection to SYM) Cassette Tape of Games for SYM

programs should keep a new SYM owner (without other software) happy until he is ready to graduate to more important things.

Another way to use these items, and here we begin to play a more serious game, is to modify the programs as suggested in the exercises or to even program the games from scratch. I believe anyone who could learn to do this would become an accomplished machine language programmer. In this case more reference material and possibly an assembler would be required. The book suggests other volumes in the SYBEX 6502 Series, which I have found helpful. Other reference books have recently become available. Let me emphasize how much one could learn if this path is followed. Let me also encourage readers as a first project, to construct their own games board. The author, Rodnay Zaks, feels (and I agree) that a much better understanding of the hardware will result.

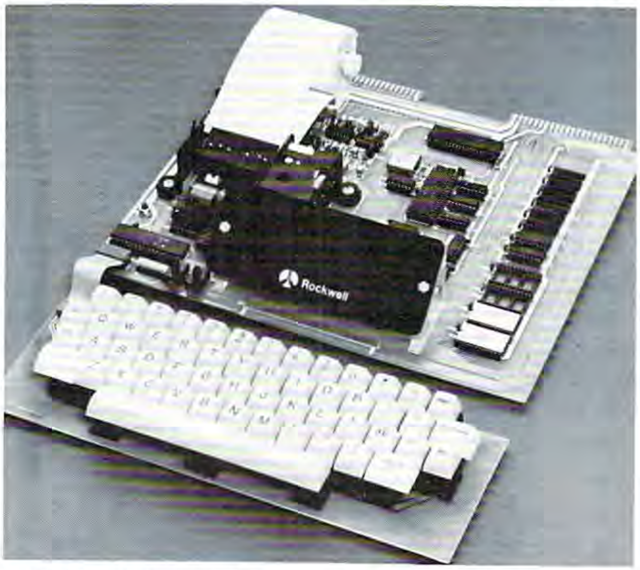
If you are a complete novice, let me caution you about one thing. The tape I received was not labelled and it was quite frustrating to figure out the order of programs. I hope this will be corrected in later shipments.

Order of Programs on my Tape (each recorded twice)

Title	*Location*	*Description*
1. Mindbender	$100-$3FF	Like Master Mind
2. Music Player	$100-$3FF	Play music from the keyboard
3. Magic Square	$100-$3FF	Light up a perfect square
4. Blackjack	$100-$3FF	Modified blackjack
5. Hexguess	$100-$3FF	Number Guessing game
6. Tic-Tac-Toe	$00-$F0 $100-$3FF	Plays game with you
7. Slot Machine	$00-$F0 $100-$3FF	Slot machine simulation
8. Echo	$100-$3FF	Like Simon
9. Spinner	$00-$F0 $100-$3FF	Check your reflexes
10. Translate	$00-$F0 $100-$3FF	Two player game

www.commodore.ca

AIM 65 BY ROCKWELL INTERNATIONAL



AIM 65 is fully assembled, tested and warranted. With the addition of a low cost, readily available power supply, it's ready to start working for you.

AIM 65 features on-board thermal printer and alphanumeric display, and a terminal-style keyboard. It has an addressing capability up to 65K bytes, and comes with a user-dedicated 1K or 4K RAM. Two installed 4K ROMS hold a powerful Advanced Interface Monitor program, and three spare sockets are included to expand on-board ROM or PROM up to 20K bytes.

An Application Connector provides for attaching a TTY and one or two audio cassette recorders, and gives external access to the user-dedicated general purpose I/O lines.

Also included as standard are a comprehensive AIM 65 User's Manual, a handy pocket reference card, an R6500 Hardware Manual, an R6500 Programming Manual and an AIM 65 schematic.

AIM 65 is packaged on two compact modules. The circuit module is 12 inches wide and 10 inches long, the keyboard module is 12 inches wide and 4 inches long. They are connected by a detachable cable.

THERMAL PRINTER

Most desired feature on low-cost microcomputer systems . . .

- Wide 20-column printout
- Versatile 5 x 7 dot matrix format
- Complete 64-character ASCII alphanumeric format
- Fast 120 lines per minute
- Quite thermal operation
- Proven reliability

FULL-SIZE ALPHANUMERIC KEYBOARD

Provides compatibility with system terminals . . .

- Standard 54 key, terminal-style layout
- 26 alphabetic characters
- 10 numeric characters
- 22 special characters
- 9 control functions
- 3 user-defined functions

TRUE ALPHANUMERIC DISPLAY

Provides legible and lengthy display . . .

- 20 characters wide
- 16-segment characters
- High contrast monolithic characters
- Complete 64-character ASCII alphanumeric format

PROVEN R6500 MICROCOMPUTER SYSTEM DEVICES

Reliable, high performance NMOS technology . . .

- R6502 Central Processing Unit (CPU), operating at 1 MHz. Has 65K address capability, 13 addressing modes and true index capability. Simple but powerful 56 instructions.
- Read/Write Memory, using R2114 Static RAM devices. Available in 1K byte and 4K byte versions.
- 8K Monitor Program Memory, using R2332 Static ROM devices. Has sockets to accept additional 2332 ROM or 2532 PROM devices, to expand on-board Program memory up to 20K bytes.
- R6532 RAM-Input/Output-Timer (RIOT) combination device. Multipurpose circuit for AIM 65 Monitor functions.
- Two R6522 Versatile Interface Adapter (VIA) devices, which support AIM 65 and user functions. Each VIA has two parallel and one serial 8-bit, bidirectional I/O ports, two 2-bit peripheral handshake control lines and two fully-programmable 16-bit interval timer/event counters.

BUILT-IN EXPANSION CAPABILITY

- 44-Pin Application Connector for peripheral add-ons
- 44-Pin Expansion Connector has full system bus
- Both connectors are KIM-1 compatible

TTY AND AUDIO CASSETTE INTERFACES

Standard interface to low-cost peripherals . . .

- 20 ma. current loop TTY interface
- Interface for two audio cassette recorders
- Two audio cassette formats: ASCII KIM-1 compatible and binary, blocked file assembler compatible

ROM RESIDENT ADVANCED INTERACTIVE MONITOR

Advanced features found only on larger systems . . .

- Monitor-generated prompts
- Single keystroke commands
- Address independent data entry
- Debug aids
- Error messages
- Option and user interface linkage

ADVANCED INTERACTIVE MONITOR COMMANDS

- Major Function Entry
- Instruction Entry and Disassembly
- Display/Alter Registers and Memory
- Manipulate Breakpoints
- Control Instruction/Trace
- Control Peripheral Devices
- Call User-Defined Functions
- Comprehensive Text Editor

LOW COST PLUG-IN OPTIONS

- A65-010—4K Assembler—symbolic, two-pass **$79.00**
- A65-020—8K BASIC Interpreter **99.00**
- 3K RAM Expansion Kit **50.00**

POWER SUPPLY SPECIFICATIONS

- +5 VDC ± 5% regulated @ 2.0 amps (max)
- +24 VDC ± 15% unregulated @ 2.5 amps (peak) 0.5 amps average

PRICE: $389.00 (1K RAM)

Plus $4.00 UPS (shipped in U.S. must give ***street*** address), $10 parcel post to APO's, FPO's, Alaska, Hawaii. All international customers write for ordering information.

We manufacture a complete line of high quality expansion boards. Use reader service card to be added to our mailing list, or U.S. residents send $1.00 (International send $3.00 U.S.) for airmail delivery of our complete catalog.

RNB ENTERPRISES INCORPORATED

2951 W. Fairmount Avenue
Phoenix AZ 85017
(602)265-7564

www.commodore.ca

KIM-1 Tidbits

Harvey B. Herman
Chemistry Department
University of North Carolina
at Greensboro
Greensboro, North Carolina 27412

If you are a regular reader of Compute magazine, you may have noticed that I am an owner of several small computers - KIM, SYM, and PET. (If that sounds as if I am some sort of addict, I confess that I am.) I don't have any favorite but PET has one feature, a screen editor, that I particularly like. Mike Louder (see, Best of PET Gazette) has taken the screen editor one step further and given us a "dynamic keyboard". With this it is possible to add, modify or delete BASIC statements while a program is executing. For example, it is possible to convert a machine language program, already in memory, to a series of DATA statements in a BASIC program. These DATA statements can be used to POKE the machine language program back into memory at any convenient later time. There are several advantages to doing it this way rather than by a conventional LOAD. The BASIC program could also include instructions on how to use the machine language program. It is not difficult to write the BASIC program in such a way so as to make the machine language program relocatable. Protection of the program is easily done by POKEs from BASIC. The machine language call can be done by SYS or USR. In my opinion, converting machine language programs to DATA statements is an altogether useful application of PET's "dynamic keyboard".

The dynamic keyboard idea, as far as I know, has not been extended to KIM. Unfortunately, KIM BASIC, unlike PET, does not use a keystroke buffer which is essential to the published method. I have been brooding over this difference for some time and have finally come up with a KIM procedure (program "DATA") which is described in this article. The example shown is for converting machine language programs to data statements but could easily be adapted to other uses of the dynamic keyboard.

KIM BASIC normally gets its character input using the monitor routine, GETCH, at location $1E5A. My idea is to temporarily modify the jump instruction to the KIM subroutine (locations $2456 -$2458 in my version of BASIC). BASIC would then jump to another routine which gets its input from a buffer in high memory instead of the terminal keyboard. In this example, the buffer contains BASIC DATA statements in ASCII format. The buffer is set up in a separate step using string concatenations, ASCII conversions, and POKEs to high memory. Since the "DATA" program is slightly complicated, even confusing me if I have not seen it for a while, I have described its operation in an accompanying table. Between the comments in the program and the table, I hope readers will understand the program more easily and be able to modify it to suit their own needs.

The routine which BASIC temporarily uses for input is POKEd into memory early in the program "DATA". I have included for reference the source code ("INPUT MOD") shown in the figure. This program was assembled with Eastern House Software's Assembler/Text Editor. It makes use of an address pointer which tells what location in high memory to get the next character from. Normal BASIC input is restored when the end-of-data character ($1A) is read. However, the last character returned is not $1A, but $0F (Control O) which toggles back the BASIC output suppress switch so BASIC will print again normally. Because the high ASCII buffer contained line feed characters it was necessary to toggle the switch initially, by reading $0F as the first character, to prevent unwanted double spacing.

In summary, one starts with a machine language program and the BASIC program called "DATA". After running "DATA" one is left with a number of DATA statements and a FOR/NEXT POKEing routine which can restore the machine language program at any subsequent BASIC session. The program left after running "DATA" can be augmented with instructions and protection POKEs if desired. I am aware of two obvious restrictions. One, the machine language program cannot overlap with BASIC or this BASIC program. Two, if BASIC is in ROM another method must be used. Please let me know if other KIM owners find this program useful or if there are any questions (SASE for reply).

Steps in "DATA" Program
(Line Numbers in Parenthesis)

1. Protect high memory (63047).
2. POKE machine language program (63048, 63500 SUB).
3. Input starting and ending locations of machine language program (63050-63055).
4. POKE control O ($0F) into first location of buffer (63065).
5. Construct one data statement from each 8 bytes (63080-63150).
6. POKE ASCII characters to high memory (63400 SUB).
7. Construct POKEing program and POKE ASCII to high memory (63160-63210, 63400 SUB).
8. POKE last character ($1A) to high memory (63212).
9. Change BASIC input character subroutine (63220).
10. Unprotect high memory and erase "DATA" program (63240).
11. Input is now from high memory (with echo to terminal) until last character ($1A) is read. At this point LIST should show a series of DATA statements and a FOR/NEXT POKEing routine. This program can be SAVEd for later use.

www.commodore.ca

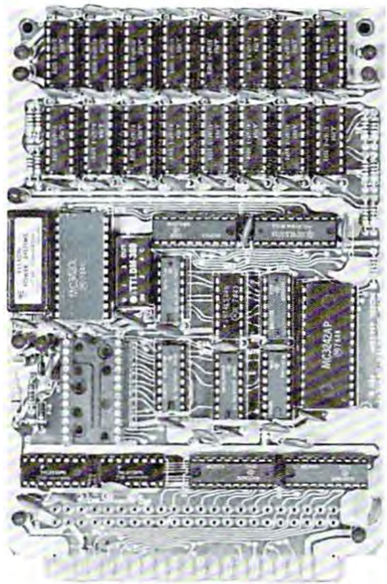
32 K BYTE MEMORY

RELIABLE AND COST EFFECTIVE RAM FOR 6502 & 6800 BASED MICROCOMPUTERS

AIM 65-*KIM*SYM PET*S44-BUS

- ★ PLUG COMPATIBLE WITH THE AIM-65/SYM EXPANSION CONNECTOR BY USING A RIGHT ANGLE CONNECTOR (SUPPLIED) MOUNTED ON THE BACK OF THE MEMORY BOARD.
- ★ MEMORY BOARD EDGE CONNECTOR PLUGS INTO THE 6800 S 44 BUS.
- ★ CONNECTS TO PET OR KIM USING AN ADAPTOR CABLE.
- ★ RELIABLE—DYNAMIC RAM WITH ON BOARD INVISIBLE REFRESH—LOOKS LIKE STATIC MEMORY BUT AT LOWER COST AND A FRACTION OF THE POWER REQUIRED FOR STATIC BOARDS.
- ★ USES +5V ONLY, SUPPLIED FROM HOST COMPUTER.
- ★ FULL DOCUMENTATION. ASSEMBLED AND TESTED BOARDS ARE GUARANTEED FOR ONE YEAR AND PURCHASE PRICE IS FULLY REFUNDABLE IF BOARD IS RETURNED UNDAMAGED WITHIN 14 DAYS.

ASSEMBLED	WITH 32K RAM	$395.00
&	WITH 16K RAM	$339.00
TESTED	WITHOUT RAM CHIPS	$279.00
HARD TO GET PARTS (NO RAM CHIPS) WITH BOARD AND MANUAL		$109.00
BARE BOARD & MANUAL		$49.00



PET INTERFACE KIT—CONNECTS THE 32K RAM BOARD TO A 4K OR 8K PET. CONTAINS: INTERFACE CABLE, BOARD STANDOFFS, POWER SUPPLY MODIFICATION KIT AND COMPLETE INSTRUCTIONS. $49.00

U.S. PRICES ONLY

16K MEMORY EXPANSION KIT

ONLY **$58**

FOR **APPLE, TRS-80** KEYBOARD, **EXIDY,** AND ALL OTHER 16K DYNAMIC SYSTEMS USING **MK4116-3** OR EQUIVALENT DEVICES.

- ★ 200 NSEC ACCESS, 375 NSEC CYCLE
- ★ BURNED-IN AND FULLY TESTED
- ★ 1 YR. PARTS REPLACEMENT GUARANTEE
- ★ QTY. DISCOUNTS AVAILABLE

ALL ASSEMBLED BOARDS AND MEMORY CHIPS CARRY A FULL ONE YEAR REPLACEMENT WARRANTY

BETA COMPUTER DEVICES
1230 W.COLLINS AVE.
ORANGE, CA 92668
(714) 633-7280

Calif. residents please add 6% sales tax. Mastercharge & Visa accepted. Please allow 14 days for checks to clear bank. Phone orders welcome. Shipping charges will be added to all shipments.

KIMEX-1 HERE'S A NEAT COMBINATION

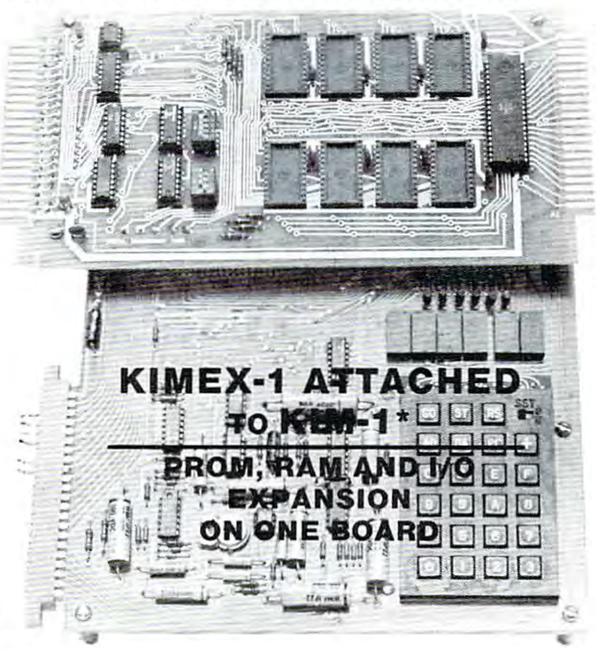
IDEAL FOR DEDICATED INDUSTRIAL OR PERSONAL APPLICATION

FEATURES

- PLUGS DIRECTLY INTO AND COVERS UPPER HALF OF KIM-1. EXPANSION FINGERS CARRIED THROUGH FOR FURTHER EXPANSION.
- I/O-POWERFUL 6522 VIA PROVIDED. (VERSATILE INTERFACE ADAPTER) 16 BI-DIRECTIONAL I/O LINES 4 INTERRUPT/HANDSHAKE LINES 2 INTERVAL TIMERS SHIFT REGISTER FOR SERIAL-PARALLEL/PARALLEL-SERIAL OPERATIONS.
- RAM-SOCKETS PROVIDED FOR 4K RAM CONTIGUOUS WITH KIM RAM. (LOW POWER MOSTEK 4118 1KX8's)
- COMPLETE DOCUMENTATION
- EPROM-SOCKETS PROVIDED FOR 8K EPROM. (INTEL 2716 2KX8's)
- BLOCK SELECT SWITCHES FOR EPROM. EPROM USABLE IN ANY ONE OF FOUR 8K BLOCKS FROM 8000H.
- AUTOMATIC RESET ON POWER-UP AND SWITCH SELECTABLE INTERRUPT VECTORS.
- PERMITS UNATTENDED OPERATION.
- LOW POWER CONSUMPTION-5V AT 300 Ma. FULLY LOADED
- BUFFERED ADDRESS LINES
- HIGH QUALITY PC BOARD, SOLDER MASK
- ASSEMBLED AND TESTED

APPLICATIONS

PROM, RAM AND I/O EXPANSION ON ONE BOARD HAVING MANY INDUSTRIAL/HOME APPLICATIONS FOR DATA ACQUISITION, PROCESS CONTROL, AUTOMATIC CONTROL OF FURNACE, SOLAR HEAT, LIGHTING, APPLIANCES, ETC. . . . .



THIS IS THE COMPLETE KIMEX-1

PA RESIDENTS INCLUDE 6% STATE SALES TAX

DIGITAL ENGINEERING ASSOCIATES
P.O.BOX 207 • BETHLEHEM, PA 18016

* KIM IS A REGISTERED TRADEMARK OF MOS TECHNOLOGY, INC.

www.commodore.ca

INPUT MOD

```
                0100 ;
                0110 ;TEMPORARY MODIFICATION TO KIM BASIC INPUT ROUTINE
                0120 ;INPUT A CHARACTER FROM HIGH MEMORY
                0130 ;INSTEAD OF KEYBOARD
                0140 ;RESTORE NORMAL OPERATION WHEN $1A
                0150 ;CHARACTER IS READ FROM HIGH MEMORY
                0160 ;
                0170 ;HARVEY B. HERMAN
                0180 ;
                0190                .BA $200
0200- 18        0200                CLC
                0210 ;BUMP STORAGE POINTER
                0220 ;NOTE: PROGRAM MODIFIES ITSELF
                0230 ;        SORRY EPROM FREAKS
0201- AD 12 02  0240                LDA COUNT+1
0204- 69 01     0250                ADC #01
0206- 8D 12 02  0260                STA COUNT+1
0209- AD 13 02  0270                LDA COUNT+2
020C- 69 00     0280                ADC #00
020E- 8D 13 02  0290                STA COUNT+2
0211- AD FF 5F  0300 COUNT          LDA HIGH-MEM
                0310 ;CHECK FOR DONE($1A)
0214- C9 1A     0320                CMP #$1A
0216- F0 06     0330                BEQ END
                0340 ;KIM OUT
0218- 48        0345                PHA
0219- 20 A0 1E  0350                JSR $1EA0
021C- 68        0355                PLA
021D- 60        0360                RTS
                0370 ;RESTORE KEYBOARD INPUT
021E- A9 5A     0380 END            LDA #$5A
0220- 8D 57 24  0390                STA $2457
0223- A9 1E     0400                LDA #$1E
0225- 8D 58 24  0410                STA $2458
0228- A9 0F     0420                LDA #$0F
                0430 ;RETURN WITH CONTROL O
022A- 60        0440                RTS
                0450 HIGH-MEM       .DE $5FFF      ;-1
                0460                .EN
```


KIM AIM SYM TIM

END FRUSTRATION!!

FROM CASSETTE FAILURES
PERRY PERIPHERALS HAS
THE HDE SOLUTION
OMNIDISK SYSTEMS (5" and 8")

ACCLAIMED HDE SOFTWARE

- Assembler, Dynamic Debugging Tool, Text Output Processor, Comprehensive Memory Test
- Coming Soon—HDE BASIC

PERRY PERIPHERALS S-100 PACKAGE

Adds Omnidisk (5") to
Your KIM/S-100 System

- Construction Manual—No Parts
- FODS & TED Diskette
- $20. +$2. postage & handling. (NY residents add 7% tax) (specify for 1 or 2 drive system)

Place your order with:
PERRY PERIPHERALS
P.O. Box 924
Miller Place, N.Y. 11764
(516) 744-6462

Your Full-Line HDE Distributor/Exporter

DATA

```
63000 REM KIM PROGRAM WHICH MAKES DATA STATEMEMENTS
63010 REM FROM A MACHINE LANGUAGE PROGRAM
63020 REM
63030 REM HARVEY B. HERMAN
63040 REM
63045 REM PROTECT HIGH MEMORY
63047 POKE 132,0:POKE 133,96:CLEAR
63048 GOSUB 63500:REM POKE MACHINE LANGUAGE PROGRAM
63050 INPUT "STARTING LOCATION";S
63055 INPUT "ENDING LOCATION";E
63060 N=10
63064 REM H IS HIGH MEMORY LOCATION FOR SAVE
63065 H=24576:POKEH,15:H=H+1
63080 FOR I=S TO E STEP 8
63085 A$=STR$(N)+" DATA "
63090 FOR J=0 TO 6
63100 A=PEEK(I+J)
63110 IF (I+J)=E THEN 63130
63115 A$=A$+STR$(A)+","
63120 NEXT J
63125 A=PEEK(I+J)
63130 A$=A$+STR$(A)+CHR$(13)+CHR$(10)
63140 GOSUB 63400
63145 N=N+10
63150 NEXT I
63160 A$=STR$(N)+" FOR I="+STR$(S)+" TO"+STR$(E)+
      CHR$(13)+CHR$(10)
63170 GOSUB 63400
63190 N=N+10
63200 A$=STR$(N)+" READ A:POKE I,A:NEXT I"+CHR$(13)+
      CHR$(10)
63210 GOSUB 63400
63212 POKE H,26:REM LAST CHARACTER
63215 REM CHANGE INPUT ROUTINE
63220 POKE 9303,0:POKE 9304,2
63230 REM UNPROTECT HIGH MEMORY
63240 POKE 132,0:POKE 133,144:NEW
63390 REM PICK APART STRING AND SAVE IN HIGH MEMORY
63400 FOR K=H TO H+LEN(A$)-1
63410 POKE K,ASC(MID$(A$,K-H+1,1))
63420 NEXT K
63430 H=H+LEN(A$)
63440 PRINT A$;:POKE 22,0:REM ZERO PRINT POSITION
63450 RETURN
63500 DATA 24,173,18,2,105,1,141,18
63510 DATA 2,173,19,2,105,0,141,19
63520 DATA 2,173,255,95,201,26,240,6
63530 DATA 72,32,160,30,104,96,169,90
63540 DATA 141,87,36,169,30,141,88,36
63560 DATA 169,15,96
63570 FOR I= 512 TO 554
63580 READ A:POKE I,A:NEXT I
63590 RETURN
OK
```

©

www.commodore.ca

AIM 65 Tape Copy Utility

Christopher J. Flynn

Introduction

If you're an AIM 65 user, you've probably stored your favorite programs and important data bases on cassette tape. Have you thought about making backup copies of your tapes? I didn't until my tape recorder ate my only copy of a 1000 line assembly language program that I was writing.

You may be thinking it is too much trouble to make backup tapes on the AIM. Each file has to be loaded into memory and then written back out. If you have machine language programs, Basic programs, and text files, then you have to follow three different load and dump procedures. Machine language programs are the worst to copy. Sure, it is very easy to load them into memory. Have you tried dumping such a program when you've lost the little piece of paper that had the memory addresses on it?

Well, here is a little 44 byte program that will make tape copying easy. All you do is put the tape to be copied in drive 1 and a blank tape in drive 2. Then, position the tapes and let the program do the rest. The program will copy any kind of AIM file. It will even copy multiple input files from the same tape. So now, none of us should have any excuse for not having backup copies of our important tapes.

Hardware Required

First of all, I'll assume that you have an AIM. An AIM with just 1K of RAM will do fine.

Next, you'll need to attach two cassette recorders to your AIM. Chances are you already have one. If nothing else, maybe this article will give you an excuse to buy a second one. By the way, the versatility of the AIM definitely improves with the second recorder.

Finally, you should connect the remote control circuits to each of the recorders. You should experiment with the setting of GAP ($A409) as described in the AIM manual. Pick a value of GAP that lets you record on one device and play back on the other reliably. I have found that the default value of $08 works well. It only worked, however, after I modified my recorders (Radio Shack) so that their electronics would remain on even when the motor was toggled off.

Tape Copy Procedure

Let's go through the step by step procedure of copying a tape.

1. Load the tape copy program into the AIM's memory starting at $0200. The program is easily relocated, but you'll have to observe the cautions described in a later section.
2. Place the tape to be copied in drive 1. This program assumes that drive 1 is used only in the playback mode.
3. Place a blank tape in drive 2. This program assumes that drive 2 is used only in the record mode.
4. Position the tapes.
   a. Position the tape in drive 1 to a point just beyond the leader. Use the "1" monitor command to toggle drive 1 off.
   b. Position the tape in drive 2 to a point about 4 turns beyond the leader. Use the monitor "2" command to toggle drive 2 off.
5. Start the tape copy program.
   a. Use the monitor "*" command to set the AIM's program counter to $0200.
   b. Use the monitor "G" command to begin the program.
6. Watch the AIM display. The display will alternately show an "S" and a "W". The "S" means that the program is searching for the next block. The "W" means that the program is in the process of writing a block to drive 2.
7. Hit reset to stop the copy program when a steady display of "S" appears without any intervening "W"'s.
   a. Drive 1 will be on and you can rewind and remove the input tape.
   b. Drive 2 will be off. This allows you to stack additional programs or data on the same output tape. You will have to toggle drive 2 with the "2" command when you are ready to rewind the output tape.

That's all there is to copying a tape. Notice that at no time did the AIM ask you "IN =" or "OUT =". It did not even ask you for the input and output file names.

By the way, you should probably verify the first few tape copies that you make just to be sure that the program works and that GAP is set properly.

How It Works

The Tape copy program makes use of subroutines in the AIM monitor. Basically, the program reads a data block from drive 1 (subroutine TIBY1 at $ED53) into the AIM's tape buffer. The data block is then written from the buffer to drive 2 by an AIM subroutine beginning at $F19C which I've called BLKOUT. In between data blocks, the program writes either an "S" or a "W" to the AIM display. This process of reading and writing a block continues forever or until reset is pushed or the plug is pulled.

www.commodore.ca

FIGURE 1

```
AIMASM

LOC      CODE       LINE

022C                   *=$0200
0200                   .OBJ $8000
0200                ;
0200                ;AIM 65 TAPE COPY UTILITY
0200                ;
0200                ;DRIVE 1 IS INPUT DRIVE
0200                ;DRIVE 2 IS OUTPUT DRIVE
0200                ;
0200                ;BY CHRIS FLYNN 8/80
0200                ;
0200                ;
0200                ;AIM 65 MONITOR ROUTINES USED
0200                ;
0200                CLR    =$EB44
0200                OUTDP  =$EEFC
0200                TIBY1  =$ED53
0200                PHXY   =$EB9E
0200                BLKOUT =$F19C
0200                ;
0200                ;AIM 65 RAM LOCATIONS USED
0200                ;
0200                TAPIN  =$A434
0200                TAPOUT =$A435
0200                BLOCK  =$0115
0200                ;
0200                ;TAPE COPY INITIALIZATION
0200                ;
0200 A9 00          COPY   LDA #0
0202 8D 34 A4              STA TAPIN       ;SET DRIVE 1 AS INPUT
0205 8D 15 01              STA BLOCK       ;CLEAR BLOCK COUNT
0208 A9 01                 LDA #1          ;SET DRIVE 2 AS OUTPUT
020A 8D 35 A4              STA TAPOUT
020D                ;
020D                ;READ A TAPE BLOCK INTO AIM 65 BUFFER
020D                ;
020D 20 44 EB       READ   JSR CLR
0210 A9 53                 LDA #'S         ;INDICATE SEARCHING FOR BLOCK
0212 20 FC EE              JSR OUTDP
0215 A2 00                 LDX #0
0217 20 53 ED              JSR TIBY1       ;READ A BLOCK
021A                ;
021A                ;WRITE THE BLOCK FROM THE AIM BUFFER
021A                ;NOTE: BLKOUT WILL DO A JSR PLXY AND THEN RTS.
021A                ;THEREFORE, WE PRELOAD RETURN ADDR ON STACK.
021A                ;
021A 20 44 EB       WRITE  JSR CLR
021D A9 57                 LDA #'W         ;INDICATE WRITE IN PROGRESS
021F 20 FC EE              JSR OUTDP
0222 A0 02                 LDY #>READ      ;PUT RETURN ADDRESS IN Y,X
0224 A2 0C                 LDX #<READ-1    ;HI PART IN Y, LO PART IN X
0226 20 9E EB              JSR PHXY        ;NOW PUT RETURN ADDRESS ON STACK
0229 20 9C F1              JSR BLKOUT      ;OUTPUT THE BLOCK AND READ NEXT ONE
```

The listing in Figure 1 shows the assembly language code for the tape copy program. The only tricky part of the program is the JSR to BLKOUT. BLKOUT is really a part of the AIM subroutine TOBYTE ($F18B). A problem arises because the tape copy program calls TOBYTE at a point other than its normal entry point.

The first and last two statements of TOBYTE are:

JSR PHXY

●

●

www.commodore.ca

```
•
JSR PLXY
RTS
```

Notice that TOBYTE saves the X and Y registers on the stack. When TOBYTE is called in the middle, the X and Y registers do not get saved. So, when TOBYTE finishes, the JSR PLXY does not pick up X and Y. Instead, it removes the return address from the stack. Therefore, the RTS picks up garbage from the stack and the AIM hangs!

To get around this problem, the tape copy program preloads X and Y before calling BLKOUT. The values loaded into X and Y represent the return address. X and Y are then stored on the stack. Lastly, the JSR to BLKOUT is done.

Figure 1 shows the way X and Y are loaded. The most significant byte of (return address - 1) is placed in Y. The least significant byte of (return address - 1) is placed in X. One is subtracted from the return address in order to mimic the way the 6502 stores return addresses on the stack. If you relocate this program, you will have to load X and Y with the appropriate values.

Summary

This article has described a simple tape copy utility for the AIM 65. I hope that you find it both useful and easy to use. ©

Model EP-2A-79

EPROM Programmer

PET · APPLE · AIM-65 · KIM-1 · SYM-1 · OHIO SCIENTIFIC

Software available for F-8, 6800, 8085, 8080, Z-80, 6502, 1802, 2650,6809, 8086 based systems.

EPROM type is selected by a personality module which plugs into the front of the programmer. Power requirements are 115 VAC 50/60 Hz. at 15 watts. It is supplied with a 36-inch ribbon cable for connecting to microcomputer. Requires 1½ I/O ports. Priced at $155 with one set of software. (Additional software on disk and cassette for various systems.) Personality modules are shown below.

Part No.	Programs	Price
PM-0	TMS 2708	$15.00
PM-1	2704, 2708	15.00
PM-2	2732	30.00
PM-3	TMS 2716	15.00
PM-4	TMS 2532	30.00
PM-5	TMS 2516, 2716, 2758	15.00
PM-8	MCM68764	33.00

Optimal Technology, Inc.
Blue Wood 127, Earlysville, Virginia 22936
Phone (804) 973-5482

Compare Our Prices With Any Others

Rockwell's	AIM-65	1K System:	**$405.**	4K System	**$459.**
Synertek's	SYM-1	1K System:	**235.**	4K System	**259.**
Commodore's	KIM-1	1K System:	**175.**		

FOR YOUR SYSTEM'S EXPANSION

The Computerist, Inc's:

16K DRAM	**$279.**	Proto Plus II	**$42.**
32K DRAM	**375.**	ASK I/O Board	**55.**
Video Plus II	**279.**	DRAM & Video Cable	**15.**
Mother Plus II & Card Cage	**115.**	Power Supply for SYM-1	**39.**

Power Supply and Enclosure for AIM-65 **$119.**
Power Supply and Enclosure for KIM-1 **65.**

All products are factory warrantied. Prices include full documentation.

Send Check or Money Order to:
Hepburn MCA*
12 Grosvenor Street
Lowell, MA 01851

Please add $5.00 shipping and handling. MA residents add 5% sales tax.
** Mini Computers and Accessories*

www.commodore.ca

Combining BASIC And Machine-Language Programs On Tape

George Wells

This article describes a procedure to combine a machine-language program and a BASIC program into a single cassette tape file which can be LOADed and RUN without exiting BASIC. This procedure is specifically applied to a SYM-1, but the technique may be applicable to other machines, particularly Microsoft BASICs that store programs on tape in tokenized form exactly as they appear in memory.

General Discussion Of Technique

Whenever a BASIC programmer wants to jump to a machine-language program by way of the USR command, he has to decide where in RAM he is going to put the object code for the machine-language program. The usual place to put such code (assuming it is too big to squeeze into one of the unused areas on page zero or page one or some other place) is near the top of his contiguous RAM space which starts at page zero and includes at least 4K or 8K of memory. The method by which this is accomplished is to exit BASIC, load the object code from a file on tape, re-enter BASIC with an appropriate response to MEMORY SIZE? so BASIC will not use the memory allocated to the machine program and finally LOAD and RUN the BASIC program. In order to avoid this cumbersome procedure, we can put the two types of programs next to each other so that they can be LOADed together from one tape file into memory.

The technique to perform this is to make two tape files, the first one containing the BASIC program and the second one containing the machine code assembled somewhere in memory after the end of the BASIC program. Then all you have to do is enter BASIC, LOAD the BASIC program, LOAD the machine program, and SAVE the combined program. Now you have both programs on the same tape file which can be LOADed just like any other BASIC program. If you change the BASIC program, you will have to reLOAD the machine program and reSAVE the combined program. There are two pitfalls to be avoided when making changes. First, if the BASIC program expands to the point where it runs into the machine code, you will have to reassemble the machine program at a higher address, make a tape copy, modify the BASIC program to link properly to the new machine code, reLOAD the new machine code, and reSAVE the new combined program. Second, if you get a BAD LOAD error when trying to LOAD the machine code, your BASIC program will be deleted; so it's a good idea to SAVE the BASIC program after making any changes. In order to avoid these problems, you will probably want to assemble your machine-language program at the top of your RAM and check out your BASIC program as much as possible before combining the two programs together.

Specific Example On A SYM-1

This example will take a BASIC program that uses the trig functions and combine it with the machine code which the user must supply in order to use trig with the SYM-1 BASIC. It's a good idea to practice this technique on a simple BASIC program to get a feel for how it works before attempting a serious application.

STEP 1: Cold start to BASIC and enter the following program:

```
100 X = Y: REM CHANGE Y TO LAST
PAGE OF
TRIG.
110 POKE 196,104: POKE 197,X
120 PRINT SIN(1), COS(2), TAN(3),
ATN(4)
```

STEP 2: Save the BASIC program on tape with SAVE B.

STEP 3: Go to the monitor (by way of Reset) and look at memory locations $7D and $7E. These two values are the low and high bytes of the first available address after the BASIC program. The value of this address should be increased by at least 30 or 40 or even several hundred if extensive changes are expected in the BASIC program. In this example, we could safely start the machine code anywhere after address $0290.

STEP 4: Store the object code for the trig functions (from Synertek Systems, Inc. Technical Note 53) so that it ends at the end of page three.

STEP 5: Save the machine code on a second tape using an ID of $4D (ASCII "M") with the following command:

```
.S2 4D,2C7-3FF
```

STEP 6: Cold start back to BASIC and LOAD B to get the BASIC program.

STEP 7: Since we now know the location of the machine code, re-enter line 100:

www.commodore.ca

100 X = 3: REM TRIG FUNCTIONS
END ON
PAGE 3.

STEP 8: Save the modified BASIC program on your first tape with SAVE B.

STEP 10: Enter LOAD M to load the machine code. If you get a LOADED message, go to STEP 12. If you get a BAD LOAD error message continue with STEP 11.

STEP 11. Reload the BASIC program with LOAD B and continue from STEP 10.

STEP 12: Save the combined program on a third tape with SAVE C. At this point, you can enter any valid BASIC command (try RUN and LIST) but when you get ready to modify the BASIC program continue from STEP 13.

STEP 13: Make as many changes as desired but DO NOT RUN the program.

STEP 14: Save the program on your first tape with SAVE B. This tape will now contain a valid BASIC program combined with invalid machine code. If you are sure that there is no danger of your BASIC program expanding into your machine code then continue from STEP 10. If you are not sure, continue with STEP 15.

STEP 15: There is no easy way to tell how big the BASIC part of the combined program is since the address at $7D, $7E is pointing somewhere near the end of the machine code. You could go to the monitor and manually search for three zero-bytes in a row which shouldn't be too hard if you have a general idea of where to look. Don't forget to insure that the system RAM is not write-protected after returning to BASIC. Another way to accomplish the same thing without leaving BASIC is to enter the following direct command (without spaces):

FORI = 515TO33333:IFPEEK(I-3) (   ) 0ORPEEK (I-2) (   ) 0ORPEEK (I-1) (   ) 0THENNEXT

and wait for BASIC to respond with OK (it can take minutes). Then enter PRINT I and the computer will give the decimal equivalent of the first unused memory location after the BASIC program. If you run out of space between the two programs, reassemble the machine-language program at a higher address and continue at STEP 5. If you decide that you have sufficient space between the programs, you can continue at STEP 10.

NOTE: If at any time you suspect that the BASIC program has clobbered the machine program, you should reset your system, cold start to BASIC, LOAD B with the latest version of your program and continue at STEP 15.

NOTE: If you continue the trig functions with a BASIC program as in this example, you should take precaution to set the pointer at 196 and 197 back to

6502 FORTH

- 6502 FORTH is a complete programming system which contains an interpreter/compiler as well as an assembler and editor.
- 6502 FORTH runs on a KIM-1 with a serial terminal. (Terminal should be at least 64 chr. wide)
- All terminal I/O is funnelled through a jump table near the beginning of the software and can easily be changed to jump to user written I/O drivers.
- 6502 FORTH uses cassette for the system mass storage device
- Cassette read/write routines are built in (includes Hypertape).
- 92 op-words are built into the standard vocabulary.
- Excellent machine language interface.
- 6502 FORTH as user extensible.
- 6502 FORTH is a true implementation of forth according to the criteria set down by the forth interest group.
- Specialized vocabularies can be developed for specific applications.
- 6502 FORTH resides in 8K of RAM starting at $2000 and can operate with as little as 4K of additional contiguous RAM.

6502 FORTH PRICE LIST

KIM CASSETTE, USER MANUAL, AND COMPLETE ANNOTATED SOURCE LISTING $90.00
($2000 VERSION) PLUS S&H 4.00

USER MANUAL (CREDITABLE TOWARDS SOFTWARE PURCHASE) $15.00
PLUS S&H 1.50

SEND A S.A.S.E. FOR A FORTH BIBLIOGRAPHY AND A COMPLETE LIST OF 6502 SOFTWARE, EPROM FIRMWARE (FOR KIM, SUPERKIM, AIM, SYM, and APPLE) AND 6502 DESIGN CONSULTING SERVICES AVAILABLE . . . .

Eric Rehnke
1067 Jadestone Lane
Corona, CA 97120

Now Available For KIM, AIM, And SYM

www.commodore.ca

its original value when leaving your program or avoid using any of the trig functions unless you properly re-attach the trig function object code. The original values of 196 and 197 are 2 and 208, respectively.

Theory Of Operation

The key to understanding how this technique works is in knowing the three ways that the Microsoft BASIC interpreter modifies the pointer to the start-of-variables ($7D and $7E in the SYM-1), and in realizing that the pointer to the start-of-program ($7B and $7C in the SYM-1) never gets modified once it is initialized by a cold start. In the SYM-1 the BASIC program always begins at location $0201 and there is a mandatory zero-byte at location $0200 which is put there only during cold start.

The first way that the interpreter modifies the start-of-variables pointer is through the NEW command which sets the pointer to a value that is equal to the start-of-program pointer plus two ($0203 in the SYM-1). This reduces the size of the BASIC program to two bytes which the NEW command clears to zeroes. In addition to being executed by a direct or indirect command or by a cold start, the NEW command is also automatically executed any time a tape LOAD command results in a BAD LOAD. This is why STEP 11 is required in the above example.

The second way the interpreter can modify the start-of-variables pointer is when a tape LOAD command results in the file being LOADED correctly. In this case, the pointer is set to one greater than the location of the last byte in the tape file and the other required pointers are updated with the NEW command. This is why it is possible to LOAD the machine code after the BASIC program and allow the interpreter to automatically adjust the pointers to allow you to SAVE and RUN the combined program. This is also why the Synertek Tech Note for using trig functions states that you must type either NEW or LOAD x after loading the file containing the trig object code into the top of your RAM space. If you didn't the variables would reside in non-existent RAM!

The third occasion in which the interpreter modifies the start-of-variables pointer is when a new line of a BASIC program is entered, although not in the way you might expect. After the interpreter finds the place in memory where the new line is to go, it calculates the change in the number of bytes that the new line will cause, either plus or minus. It then shifts memory by this amount beginning with the next line in the BASIC program and ending with the byte just before the start-of-variables. Next it updates the start-of-variables pointer by the same amount and then copies the new line into place. The important thing to note is that the interpreter is not influenced by the actual end of the BASIC program (the three zero-bytes) when it moves memory, so the machine code gets moved too. This is why it is necessary to reLOAD the machine code whenever a change is made in the BASIC program.

Conclusion

Now that you SYMmers know how easy it is to combine BASIC and machine-language programs, how about some neat utilities for BASIC? The rest of you can try this same technique on your own machines to see if it will work. Maybe someone with access to alot of different micros can publish a list of those that will and won't allow this technique to work. ©

SYM (AIM) Hi-Speed Tape Revisited

Gene Zumchak

Only a few days after I mailed in the article on SYM's high speed tape, and how loading might be improved by tweaking the value of HSPBDY, I received issue #3 of SYM-PHYSIS, the SYM Users Group newsletter. It contained an interesting note by Jay C. Sinnett, U.S. Environmental Protection Agency, South Ferry Road, Narragansett, R.I. 02882. He claimed that the volume range for loading SYM tapes could be expanded by making a hardware modification. Figure 1. shows his mod. He merely reconnected the clipping diodes so that clipping action does not occur until a diode drops above +5, and below ground. He explains that for many recorders, the amplitude of the positive and negative going peaks is not always equal, or constant. The diodes as connected allow charge to be trapped on C16 which changes the threshold point.

I made the change on my SYM and the results were amazing. Previously, I was only able to read in tapes with the volume level on my recorder at 7 plus or minus one-half. After the change, I could load from levels of 1 to 8. On another SYM, I was unable to load tapes at all. I made the change and was able to load tapes consistantly, and over a wide range of volume settings. Since the AIM and SYM tape circuits are similar, particularly in regard to the connection of the diodes, AIM users with marginal tape reading might also benefit from the mod.

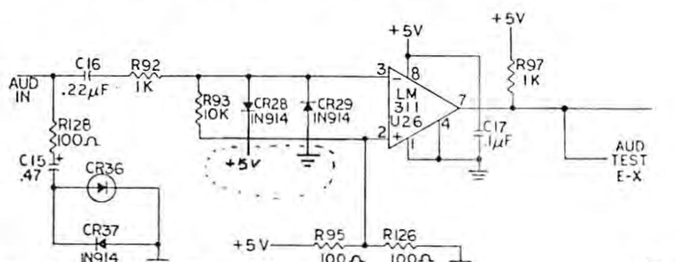


Figure 1. SYM Tape Hardware Mod ©

www.commodore.ca

www.commodore.ca

CAPUTE!

Wherein We Acknowledge Recent Goofs

The Group of Us

From James Bruun, author of "Reading The Atari Keyboard On The Fly", (Issue #6,p.81) comes this missing portion of the program:

```
4999 END
5000 CHAR=0
5010 IF PEEK(764)<>255 THEN GET #1,CHAR:
     CHAR$=CHR$(CHAR):POKE764,255
5020 RETURN
```

, along with these comments:

"Cell 764 is POKEd with 255 only if it isn't already 255. This is to prevent the case of reading the cell, finding it 255, then having a key pressed while the POKE 764,255 instruction is being interpreted. This would cause the keystroke to be lost. In a long program the keystroke isn't often lost, but in a short program it happens quite often."

And from A.M. Mackay, regarding his article "SYM-1 Home Warning System" (Issue 3, compute II, page 26):

"In line 0240 of the program listing, "LDA" should be changed to read "STA" so that the line reads

0240 STA STATUS + 2: FOR INPUT

The missing program listing from Charles L. Stanford's compute II article "Fast Graphics On The OSI C1P" is printed in this issue's OSI Gazette.

EDUCATORS...Are You Using Microcomputers?

A major publishing company is seeking reviewers of CAI Software for grade levels K-12.
Reviewers should have experience with classroom use of one or more microcomputing systems (PET, Apple, TRS-80, etc.).

Write:
Dept. A
900 Sylvan Avenue
Englewood Cliffs, New Jersey 07632

These corrections are for Steven Schulman's Issue 6, Atari article "What to Do If You Don't Have Joysticks":

The arrows weren't drawn into the text on page 75. Here is what should have been presented for the first group of four values.

[up-arrow] = 14
[down-arrow] = 15
[right-arrow] = 7
[left-arrow] = 6

The corresponding arrows match the respective shifted locations in the second chart on page 75.

The second correction consists of swapping two values in lines 1070 and 1090:

line 1070 I = 71
line 1090 I = 70

At last but not least, the program line 20 on page 72 should read **PEEK(560)** rather than **500.** ©

Base Converter

Fred D. Bean

When my 12 year old brother recently requested my assistance on some of his "New Math" homework, I immediately looked to my PET to rescue me from the situation. My brother's assignment was to convert a list of decimal numbers to various other number bases, a chore required often of computer enthusiasts.

The following BASIC program will convert decimal numbers to any base from base 2 (Binary); to base 16 (Hexidecimal). It can be used as shown, or easily modified to allow a child to check his assignments without giving away the answers.

```
100 PRINT"ENTER THE NUMBER TO BE ¬
      ¬CONVERTED"
110 PRINT"AND THE BASE TO WHICH IT WILL ¬
      ¬BE"
120 PRINT"CONVERTED."
130 INPUT NUMBER,BASE
140 IF BASE<17 AND BASE>1 THEN 160
150 PRINT"BASE MUST BE BE 2-16":GOTO 100
160 ARRAY$="0123456789ABCDEF"
170 RECALL=NUMBER
180 A=INT(NUMBER/BASE)
190 B=NUMBER-(A*BASE)
200 NUMBER=NUMBER-(A*16)
210 GOSUB 500
220 IF A=0 THEN 600
230 NUMBER=A
240 GOTO 180
500 A$=MID$(ARRAY$,B+1,1)
510 B$=A$+B$
520 RETURN
600 PRINT:PRINT RECALL;"BASE 10= ";B$;" ¬
      ¬BASE ";BASE:PRINT
610 A$="":B$="":GOTO100
READY.
```

©

www.commodore.ca

PET TWO-WAY RS-232 and PARALLEL OUTPUT INTERFACE

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
- Transfers programs between PETs over the phone line using a modem.
- Selectable reversal of upper and lower case
- PET IEEE connector for daisy chaining
- 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

Order direct or contact your local computer store.

CONNECTICUT microCOMPUTER, Inc.
150 POCONO ROAD
BROOKFIELD, CONNECTICUT 06804
TEL: (203) 775-9659 TWX: 710-456-0052

VISA AND M/C ACCEPTED - SEND ACCOUNT NUMBER, EXPIRATION DATE AND SIGN ORDER.
ADD $3 PER ORDER FOR SHIPPING & HANDLING - FOREIGN ORDERS ADD 10% FOR AIR POSTAGE.

DISK DRIVE WOES? PRINTER INTERACTION?
MEMORY LOSS? ERRATIC OPERATION?
DON'T BLAME THE SOFTWARE!

ISO-1 ISO-2

Power Line Spikes, Surges & Hash could be the culprit! Floppies, printers, memory & processor often interact! Our unique ISOLATORS eliminate equipment interaction AND curb damaging Power Line Spikes, Surges and Hash.

- *ISOLATOR (ISO-1A) 3 filter isolated 3-prong sockets; integral Surge/Spike Suppression; 1875 W Maximum load, 1 KW load any socket $56.95
- *ISOLATOR (ISO-2) 2 filter isolated 3-prong socket banks; (6 sockets total); integral Spike/Surge Suppression; 1875 W Max load, 1 KW either bank $56.95
- *SUPER ISOLATOR (ISO-3), similar to ISO-1A except double filtering & Suppression $85.95
- *ISOLATOR (ISO-4), similar to ISO-1A except unit has 6 individually filtered sockets $96.95
- *ISOLATOR (ISO-5), similar to ISO-2 except unit has 3 socket banks, 9 sockets total . . . $79.95
- *CIRCUIT BREAKER, any model (add-CB) Add $ 7.00
- *CKT BRKR/SWITCH/PILOT any model (-CBS) . . . . Add $14.00

PHONE ORDERS 1-617-655-1532

ESP Electronic Specialists, Inc.
171 South Main Street, Natick, Mass. 01760

Dept. C

PET-APPLE-KIM-TRS-80 Computer Interfacing

PET - Bidirectional Serial and Parallel interface. (SADI) $295.00
Microprocessor based. Talk to another computer and a printer at the same time. Transfer programs between PETs. 32 character buffer. RS-232 in and out. Centronics compatible. Much more. Packed with features.

PET - RS-232 Addressable Printer interface (ADA 1400) $179.00
Complete with cables, case and power supply. Cassette with programs included.

PET - Centronics or NEC 5530P SPINWRITER (ADA 1600) $129.00
Complete with case, cables and connectors. Three position switch for upper/lower case, reverse case and upper case only. Works with WORDPRO and BASIC.

PET Word Processor. On tape - $39.50. On disk - 49.50
Compose and print letters, flyers, ads, manuscripts, etc. Uses disk or tape. 30 page manual included.

Analog to Digital Conversion Systems
16 inputs - 8 bits - 0 to 5 volts - 80 usecond conversion time. Read temperature, light levels, voltages, etc. Cabling, power supplies, software included.
PET, APPLE, TRS-80 $295.00
KIM, AIM65, SYM $285.00

Clock, Calendar, Remote Controller (Super X10) for your computer. $295.00
Transmits to all the BSR X10 remote control modules (up to 255 devices). Stores sequences of control commands that can be initiated by time or an external even such as a switch closure. Maintains a month, date, day of week, and year calendar. Stays on when your computer is off. Complete with cable and connectors. PET, APPLE, TRS-80 (specify).

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.

All our products are assembled and tested with a 30 day money back guarantee. 120 day limited warranty. VISA, Mastercard or check. Add $3.00 S&H. Foreign orders add 10%. Mention this magazine and deduct 3%.

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

www.commodore.ca

apple computer Authorized Dealer

FULL SELECTION OF APPLE COMPUTERS AND ACCESSORIES.

IN CANADA

For Personal · Business · Educational applications and more, there's only one place to come for all your APPLE needs.

OFF THE SHELF DELIVERY.

HOUSE oF COMPUTERS inc.

368 EGLINTON AVE. W. (at Avenue Rd.)
TORONTO, ONTARIO M5N 1A2
(416) 482-4336

Atari® 800™ 16K Personal Computer

$125 FREE BONUS OFFER:
8K RAM with purchase before Dec. 31, 1980

ATARI

List $1080

$749

Atari® Peripherals:

400 8 K	$399
410 Recorder	59
815 Disk	1079
810 Disk	499
822 Printer	349
825 Printer	729
830 Modem	144
850 Interface Module	164

Atari® Accessories:

CX852 8K RAM	$89
CX853 16K RAM	139
CX70 Light Pen	59
CX30 Paddle	17
CX40 Joystick	17
CX8100 Blank Diskette	21

To Order:
Phone orders invited. Or send cashiers check or money order. Equipment shipped UPS collect. Pennsylvania residents add 6% sales tax. Equipment is subject to price change and availability without notice.

Software:

CXL4004 Basketball	$29
CX4105 Blackjack	12
CXL4009 Chess	29
CXL4011 Star Raiders™	44
CXL4006 Super Breakout™	29
CXL4005 Video Easel™	29
CXL4104 Mailing List (Tape)	16
CXL4007 Music Composer™	44
CX4110 Typing	19
CX4101 An Invitation to Programming™	16
CXL4015 TeleLink	19
CX8106 Bond Analysis	19
CX8107 Stock Analysis	19
CX8101 Stock Charting	19

Computer Mail Order
501 East Third St.
Williamsport, PA 17701
(717) 323-7921

COMPUTE!

Is Looking For

Qualified Retail Outlets

If You're A **Computer** Dealer Or Newstand Anywhere In The World And You're Not Selling **COMPUTE!**

Drop us a letter or give us a call for full information on our retail plan.

Phone: 919 275-9809
COMPUTE!
200 East Bessemer Avenue
Greensboro, NC 27401

CBM/PET Computers & Such!

Northern Virginia's single-stop source for Commodore computers, printers, disks, software, and factory-authorized maintenance Also providing a wide range of accessories and supplies from other manufacturers.

VMS features:
- o PET/CBM by COMMODORE
- o SELECTRIC Interfaces by ESCON
- o Software by MICRO SOFTWARE SYSTEMS
- o Printer interfaces by TYPEWRITERS UNLIMITED
- o Covers & Uncrashers by INTERNATIONAL TECHNICAL SYSTEMS
- o Books, magazines, paper, ribbons, diskettes, cassettes, parts

CONNECT PET/CBM TO PRINTER: Simply plug into the interface and use your PET/CBM computer with popular "standard" printers, or add a video monitor for a second display. All interfaces are assembled, tested, and waranteed. No software is required.

TU-65C Use NEC, Centronics, Xymec, or other industry standard parallel printers. Works with disk, other IEEE devices attached. $129.95

TU-6514 Use RS-232 (serial) printers, like Heathkit H-14. 300 to 4800 baud (please specify rate desired). Includes power pack. $79.95

TU-PVE Connect your Video Monitor to PET/CBM for second display. Our unique contact extensions leave the user port available for connection of other accessories. $39.95

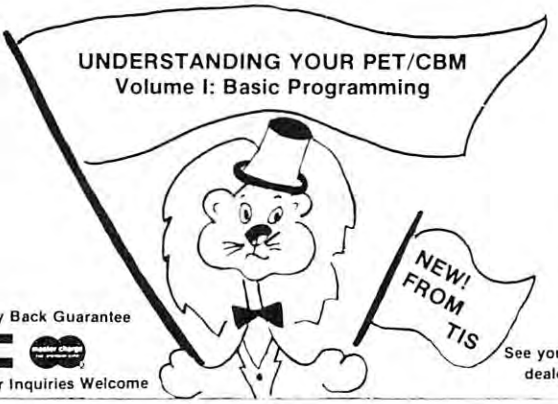
ESCON SELECTRIC Typewriters connnect to PET/CBM, other computers for quality output. We are Northern Virginia's authorized dealer for this nationally known line. Sales, installation, service! Call for details.

MSS Software that has gained an international reputation for ease of use and quality construction. Reviewed in major national magazines, and available for all versions of PET/CBM computer!

Virginia Micro Systems
14415 Jefferson Davis Highway
Woodbridge, Virginia 22191 (703) 491-6502

VISA/MC
VA add 4%
Dealer inquiries invited

www.commodore.ca



Money Back Guarantee

Dealer Inquiries Welcome

See your dealer

New 248-page book includes all the former TIS workbooks except "PET Graphics." Provides information for both ROMs and a comprehensive index. Only $14.95.

Also from TIS

WB-3 PET Graphics $4.95

Software products on cassette or floppy disk with complete instruction manual. Each $24.95 (cassette), $29.95 (diskette).

SW-1 MAIL B mailing list system
SW-2 CHECKBOOK record
SW-3 ACCOUNTS keep track of who owes you how much
SW-4 MEDIT create and maintain date files
SW-5 CALENDAR appointments, meetings at-a-glance

TIS
P.O. Box 921, Dept. C
Los Alamos, NM 87544

Add $2 ($5 foreign orders) shipping and handling

PET and CBM are trademarks of Commodore Business Machines

P^3 = *Powerful PET Products*

from

OPTIMIZED DATA SYSTEMS!

– SOFTWARE –

(Prices include Shipping)

- WORD PROCESSOR (PS-001) $16.95
  Makes documents a snap
- MAILING LIST (PS-002) $16.95
  Throw away your address book
- SPACE EATER (PS-003) $7.95
  Gobbles spaces in BASIC programs
- CATALOG (PS-004) $16.95
  File stamp/coin collections
- SATELLITE TRACKER (PS-005) . $24.95
  Tracks OSCAR Ham Satellite
- MORSE CODE KEYER (PS-006) . . $14.95
  Sends code for real or practice

– HARDWARE –

(*Shipping $1.50 per order)

- 2114 RAM ADAPTER

Replaces up to 8-6550s w/2114s (See COMPUTE #5, p. 81)

PHB-001 Bare PCB $8.95*
PHK-001 Kit. $13.95*
(PCB & Parts for two 2114 sockets only)
PH-001 Assembled $24.95*
(Assembled Kit with one 2114 IC)
PH-001S Assembled $22.95*
(Assembled with eight 2114 sockets)

WRITE FOR DETAILS – NOW!
OPTIMIZED DATA SYSTEMS
P.O. Box 595, Placentia, CA 92670

(PET is a trademark of Commodore Business Machines)

SPECIAL INTRODUCTION ONLY 99.95

FOR THE FINEST PARALLEL INTERFACING ELEMENT FOR THE PET/CBM*

THE P.I.E.-C IS A VERY COMPLETE INTERFACE. IT COMES IN AN ATTRACTIVE CASE AND MOUNTS DIRECTLY TO THE COMPUTER, THUS ELIMINATING THE NEED FOR EXTRA SPACE IN THE BACK FOR BOXES AND MESSY PILES OF WIRE. YOUR SYSTEM CAN LOOK PROFESSIONAL. THE P.I.E.-C HAS COMPLETE ADDRESS SELECTABILITY OF DEVICE NUMBERS 4 THROUGH 30. AND, THERE'S NO NEED TO DISASSEMBLE YOUR UNIT TO CHANGE THE ADDRESS; IT'S AS SIMPLE AS SETTING THE DIP SWITCH. THIS SELECTABILITY ALLOWS BOTH YOUR LETTER-QUALITY PRINTER (NEC SPINWRITER, ETC.) AND YOUR GRAPHICS PRINTER TO BE CONNECTED AT THE SAME TIME BUT USED SEPARATELY BY ADDRESSING EACH WITH A DIFFERENT DEVICE#. NOW THAT'S REAL EFFICIENCY!!

THE P.I.E.-C WILL CONVERT NON-STANDARD PET CODES TO TRUE ASCII CODE. IT CAN BE SWITCHED IN OR OUT OF SERVICE AS NEEDED. FOR INSTANCE, IF YOUR SOFTWARE DOES THE CONVERSION, JUST SWITCH THE CONVERTER OFF. IF YOU NEED THE CONVERSION FOR PRINTING FROM BASIC, JUST SWITCH IT ON.

OUR INTERFACE PROVIDES EXTENSION OF THE IEEE-488 PORT ON THE PET/CBM. YOU DON'T NEED TO BUY OTHER CABLES BECAUSE OUR INTERFACE USES THE SAME CARD EDGE TYPE AS THE COMPUTER. THUS THE CABLE FROM THE FLOPPY DISC TO THE PET/CBM WILL NOW CONNECT INTO THE INTERFACE.

THE P.I.E.-C IS INTERCONNECTED WITH, AND POWERED BY, THE PRINTER USING A 5' DATA CABLE SUPPLIED WITH THE INTERFACE. THIS MEAN THAT THERE IS NO POWER SUPPLY NECESSARY AS WITH SERIAL INTERFACES. THE +5V IS SUPPLIED ON PIN# 18 OF THE CONNECTOR USED BY ALL TRUE CENTRONICS STANDARD INTERFACED PRINTERS. THIS INCLUDES THE CENTRONICS 779, P1, AND MANY OTHERS INCLUDING THE ANADEX PRINTERS.

OUR INTERFACE GOES BEYOND THE CENTRONIC'S COMPATIBILITY. THEREFORE, IT WILL OPERATE THE ANDERSON-JACOBSON AJ841 AND THE "PAPER TIGER" BY INTEGRAL DATA SYSTEMS. IN FACT, IT WILL DRIVE ANY PARALLEL-INPUT PRINTER THAT USES 8 DATA BITS AND 2 HANDSHAKING LINES.

IF YOU ARE INTERESTED IN THE MOST INTERFACE FOR YOUR MONEY, THEN PLACE YOUR ORDER TODAY. OUR INTRODUCTORY PRICE FOR THE P.I.E.-C INTERFACE WON'T LAST LONG. SO, DON'T DELAY . . . , YOU MAY HAVE TO PAY THE LIST PRICE OF $119.95.

PLEASE SPECIFY YOUR PET/CBM TYPE (NEW OR OLD ROMS) AND THE BRAND AND MODEL OF THE PRINTER YOU WILL BE USING. WE ATTACH THE CORRECT CONNECTOR ON THE CABLE.
ALL ORDERS MUST BE PREPAID OR COD CASH. ADD $5 FOR SHIPPING AND HANDLING.
MARYLAND RESIDENTS ADD 5% SALES TAX.

LEMDATA PRODUCTS, P.O. BOX 1080, COLUMBIA, MD. 21044

PHONE (301) 730-3257

*PET/CBM are trademarks of Commodore Business Machines

NEW !!

Programs for Commodore's PET®

- **Business Research** $50
  Make better decisions with this high power MBA business tool. 16k.
- **Home & Small Business** $15–$40
  •Addresser •Inventory
  •Shopper •Dinner's On!
  Each has a built-in printer option.
- **Games & Simulations** $15ea.
  •Fur Trapper •High Seas
  •Mansion! •Pentagon! •Museum!
- **Education Pack** $15
  High School sampler with geometry, algebra & chemistry.

Send for full catalog!

HARRY H. BRILEY

P.O. Box 2913
Livermore, CA 94550
(415) 455-9139

www.commodore.ca

```
*************************************
*************** ATARI ***************
***** RANDOM CHARACTER GENERATOR *****
*************************************


10 CLR
20 ? "}"
30 DIM B$(1):DIM A(255,255)
40 A(X,Y)=INT(RND(1)*91):IF A(X,Y)<65 TH
EN 40
50 B$=CHR$(A(X,Y))
60 PRINT B$;"  ";
70 FOR Q=1 TO 500:NEXT Q
80 GOTO 40


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

IT OCCURED TO ME ONE NIGHT THAT A SIMPLE ROUTINE,SUCH AS THIS,MIGHT PROVE USEFUL AT SOME TIME OR ANOTHER. SO I THOUGHT I WOULD PASS IT ON IN THE HOPES THAT SOME OTHER OF YOUR READERS SHOULD PROVE ME CORRECT.

YOUR MAGAZINE PROVIDES WHAT I CONSIDER A COMMENDABLE ALTERNATIVE. PLEASE KEEP THE MIDNIGHT OILS BURNING!

SINCERLY,

PAUL DOBOSZ

Family Oriented Computer Games

Available for use with the PET
BINGO, DRAW POKER,
CRYPTOGRAMS, WHEEL OF FORTUNE
WORD GAMES

Educational Software

Free educational catalog listing over 140 programs in math, physics, chemistry, vocabulary, spelling. Write to:

Microphys
2048 Ford Street
Brooklyn, N.Y. 11229

Commodore Hardware

Sawyer Software now has available the complete line of Commodore micro-computers, disk drives and printers. With the purchase of a complete CBM system or any piece of Commodore hardware, you can receive up to $390.00 worth of free software. We also have memory kits to expand your 8K PET to a 24K or 32K machine.

GENERAL LEDGER — Maintains complete financial information on diskette. Generates hard copy of Trail Balance, Income Statement (with percentages), Balance Sheet and Check Journal. Includes commands to edit data, close out year end accounts and debit-credit verification.
.... TRS-80 **$125.00**
.... CBM **$195.00**

AP1 is a general ledger package with check journal, income statement, balance sheet and checkbook reconciliation routine. Designed for the small business or homeowner. Up to 50 entries and 40 accounts per period.
.... **$25.00**

AP2 has all the features of AP1, plus up to 250 entries per period, Menu, formatter for reports and more. Requires at least 16K in PET or TRS-80.
AP2 also utilizes a printer for the reports. Send device number of printer with order for PET.
.... **$45.00**

SAWYER SOFTWARE

Payroll computes tax information and updates totals for quarterly and yearly reports. Employees can be salaried or hourly and pay periods can be either weekly, bi-weekly, semi-monthly or monthly. Any number of employees (8 per cassette for PET, 25 employees per cassette for TRS-80).
.... **$30.00**

Payroll-16K for PET. Same as above plus ability to get hard copy of individual employee records and stores 25 employees per cassette.
.... **$45.00**

Printer Payroll (TRS-80) — Includes same features as PAYROLL, plus utilizes a printer for hard copies of employees records and stores 25 employees per cassette.
.... **$45.00**

SAWYER SOFTWARE OFFERS a complete line of cassette and disk based business software for your PET or TRS-80 computers. Each program has been specifically designed for ease of use and come with instruction manuals. SAWYER SOFTWARE'S disk packages run on Commodore, Compu-Think or TRS-80 disk systems with 32K and at least one disk drive. Call or write for free brochure.

Atari Hardware

SAWYER SOFTWARE carries the complete line of Atari's micro-computers and peripherals for your business, educational and entertainment needs. Call us for more details on our low prices for Atari's fully programable, color graphic personal computer.

DISK PAYROLL — Computes FICA, Federal and State taxes. State is calculated on percentage of gross pay and can be customized for your particular state. Will allow use of City tax and/or other deducations. Utilizes printer for hard copy of Payroll Register, including current, quarterly and year-to-date totals.
.... TRS-80 **$125.00**
.... CBM **$195.00**

*** NEW FOR YOUR PET ***
EXATRON STRINGY FLOPPY SOFTWARE

ESFAP2 — This program provides general ledger and complete financial information utilizing the ESF wafer for input output storage. Has all the features of our PET cassette based AP2 program.
.... **$49.95**

ESFGL — This program provides the same complete financial information as our disk based GL package. ESF wafer is used for input/output storage of all data.
.... **$149.95**

SAWYER SOFTWARE

201 Worley Rd.
Dexter, Mo. 63841
(314) 624-7611

master charge THE INTERBANK CARD

VISA

Telephone Orders Welcome

Dealer Inquires Invited

www.commodore.ca

PET/ATARI

PET KRK-1

A HARDWARE REPEAT KEY FOR THE PET NUMBER/CURSOR PAD — A BOON TO THE BUSY PROGRAMMER AND THE AMBITIOUS GAMES PROGRAM WRITER. AIDED BY THE DETAILED INSTRUCTION PROGRAM THE AVERAGE PET OWNER CAN UPGRADE HIS MACHINE IN A COUPLE OF MINUTES.

- CUTS SCREEN EDITING BY 75%
- GIVES JOYSTICK PRECISION TO THE NUMBER PAD
- INSTALLS IN MINUTES
- COMPLETE WITH SOFTWARE

$39.95

PET KRK-2

THREE MAIN FEATURES — A DEFINITIVE FULL KEYBOARD REPEAT KEY. A WARM KEYBOARD RESET FROM OTHERWISE FATAL CRASHES. A SELECTABLE TONE WHICH ALLOWS TOUCH ENTRY AT OTHERWISE IMPOSSIBLE SPEEDS. ALL THIS PLUS THE MICE TRO MUSIC PROGRAM, UNQUESTIONABLY THE BEST MUSIC PROGRAM FOR THE PET AVAILABLE.

- WARM KEYBOARD RESET
- SELECTABLE KEYBOARD TONE FOR TOUCH ENTRY
- DEFINITIVE FULL KEYBOARD REPEAT KEY
- MUSIC — INCLUDING THE MICE TRO PROGRAM
- COMPLETE WITH SPEAKER

$119.50

PET KC-1

A TOTALLY NEW CONCEPT IN PET'S COMMUNICATION WITH THE OUTSIDE WORLD. AT LONG LAST THE PROGRAMMER IS FREE FROM THE LIMITATIONS OF THE IEEE BUS. INPUT, PRINT, LOAD AND SAVE TO AN EXTERNAL RS232 DEVICE. A COMPREHENSIVE ON-BOARD FIRMWARE PACKAGE ALLOWS ALL MANNER OF HITHERTO IMPOSSIBLE FUNCTIONS TO BE ACHIEVED INCLUDING KEYBOARD SELECTABLE CONFIGURATION WITH FULL MODEM CONTROL. CALL FOR A COMPLETE INTRODUCTION BOOKLET.

$299.00

ATARI AT-16

A FULL 16K MEMORY BOARD FOR THE ATARI 800 COMPUTER. JUST PLUG IN, COMPLETELY COMPATIBLE WITH ALL EXISTING HARDWARE AND SOFTWARE. NO SPECIAL SOFTWARE ROUTINES, JUST PLUG IT IN AND GO.

$119.50

MICROTEK inc.

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

www.commodore.ca

A First Look at the TRS-80 Color Computer

David D. Thornburg
Innovision
P.O. Box 1317
Los Altos, CA 94022

!!! ...

Yes, you are still reading COMPUTE - don't worry. There are several reasons for having a review of this new computer in this magazine. First, the TRS-80 Color Computer, at a price of $399, is probably going to be among the most popular computers ever made. Second, some industry wags contend that this computer threatens the Atari 400 - a view I do not share. Also, since I have historically avoided Radio Shack computers, yet rushed to get this one, you might want to know what the excitement is all about.

The TRS-80 Color Computer is not only the first Radio Shack computer with no Roman numerals in its name, it is also their first true "consumer" computer. I feel that this application can be given to a product, which can be set up and used by the average nine year old child. For example, the set up manual devotes five pages to hooking the computer up to the television set (the built-in modulator is switch selectable for channel 3 or 4). This attention to detail is marvelous and will attract numerous first-time computer users.

The TRS-80 Color Computer, at a price of $399, is probably going to be among the most popular computers ever made.

Another strong point which will help get this computer into people's homes is its excellent styling. The cabinet looks a bit like a silver Apple. Gone are the klunky boxes and external transformers of the Model I. Everything is self-contained. The keyboard has medium travel keys which have a distinct over-center snap to them. The keytops are sculptured and placed for easy finger positioning which makes this keyboard easier to use than a membrane keyboard or the keyboard on the old PET, for example. Radio Shack chose to use slightly smaller keytops than normal, thus allowing the use of keyboard overlays similar to those used with the Interact computer. This requires that you be somewhat accurate in finger positioning, but it doesn't slow you down that much. I do find the noise from the keyboard to be a bit annoying - somewhat like typing on a plate full of pennys - but overall their feel is quite adequate.

The power switch, RESET button and all interfaces emerge from the back of the computer. The user is provided with cassette I/O, two joystick ports, and an RS-232 connection. I consider the RS-232 port to be one of the most important features of the computer since it makes it easy for a user to connect this computer to information utilities like the Source.

The user is provided with cassette I/O, two joystick ports, and an RS-232 connection.

Technical Details

The TRS-80 Color Computer uses the 6809E microprocessor (no, I don't know why they still call it a TRS-80) which is run at 0.895 MHz - a fraction of its top speed. The computer came with 4k Bytes ofRAM and an 8k Byte BASIC from Microsoft (which appears to start a location 40960, giving lots of room for ROM expansion). Soon one will be able to get this machine with 16k of RAM and a new 16k BASIC, but the 4K system is the only one being shipped as this is being written.

I felt like I had gone back to 8 crayons after knowing that boxes of 128 were available elsewhere.

I had expected to see a fixed point BASIC with limited string capability. Instead I found a floating point BASIC with string arrays, MID$, IF-THEN-ELSE, PRINT@ (for printing at a given screen address), and lots of other useful things. This BASIC also supports limited graphics (64 x 32 dots in 8 colors). I will say more about this later. Users are given a SOUND command which produces a single tone through the TV loudspeaker. This tone can be varied over more than three octaves. The principal limitations in the BASIC are the use of short variable names (2 characters) and the absence of exponentiation and all the trig functions except SIN. For many home applications, this shouldn't present too much of a problem, however, and I am glad that string manipulations were not cut to save space.

Those of you who are Atari users will be disappointed in the TRS-80's color graphics. I felt like I had gone back to 8 crayons after knowing that boxes of 128 were available elsewhere. The display looks identical in format to that on the APF Imagination Machine - 32 characters by 16 lines of upper-case text. Since Motorola was active in the design of both the APF and Radio Shack computers, this is not surprising. What I found distressing is that the background color for text is set to green, and that the numeral 4 is hard to read. However, the image quality is quite good compared to that of the APF computer. This may be a result of the precautions Radio Shack had to take to make this computer meet

www.commodore.ca

the new FCC rules.

The graphics capability is provided through a firmware shape table of mosaic characters arranged on a 2 x 2 grid. Each element in the grid can be "on" or "off". Elements which are "off" are black, and the "on" elements can have any of eight colors, provided that all "on" elements in a given shape have the same color. Plotting only works well when the background is black. The graphics commands (SET and RESET) take the work out of finding the right mosaic character and give the user access to any of the 64 x 32 picture elements. Text may be mixed with graphics if desired.

As more RAM and a new ROM is added to this computer, I would not be surprised to see the graphics resolution expand to 256 x 192 picture elements - time will tell.

The tape data rate is listed at 1500 baud - quite impressive.

An external cassette recorder (not included in the $399 price tag) can be used to save programs. Each program on the tape can have a name to facilitate searching when several programs are on a single tape. The tape data rate is listed at 1500 baud - quite impressive. The SKIPF command allows users to skip files when positioning the tape for a new entry. PET owners often use the VERIFY command for this purpose since it advances the tape past a program without altering either the program in the computer or the listing on the tape. Once I got the volume control set properly on the tape recorder, I had no trouble reading programs at all.

Competition ??

It has been said by more than one industry observer that the new Radio Shack entry will provide stiff competition to Atari. Based on my experience with quite a few computers, I conclude that either I have missed something in the TRS-80 Color Computer, or that most of these industry observers have never gotten familiar with the Atari computers. The only area of importance in which the Radio Shack machine has any advantage over the Atari 400 is the keyboard. Atari graphics and sound stand in a class by themselves. While I do prefer the Microsoft to the Atari BASIC, the use of plug-in cartridges for all firmware makes the Atari easier to upgrade.

We can expect their sales to be quite formidable.

If the Radio Shack entry is going to slow down computer sales for anyone, I would have to guess that APF and Mattel will be in for the greatest pressure. At $400, the TRS-80 Color Computer outperforms my 6800-based APF computer, and looks much nicer in the home as well. The Mattell computer is probably going to be targeted toward the same market with a much higher price tag.

This doesn't mean that Commodore, Apple and Atari won't feel some pressure from this computer, however. Besides the price advantage, the TRS-80 Color Computer holds its own on speed with the rest of the pack, and in some cases it even gets some speed records of its own.

As an example, I ran the following program on the TRS-80 Color Computer, a PET, an Atari 800 and an Apple with integer (I) and floating point (F) BASIC:

```
5 FOR J = 1 TO 100
10 FOR I = 1 TO 100
20 A = (I*I)/I
40 NEXT I
50 NEXT J
```

'The execution times (in seconds) are shown below for each computer.

TRS-80 COLOR COMPUTER:	103 SEC.
APPLE (I):	52 SEC.
APPLE (F):	75 SEC.
PET:	85 SEC.
ATARI 800:	159 SEC.

As can be seen from these figures, the new Radio Shack product is slower in executing this program than all but the Atari computer.

Next, I added one line to the program:

```
30 PRINT A
```

and ran the experiment again with the following results:

TRS-80 COLOR COMPUTER:	280 SEC.
APPLE (I):	250 SEC.
APPLE (F):	340 SEC.
PET:	500 SEC.
ATARI 800:	540 SEC.

This time the Radio Shack entry is the leader of the floating point BASIC rate, being beat only by the Apple integer BASIC.

This product is definitely not a toy, however, and I doubt that it will ever be called a "trash-80"

With more than 7000 stores selling this computer, we can expect their sales to be quite formidable. I feel that there is room for this computer in the marketplace without it necessarily cutting heavily into Apple and Atari sales. This product is definitely not a toy, however, and I doubt that it will ever be called a "trash-80" as its predecessor has been.

I wonder what's next? ©

Programmers:

Send your short and simple BASIC utility programs with a brief descriptive note to:

COMPUTE!
P.O. Box 5406
Greensboro, NC 27403 USA
Attn: Basically Useful BASIC

www.commodore.ca

Index to Advertisers

Program Listings for COMPUTE

Cursor control characters will appear in source listings as shown below:

```
h=HOME           , ĥ=CLEAR SCREEN
↓=DOWN CURSOR    , ↑=UP CURSOR
→=RIGHT CURSOR,  ←=LEFT CURSOR
r=REVERSE        , r̂=REVERSE OFF
```

Graphics (i.e. shifted) characters will appear as the unshifted alphanumeric character with an underline. This does not apply to the cursor control characters. The Spinwriter thimble doesn't have a backarrow symbol, so a "~" is used instead.

The "¬" is used to indicate the beginning of a continuation line. It is also used to indicate the end of a line which ends with a space. This prevents any spaces from being hidden. ©

www.commodore.ca

SEND FOR OUR FREE CATALOG

COMPUMART LETS YOU PUT APPLE ON TRIAL

WITH OUR EXCLUSIVE 10-DAY FREE RETURN ON ALL THESE GREAT APPLE COMPUTERS AND ACCESSORIES.

CompuMart carries complete lines of microcomputers from Apple to Zenith, as well as thousands of microcomputer peripherals and accessories. Write for our FREE 36 page catalog.

Apple Computer

We carry the most complete inventory of Apple computers, peripherals, and software. CALL!

Our Christmas Apple Special: Save over $250 on our most popular Apple System. System includes a 48K Apple II, Apple Disk & Controller, and a Sup R Mod RF Modulator.

List: **$2,020**

Compumart Sale Price: **$1,769**

New from Apple for the Apple II:

DOS 3-3 Convert disks to 16 sector format for 23% more storage and faster access **$60**

Apple Plot. The perfect graphic complement for Visicalc **$70**

Dow Jones News & Quotes **$95**

Adventure (Uses 48K) **$35**

DOS Tool Kit **$75**

Apple Fortran **$200**

Silentype Printer w/Xface **$595**

Visicalc **$149**

Tax Planner **$120**

From Symtech & Info Unlimited

Super Sound Generator ..(mono) **$159**

(stereo) **$259**

Light Pen **$249**

X-10 Controller (plugs into paddle port) **$49**

Apple Sync Controller **$49**

From Personal Software

Visicalc **$149**

Desk top plan **$99**

New from Videx! — Video Term

80 Col. x 24 line

7 x 9 matrix, plug in compatible board for the Apple II. Price **$325**

without graphics EPROM.

With graphics EPROM .... **$350.**

New from MUSE

The Voice **$39.95**

Super Text **$99.00**

Address Book **$49.95**

Mountain Hardware — Expansion accessories for your Apple

Introl/X-10 System **$289**

Super Talker **$299**

The Music System **$545**

ROM plus board w/keyboard filter **$199**

Clock Calendar **$280**

16 Channel A to D Converter **$350**

Apple Expansion Chassis **$650**

ROM Writer **$175**

Miscellaneous Apple II Accessories:

Easy Writer (80 col, need Videx) .. **$249**

Easy Mover **$49**

Easy Mailer **$69**

Dysan Diskettes .... **ea. $5**

S.S.M. Serial & Parallel Apple Interface **$225**

ABT's Numeric Key Pad **$110**

270 THIRD ST. DEPT 30 **COMPUMART** CAMBRIDGE, MA 02142

TOLL FREE 1-800-343-5504 **IN MASS 617-491-2700**

apple computer
Authorized Dealer

www.commodore.ca

The Great American Solution Machine.

Meet Commodore. The business computer that's providing solutions for more than 100,000 people all over the world. Built by one of the pioneers in office machines. With a reputation for quality that can only come from vertical integration of the total manufacturing process. Commodore builds, not assembles.

Compare Commodore's word and data processing capabilities with computers costing twice or even three times as much. You'll see why so many small businesses are turning to Commodore for solutions to problems as varied as these:

☐ A car leasing company's customers were terminating too early for account profitability. Solution: A 16K Commodore. It analyzes cash flow on over 1200 accounts, identifies those for early penalties, and even writes up lease contracts. Commodore paid for itself within weeks.

☐ A law firm needed a high quality, easy-to-use, affordable word processing system. Solution: Commodore plus its WordPro software package. At a $6,000 savings.

☐ A gasoline retailer needed to inventory, order and set prices; determine Federal and state income taxes; and comply with Federal pricing and allocation regulations. All done daily, weekly, monthly and yearly. Solution: Commodore. It keeps his business on track—and Uncle Sam off his back.

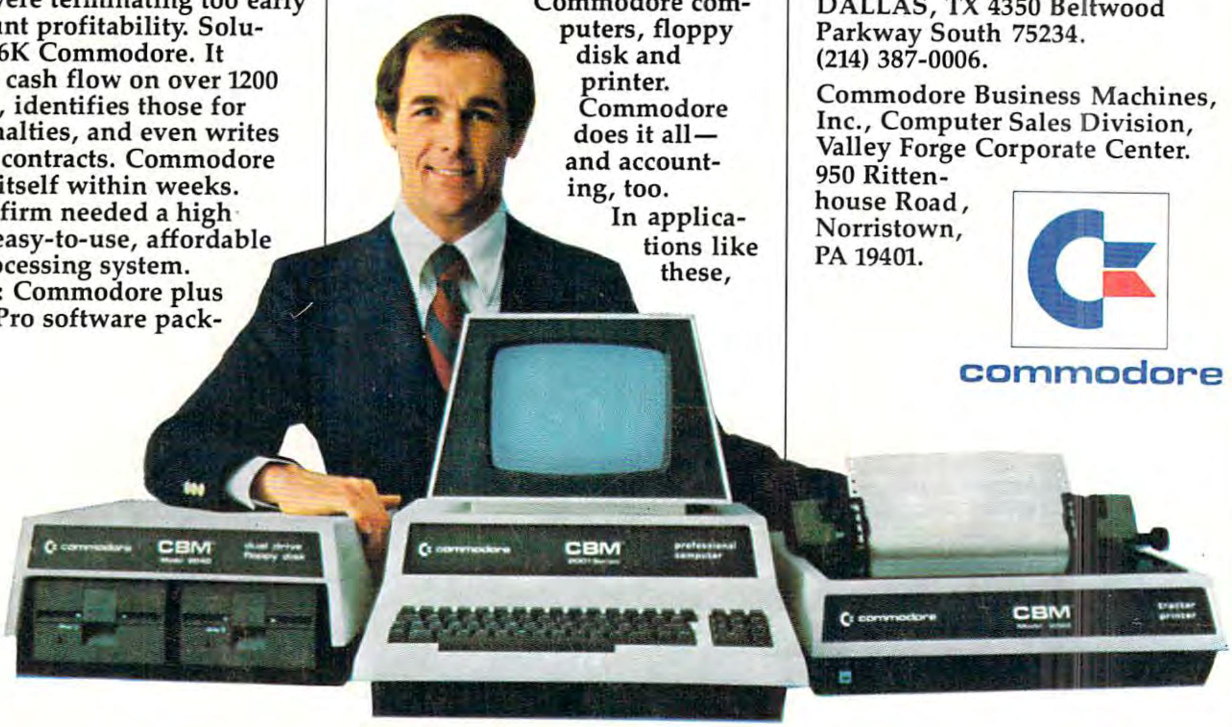
☐ A paint and wallpaper store had to inventory over 600 expensive wallpaper lines for profitability, monitor distributor sales, set and track salesmen's goals, and help the customer select the right size, pattern and quantity. Solution: Two 32K Commodore computers, floppy disk and printer. Commodore does it all—and accounting, too.

In applications like these, and many more, Commodore solves the problems that stand in the way of increased profitability. Commodore can provide the solution in your Great American business, too. Find out more by calling or writing any of Commodore's District Sales Offices.

COSTA MESA, CA 2955 N. Airway Avenue 92626. (714) 979-6307.
SANTA CLARA, CA 3330 Scott Boulevard 95051. (408) 727-1130.
DECATUR, GA 5360 Snapfinger Woods Drive 30035. (404) 987-3311.
BENSENVILLE, IL 790 Maple Lane 60106. (312) 595-5990.
NORRISTOWN, PA 950 Rittenhouse Road 19401. (215) 666-7950.
DALLAS, TX 4350 Beltwood Parkway South 75234. (214) 387-0006.

Commodore Business Machines, Inc., Computer Sales Division, Valley Forge Corporate Center. 950 Rittenhouse Road, Norristown, PA 19401.

commodore



COMMODORE

www.commodore.ca