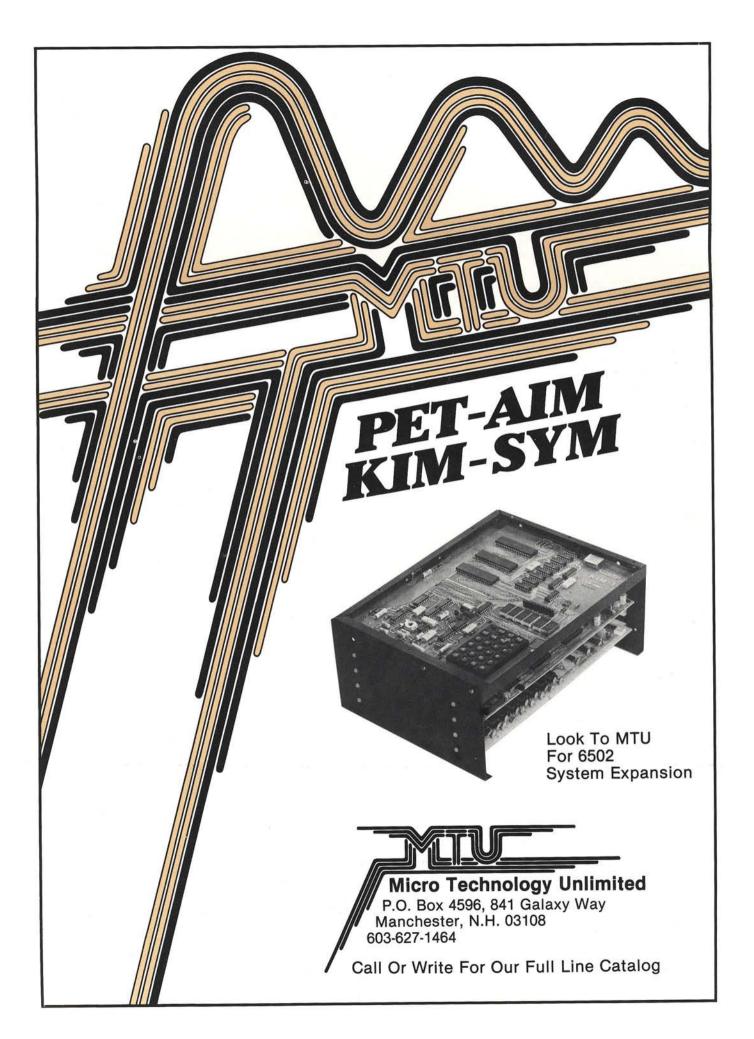
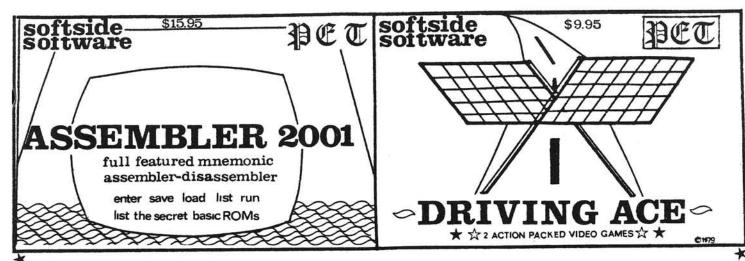
I AIM SYM KIM APPLE ATARI PET OSI AIM SYM KIM APPLE ATARI APPLE KIM ATARI OSI AIM PET APPLE SYM KIM ATAI AIM SYM KIM ATARI APPLE PET OSI AIM SYM KIM ATARI APPLE PET OSI APPLE PET OSI AIM SYM KIM ATARI PET APPL T APPLE ATARI KIM SYM AIM PET ATA<u>RI A</u>PP<u>LE OSI A</u>IM SYM KIM APPLE ATARI PET OSI AIM SYM KIM ATARI APPLE PE T SYM KIM AIM ATARI OSI PET AIM SY PLE ATARI KIM SYM OSI PET SIM KIM E ATARI OSI SYM KIM AIM KIM PET I PET AIM SYM ATARI KIM PET OSI AF A APPLE ATARI PET OSI AIM SYM KIM A T OSI PET AIM SYM KIM APPLE PET OSI PET A PET APPLE PET SYM KIM APPLE PET OSL KIM ALM ATARI OSI PET AIM SYM KIM PET APPLE ATARI PE A PET SIM KIM AIM PE T AIM SYM KIM A ARI SYM KIM APPLE PE A SYM OSI OSI APPLE PET AIM SYM PET A M KIM ATARI PETKIM PET APPLE ATARI OSI PET SYM KIM ATARI PET APPLE ATARI OSI PET SYM KIM ATARI PET APPLE



I SYM KIM APPLE PET OSI AIM SYM KIM APPLE PET OSI OSI PET APPLE ATAHI KIM SYM AIM PET ATAHI APPLE I SYM KIM APPLE ATARI APPLE KIM ATARI OSI AIM PET APPLE SYM KIM ATARI APPLE KIM ATARI OSI AIM PET APPLE SYM KIM ATARI APPLE PET OSI APPLE PET OSI AIM SYM KIM ATARI PET APPLE CPLE ATARI KIM SYM AIM PET ATARI APPLE OSI AIM SYM KIM APPLE ATARI PET OSI AIM SYM KIM ATARI APPLE PET OSI PET APPLE ATARI PET OSI APPLE PET OSI APPLE PET OSI APPLE PET OSI APPLE ATARI PET OSI APPLE PET OSI APPLE PET OSI APPLE PET OSI APPLE ATARI APPLE ATARI APPLE ATARI KIM SYM ATARI KIM PET OSI APPLE ATARI APPLE PET OSI PET ATARI APPLE OSI SPLE ATARI PET OSI AIM SYM KIM APPLE ATARI APPLE OSI SPLE ATARI PET OSI AIM SYM KIM APPLE ATARI APPLE ATARI OSI AIM PET APPLE SYM KIM ATARI APPLE ATARI ATARI APPLE ATARI APPLE ATARI ATARI ATARI A





# PET BASIC BREAKTHROUGH Softside Software presents SYMBOLIC/STRUCTURED BASIC

At last, Symbolic/Structured Basic is available for your PET 8-32K personal computer! S-Basic is a pre-compiler that enhances the PET's built-in basic monitor with the addition of extra-control statements found only in the most sophisticated computers.

WHILE ... GOSUB ... calls a subroutine as long as a condition is true. UNTIL ... GOSUB ... jumps to a subroutine unless the condition is true. The IF ... Then ... ELSE statement allows the programmer to command the computer to execute instructions if the normal IF condition is not met.

Forget about line numbers, S-Basic allows you to program naturally only naming (numerically or alphabetically!) statements that you will need to refer to, for ex-

ample: LOOP/PRINT "HI": GO TO LOOP. S-Basic program lines can be up to 255 characters long, two-and-one-half times as long as on standard Basic. S-Basic does not compromise any of PET Basic's existing features. All PET Basic commands can be used.

S-Basic includes an **editor** with full text capabilities, a **translator/pre-compiler** with its own error messages, and the **S-Basic loader**. These programs are recommended for disk-based PETs. A printer is optional but suggested. Cassette copies are available and require two cassette drives. Comprehensive instructions are included. Symbolic-/Structured Basic package is available complete for an **introductory price** of **\$35.95**.

- WHILE .... GOTO ....
- IF....THEN....ELSE
- 255 CHAR. LINES

#### A PET

PROGRAMMING BREAKTHROUGH

- UNTIL .... GOTO ....
- SYMBOLIC OPTIONAL
- LINE NUMBERS

305 RIVERSIDE DRIVE, NEW YORK, NEW YORK 10025

# **PERFECT AIM**



# ATTRACTIVE FUNCTIONAL PACKAGING FOR YOUR AIM-65 MICROCOMPUTER

- Professional Appearance
- Striking Grey and Black Color Combination
- Protects Vital Components

# ENGINEERED SPECIFICALLY FOR THE ROCKWELL AIM-65

- All Switches Accessible
- Integral Reset Button Actuator
- Easy Paper Tape Replacement

#### **EASILY ASSEMBLED**

- Absolutely No Alteration of AIM-65 Required
- All Fasteners Provided
- Goes Together in Minutes

# MADE OF HIGH IMPACT STRENGTH THERMOFORMED PLASTIC

- Kydex 100\*
- Durable
- Molded-In Color
- Non-Conductive

#### AVAILABLE FROM STOCK

- Allow Three to Four Weeks for Processing and Delivery
- No COD's Please
- Dealer Inquiries Invited

TO ORDER: 1. Fill in this Coupon (Print or Type Please)
2. Attach Check or Money Order and Mail to:

@\$43.50 each
California Residents Please Pay

\$46.33 (Includes Sales Tax)

SAE 1-2 PLEASE SHIP PREPAID \_\_\_\_\_\_ SAE 1-2(s)

@\$46.50 each California Residents Please Pay \$49.52 (Includes Sales Tax)

# enclosures group

771 bush street san francisco, california 94108



#### April 1980 Issue Number 23

#### **Table of Contents** Staff Editor/Publisher Robert M. Tripp **Applesoft II Shorthand** 5 **Associate Editor** by Allen J. Lacy Mary Ann Curtis 9 Editorial **Business Manager** The APPLE Stripper 11 Maggie E. Fisher by Bill Crouch **Circulation Manager** Graphics and the Challenger C1P, Part 4 15 Carol A. Stark by William L. Taylor Art/Advertising Coordinator SYMple BASIC Data Files 21 Terry Spillane John M. Blalock Comptroller A Perpetual Calendar Printer for the AIM 27 Donna M. Tripp by Mel Evans **Production Assistant Bi-directional Scrolling** 31 L. Catherine Bland by Roger Wagner MICRO Lab The SY6516 Pseudo-16 Bit Processor 36 R. Keith Beal by Randall Hyde 43 MICRO<sup>TM</sup>is published monthly by MICRO INK, Inc., Chelmsford, MA 01824. Tel. 617/256-5515. **PET Keysort** by Rev. James Strasma KIM Scorekeeper 59 Second Class postage paid at Chelmsford, Ma 01824. by Joel Swank Publication Number: COTR 395770. Circulation: Paid subscriptions: U.S.: 3800, Foreign: 350; Dealers: U.S.: 4500, Foreign: 1900. OSI BASIC in ROM - What's Where? 65 by Earl D. Morris, Jr. Subscription rates: U.S.: \$15 per year. 69 Letterbox Foreign, surface mail: \$18 per year. The MICRO Software Catalogue: XIX 71 For air mall rate, change of address, back issue or subscription information write to: MICRO, P.O. Box 6502, Chelmstord, MA by Mike Rowe 6502 Bibliography: Part XIX 77 by William R. Dial Entire contents Copyright © 1980 by MICRO

#### Advertiser's Index

Aardvark	79	Instant Software, Inc.	74.75
AB Computers	56	Micro Technology Unlimited	IFC,26
Andromeda Computer Systems	22	NIBBLE	13
Automated Simulation	34	Powersoft, Inc.	10
Avante Guard Creations		Programma International	35
	19		19
Beta Computer Devices	63	Progressive Computing	56
CJM-Industries	68	Progressive Computer Software	
Classified Ads	8,12,45	Progressive Software	4
Computer Applications Tomorrow	79	Rainbow Computing	IBC
The Computerist, Inc.	57	RAYGAM	73
Computer Shop	62	RNB-Enterprises	20
		Shepardson Microsystems	39
Connecticut microComputers	40,41,70,79	SKYLES Electric Works	42,67
Creative Computing	64		
Cyberdyne	29	Small Business Computer Sys.	63
Dakin 5	38	Softape	BC
Decision Systems	56	Softside Publications	30
Digital Engineering Assoc.	80	Softside Software	1
Discount Data Products	80	Southeastern Software	58
		Southwestern Data Systems	70
Electronic Specialists, Inc.	38		70
Enclosures Group	2	Stoneware	
Galaxy	80	Synergistic Software	33
Hudson Digital Electronics	76	United Software of America	38
Information Unlimited Software	14		

# **PROGRESSIVE SOFTWARE**

# Presents Software and Hardware for your APPLE

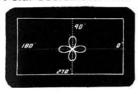
\$9.95

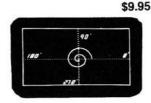
Missile—Anti—Missile (Aplsft)





**Polar Coordinator Plot** 





by TD Moteles

Tables Generator forms shape tables with ease from directional vectors-starting address, length and position of each shape. Program saves shape tables anywhere in usable memory.

Murray Summers

\$9.95

Sales Forecast provides the best forecast using the four most popular forecasting techniques: linear regression, log trend, power curve trend, and exponential smoothing.Neil D. Lipson \$9.95

Single Drive Copy is a special utility program, written by Vince Corsetti in Interger BASIC, that will copy a diskette using only one drive. It is supplied on tape and should be loaded on to a diskette. It automatically adjusts for APPLE memory size and should be used with DOS 3.2 \$19.95

Curve Fit accepts any number of data points, distributed in any fashion, and fits a curve to the set fo points using log curve fit, exponential curve fit, least squares, or a power curve fit. It will compute the best fit, or employ a specific type of fit, and display a graph of the result. By Dave Garson. \$9.95

Touch Typing Tutor teaches typing. Indicates speed and errors made. Finger Bldrs, Gen. Typing, Basic Language and User Supplied. Diskette. Written by Wm. A. Massena. \$19.95

Apple Menu Cookbook index-accessed data storage/retrieval program. Recipes stored, umlimited lines per entry. Easy editing. Formulated after N.Y. Times Cookbook. Other useful features included. Written by Wm. Merlino, M.D. \$19.95

Mailing List Program maintains complete record of name, address, phone no., mailing lables accommodated parallel card or built-in printer driver, easy data entry Diskette. 32K. \$19.95

#### POSTAGE AND HANDLING

Please add \$1.25 for the first item and \$.75 for each additional item.

- · Programs accepted for publication
- · Highest royalty paid

Utility Pack combines four versatile programs by Vince Corsetti, for any memory configuration.

- Interger to Applesoft conversion: Encounter only those syntax errors unique to Applesoft after using this program to convert any Integer BASIC source.
- Disk Append: Merge any two Interger BASIC sources into a single program on disk.
- Interger BASIC copy: Replicate an Integer BASIC program from one disk to another, as often as required, with a single keystroke.
- Applesoft Update: Modify Applesoft on the disk to eliminate the heading always produced when it is first run.
- Binary Copy: Automatically determines the length and starting address of a program while copying its binary file from one disk to another in response to a single keystroke.

**Solitare** — Old European peg game, played by one (similar to Chinese checkers). Object — to finish with last peg in center.

Charles B. Smith \$9.95

Water The Flowers — Math (add., subt., mult., div,.) grades 1-6. (disk) Judy Pegg \$9.95

Othello one or two players similar to chess in strategy. You can win with a single move. Vince Corsetti's program keep board details and flip piece. \$19.95

Blockade lets two players compete by building walls to obstruct each other. An exciting game written in Integer BASIC by Vince Corsetti. \$9.95

Saucer Invasion, Space Maze, Starwars, Rocket Pilot: Written by Bob Bishop Each \$9.95

#### Hardware

Light Pen with seven supporting routines. The light meter takes intensity readings every fraction of a second from 0 to 588. The light graph generates a display of light intensity on the screen. The light pen connects points that have been drawn on the screen, in low or high resolution, and displays their coordinates. A special utility displays any number of points on the screen, for use in menu selection or games, and selects a point when the light pen touches it. The package includes a light pen calculator and light pen TIC TAC TOE. Neil D. Lipson's programs use artificial intelligence and are not confused by outside light. The HIRES light pen, only requires 48K and ROM card

#### TO ORDER

Send Check or Money Order to:

P.O Box 273 Plymouth Meeting, PA 19462

PA residents add 6% sales tax.

U.S. and foreign dealer and distributor inquires invited All programs require 16K memory unless specified

# Applesoft II Shorthand

If you want to make Applesoft a little easier to use, try this program which permits entire commands to be input with a single control key. Since the command lookup is table driven, you can select the keys to conform to your own preferences. The techniques used provide a valuable understanding of how to add your own modifications.

Allen J. Lacy 1921 W. Oglethorpe Albany, GA 31707

This routine allows a programmer to type in an entire Applesoft command with the use of one control key.

#### Overview

The routine Shorthand ties into the input hooks at \$38 and \$39 (56 and 57 decimal) and uses a table inside the RAM version of Applesoft II. In Applesoft's table, each command is represented as an ASCII string with the high bit off except for the last character of the string which has the high bit set. The routine also uses a monitor routine to read a key. If it is a control character, shorthand gets an address from its internal table. If the high byte of the address is 0, the routine passes the control character back. If the address is not 0 shorthand passes the command stored at that location back.

Step 1 turns DOS off. Step 2 turns Shorthand on. Step 3 turns DOS back on. But DOS will not be on at the same time as shorthand.

#### To use with ROM version.

Shorthand could be adapted to run with the ROM version of Applesoft II. The addresses in Shorthand would have to be changed. I do not have access to a ROM card and so do not know the addresses. But if the ROM version is just a relocated RAM version, the addresses in Shorthand and table just need \$C800 added to them.

Shorthand does not use all of the control keys because some have special functions. These functions are shown in Table 1. If you do not mind losing these functions, these keys can be used also. The choices for which command is tied to which key is shown in the program listing. If you do not like my choices, you can change the command addresses stored in Table 2. The addresses are for the RAM version and will not work for the ROM ver-

#### **Use Of Shorthand**

Shorthand is relocatable and can be placed anywhere in memory. I normally load it at \$300-\$3AE, which is where I assembled it. But it can be placed anywhere. Applesoft's HIMEM: can be used to protect some upper memory.

Example:

A 32K system without DOS can have Shorthand loaded at \$7F51-7FFF and then HIMEM: can be set to 32593. So to bring up Shorthand use the following steps:

- 1. LOAD and RUN the Applesoft
- Enter the monitor by pressing RESET or do a CALL-151
- Type 300.3AER or type 7F51.7FFFR

- 4. Start tape with Shorthand on it and press RETURN, stop the tape when it has loaded
- Type Press Return
- Type **POKE 1144.0** Press RETURN

- 7. If Shorthand is at \$300-\$3AE type POKE 56,0; POKE 57,3 If Shorthand is at \$7F51-\$7FFF POKE 56,81: POKE 57,127
- Press RETURN
- If Shorthand is at 7F51 type HIMIM: 32593 Press RETURN

Another good place to store Shorthand is between Applesoft II and your program. The problem is that Applesoft's LOMEM: does not set the lowest memory used by Applesoft, but sets the point at which Applesoft will start storing variables. But the monitor can be used to set pointers. To do this the following steps are used:

- 1. LOAD and RUN the Applesoft II
- Enter the monitor by pressing RESET or do a CALL-151
- 3000.30AER
- Start the tape with Shorthand on it and press RETURN When it has loaded stop the tape.
- Type 67:B0 30 Press RETURN
- Type 30AF:0 30AF:0 Press RETURN
- Type 0Ġ

Simple damed Sins dor	00 LEAVE	.EQ * KNOWN POINT	LDA SW	NB YT CHECK SI	RESTORE	JSR RKEY READ A KEY	# SQR CONTROL KI	#435 CTR		******************	***	IF NOT A CONTROL JUST RETURN *	* * * * * * * * * * * * * * * * * * * *	RET PLA RESTORE KEY	LDX XSAV RESTORE	YSAV RESTORE Y		CMA ZPS RESTORE ZERO	ZPS+1			RTS	#\$7F WHICH	ASL TIMES 2	#IMB AN OF FORI		LDA (ZP),Y LOAD ENTRY	************		IF VALUE OF THE HIGH BYTE IN *	æ	AR ELSE SET	CHARACTERS	APPLESOFT'S INTERNAL TABLE *	**************		RET IF O RETU	STA POIN+I STORE IN POIN	LDA (ZP), Y	POIN	LDA #\$FF SET SW
1520 *	1540	1550 KP	1560	1570	1580	1590	1610	1620	1630 *	1640 **	1650 *	1660 *	1670 *			1720	1730	1750	1760	1770	1780	1790		1810	1830	1840	1850	1870 **	1880 *	* 0681	* 0061	1910 *	1920 *	1930 *	1940 *	1960	1970	1980	2000	2010	2020
311- 20 89 66	5 00		315- AD 78 04		5			90							AE 7A	327- AC 7B 04	48	326- AD /E U4	AD O	85 1F	89	09		339- UA		82	33E- Bl 1E										F0 E1	342- 8D /D 04 345- 88		48- 8D	34B- A9 FF
******************	APPLESOFT II SHORTHAND *		ALLEN J	7	**********		. EQ \$1E R15 OF SWEET16		** * * * * * * * * * * * * * * * * * * *	1	478-47F NOT USED	SEE MICKO #	*******	 EQ \$0478 SWITC	.EQ 504	XSAV .EQ >04/A	504	.EQ \$047E		Y .EQ SFDIB	116 .EQ \$F689 SWEET16	12	*********		START LOCATION OF SHORTHAND *	*		XSAV SAVE X	YSAV SAVE	SAVE ACC	ZP S		STA 275+1		******		SWEET16 IS USED TO STORE KP * PROGRAM COUNTER IN SIE SIF *	IS USED TO FIN	THE LOCATION OF THE TABLE IN *	SHORTHAND	**************************
1000 **	1020 *	1030 *	1040 *	* 0501		1080 *		*		1120 *	1130 *			1180 SW						1250 RKE		1280 *		1300 *		1320 *	*	S	1360	1370	1380	1390	1400	1420 *	1430 **		1450 *	*	* •	* *	*

100- 8E 7A 04 803- 8C 7B 04 806- 48 107- A5 1E 109- 8D 7E 04 80C- A5 1F

H INPUT J K CONT L LIST M NEXT O THEN P PLOT O HLIN R COLOR= S GOSUB T GOTO	U VTAB W VLIN X HTAB Z POKE		
.DA \$000 .DA \$8DE .DA \$9D0 .DA \$9D0 .DA \$9D4 .DA \$9D6 .DA \$9EF .DA \$9EF .DA \$95B .DA \$933	ZAAAAA	CT 0479 POIN 047C SW16 F689 RET 0323 NBYT 0355 LS 03AF	
2510 2520 2530 2550 2550 2550 2570 2590 2600 2610 2630	2640 2650 2660 2770 2680 2700 LS Symbol Table	SW 0478 YSAV 047B RKEY FD1B KP 0315 CTR 0337 TAB 0379  -> <	g
389-00 00 38B-DE 08 38D-00 00 391-D4 09 391-D4 09 393-D6 08 397-EF 09 399-FD 08 39B-SB 09 39B-SB 09 39B-SB 09	00 00 00 00 00 00 00 00 00 00 00 00 00	ZP 001E XSAVE 047A ZPS 047E SH 0300 RT 0324 END 0370 Control U Control H Control G Control G Control G Control C	Control D
SET CT TO 0  ***********************************	LOAD CHAR CT STORE POIN IN ZERO PAGE LOAD NEXT CHAR LAST CHAR? INCREMENT CT RETURN CHAR SAVE CHAR RESET SW	RESTORE CHAR **********  E ADDRESSES OF * APPLESOFT II *  BE CHANGED FOR *  A CALL B PEEK C D	E TEXT F FOR G
STA SW LDA #0 STA CT STA CT	0 NBYT PLA 1 LDY CT 1 LDA POIN 2 LDA POIN 2 LDA POIN 3 TA ZP 1 LDA POIN+1 3 TA ZP+1 4 CMP #\$80 6 CMP #\$80 7 CM	STA SW PLA BNE RT ************************************	. DA \$
78 04 203 00 204 79 04 205 206 207 210 211 211 212 213 215	216 9 04 217 0 04 218 0 04 220 0 04 220 0 223 0 223 0 226 0 226 0 226 0 226	8 04 2330 8 04 2330 8 04 2330 8 04 2330 8 0 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	249 249 250
111	1555 - 68 1556 - 85 1557 - 85 1557 - 85 1557 - 85 1567 - 85 1568 - 85	0 8 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	E O O

#### Press RETURN

- 8. Type NEW Press RETURN
- 9. Type POKE 1144,0 Press RETURN
- 10. Type POKE 56,0:POKE 57,48 Press Return

Shorthand will now be tied in.

Step 5 sets the pointer which tells Applesoft II where to start storing a program to \$30B0. Step 6 sets the byte just below the start point to 0, I do not know why Applesoft wants this, but it will bomb if it is not done. Step 8 causes Applesoft to reset the rest of its pointers to reflect the new start point.

Now every time you want to type one of the commands stored in the table just press the control key and another key at the same time.

Example:

To enter INPUT press the control key at the same time as the I.

I have made labels for my keyboard showing which command is under which key. To return full control to the key board, use the command IN 0. To turn Shorthand back on just POKE the correct values back into 56 and 57. Shorthand does not have to be turned off when you are finished programing and want to run a program, unless the program wants for

input one of the control keys which Shorthand uses. I normally set the hooks when I bring up Applesoft and leave them set.

The routine should work with DOS. I do not have DOS so these techniques are not tested. Since DOS communicates with the rest of the system via the input and output hooks at \$36-39, you can not set the hooks to tie in shorthand without turning off DOS. But DOS has its own internal hooks. Unfortunately the hooks are at different places for different memory sizes. In a 48K system the input hook is at \$A998, \$A999 (22120, 22119 decimal). For smaller systems subtract 48K-X from the numbers, where x is the memory size. The above information came from Exploring the APPLE II DOS by Andy Hertzfeld in MICRO 9. So POKE the address of Shorthand in the DOS hooks

Another way that should work is to turn DOS off by the use of the following steps.

- After bringing up Applesoft and loading Shorthand type PR 0:IN O Press RETURN
- Use POKEs to set 56 and 57 if Shorthand is at \$300 POKE 56,0:POKE57,3
- When you are finished type CALL 976 Press RETURN

Step 1 turns DOS off. Step 2 turns shorthand back on. DOS will not be on at the same time as Shorthand.

#### Table 2

8D0	END	8D3	FOR	8D6	NEXT	8DA	DATA
8DE	INPUT	8E3	DEL	8E6	DIM	8E9	READ
8ED	GR	8EF	TEXT	901	HLIN	905	VLIN
909	HGR2	90D	HGR	910	HCOLOR=	917	HPLOT
91C	DRAW	920	XDRAW	925	HTAB	929	HOME
92D	ROT=	931	SCALE=	937	SHLOAD	93D	TRACE
942	NOTRACE	949	NORMAL	94F	INVERSE	956	FLASH
95B	COLOR =	961	POP	964	VTAB	968	HIMEM:
96E	LOMEM:	974	ONERR	979	RESUME	97F	RECALL
985	STORE	98A	SPEED=	990	LET	993	GOTO
997	RUN	99A	IF	99C	RESTORE	9A 3	&
9A4	GOSUB	9A9	RETURN	9AF	REM	9B2	STOP
9B6	IN	9B8	WAIT	9BC	LOAD	9D0	CONT
9D 4	LIST	9D8	CLEAR	9DD	GET	9E 0	NEW
9E3	TAB (	9E7	TO	9E9	FN	9EB	SPC (
9EF	THEN	9F 3	AT	9F5	NOT	9F8	STEP
9FC	+	9FD	-	9FE	*	9 FF	/
A00	+	A01	AND	A 04	OR	A06	>
A07	=	80A	>	A09	SGN	A OC	INT
AOF	ABS	A12	USP	A15	FRE	A18	SCRN (
AlD	PDL	A20	POS	A23	SQR	A 26	RND
A 29	LOG	A 2C	E XP	A 2F	COS	A32	SIN
A35	TAN	A38	ATN	A3B	PEEK	A3F	LEN
A42	STR\$	A46	VAL	A49	ASC	A4C	CHR\$
A50	LEFT\$	A55	RIGHT\$	A5B	MID\$		

#### **Classified Ads**

TEXTPRO: Hardware, Firmware and powerful Software. Apple II plug-in addition for (1) Text editing, (2) Screen editing, (3) programming Upper/Lower case, Powerful Supermon-3 2716 EPROM Monitor I.C. Complete: 45 page documentation, text-processing software (disk) and all hardware. 15-minute installation. Call collect for information. \$349.

TextPro Corporation 1367 Post St. Studio 19 San Francisco, Ca 94109 (415) 775-5708

PET MACHINE LANG. GUIDE: Comprehensive manual to aid the machine language programmer. More than 30 routines are fully detailed so that the reader can put them to immediate use. For either Old or New ROMs. \$6.95 plus .75 postage. VISA or Mastercharge accepted. Order from:

Abacus Software P.O. Box 7211 Grand Rapids, MI 49510

APPLE Jewelry: Tie Tack, Stick Pin, or Lapel Pin. 1" diameter. Your choice \$4.95 plus 45° postage. Quantity discounts available for 10 or more pieces. Order from:

C. Armstrong P.O. Box 15614 Columbus, Ohio 43215

The Life Dynamic Transformation Experience on the Apple II with 48K. In Applesoft and Machine Language; Apple II Plus, Disk II. Unique! This program is designed for all those people who desire to experience self-transformation, life-awareness, making relation-ships work, and "getting your act together", but do NOT desire to pay est or Lifespring or any of the other "trips" of the Human Potential Movement, the \$300 or so. Includes game-playing as a means of a fun way to increase awareness. \$15.95 (disc with instructions) Order from:

Avant-Garde Creations P.O.Box 30161 Dept. MC Eugene, OR. 97403

## The Value of 16 Bits

Several years ago, the guest speaker at the local computer club, gentleman from Texas Instruments, talked about the importance of the size of a microprocessor. Using all kinds of charts, tables, and various rather logical sounding arguments, he determined that 8 bit micros did not make any sense and would never find much popularity or application! A 4 bit micro is all that is required in most process control situations, and anyone wanting to do real computer type stuff -number crunching, assembling, text processing - would much prefer a 16 bit micro. Conclusion: the 8 bit micro was doomed. Well, hundreds of thousands of 8 bit microcomputers later, it is obvious that there is a market for the 8 bit micro. Isn't 20/20 hindsight wonderful!

Actually, I did not buy this thesis at the time it was presented. I had worked on a number of projects with either minis or a precursor of the micros, and had discovered a number of instances in which an 8 bit processor was superior to its bigger brother. Does this seem strange. Let's examine the details.

One obvious type of application, in which we all participate to some degree, is any form of word processing. How many bits does it normally take to represent the normal alphanumerics and special symbols that we use in everyday writing, BASIC, assembler programming, and so forth? ASCII defines 128 characters, including a bunch of specialized control codes, and that seems to be enough for most applications. Even if you want to add special sets, such as greek for APL, the total number of unique codes required is normally going to be less than 256 decimal. Can you imagine

a keyboard to generate more than 256 characters? Since 8 bits can be used to represent 256 unique values, it is adequate for this work. In fact, it is ideal. A 16 bit machine either must ignore half of each byte, which is of course wasteful and essentially reduces it to an 8 bit machine, or must pack two 8 bit bytes into each 16 bit word. And then it must, of course, unpack the two bytes for processing, repack them again, and so forth. Therefore, the 8 bit micro is perfect for most word processing based applications. Since this single application category must account for a large percentage of the systems being purchased today, the strength of the 8 bit micro should not be surprizing.

Another application I worked on used a high speed photo scanner to digitize material for use in newspaper production - halftones and text. The scanner produced 8 bit chunks of data. The minicomputer was 16 bit based, and a lot of overhead was spent in packing and unpacking data, making records come out to an integral number of words, and other such nonsense. While the fact that 8 bits were appropriate to this particular application may have been pure serendipity, I am sure that there are numerous process control types of application which have a similar data range and which could best be served by the 8 bit micro.

Okay, how about number processing. Surely the 16 bit micro is better at performing math functions than the 8 bit micro. True, there is some advantage to a 16 bit micro if your application requires a lot of number crunching. 16 bit math operations can handle twice as much data as 8 bit ones. But, the savings may be minimal. In many numeric calculations, the

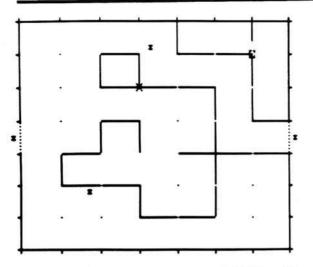
amount of code and time spent actually performing math functions may be insignificant relative to the amounts required to do all of the other programming steps required the set up, testing one bit, branching, subroutine jumps, and so forth. So, while there will probably be a time improvement with a 16 bit micro in heavy math programs, the savings may not be as great as initially imagined.

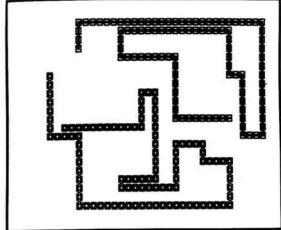
Where does the 16 bit computer excel then? I am not sure that, in general, it does. Given the generally higher cost of the micro, the higher cost and complexity of a 16 bit data bus, and so forth, the 16 bit must justify itself for a particular application. It is not a generally "better" solution. There are some features of a typical 16 bit micro that would be nice to have in the 8 bit as well. This is particularly true addressing improved capabilities. Since the address space of most 8 bit micros is actually 16 bits, it would make sense in many instances to be able to handle the full range of address space with 16 bit registers. In the 6502, a number of 16 bit addressing modes are already supported. The two main places where the 8 bit limit is restrictive are in the relative branches and in the indexed instructions. The "proposed" 6516 discussed by Randall Hyde in this issue shows how the benefits of a 16 bit micro can be combined with the strengths of the 8 bit micro to form a superior computer. It is interesting to note, however, that many of the improvements are not based on 16 bits, but are independent enhancements. My latest intelligence suggests that the initial statement in the referenced article -"Synertek is almost ready to ship the SY6516" - is a bit optimistic. But, if we all call and ask our Synertek Reps about this superior product, maybe we can get some action!

Robert M. Truffs

# Software for the Apple II







SCORE: 108 SCORE: 105

DYNAMAZE—a dazzling new real-time game. You move in a rectangular game grid, drawing or erasing walls to reflect balls into your goal (or to deflect them from your opponent's goal). Every ball in your goal is worth 100 points, but you lose a point for each unit of elapsed time and another point for each time unit you are moving. Control the speed with a game paddle: play as fast as ice hockey or as slowly and carefully as chess. Back up and replay any time you want to; it's a reversible game. By Don Stone. Integer Basic (plus machine language); 32 K; \$9.95.

ULTRA BLOCKADE— the standard against which other versions have to be compared. Enjoy Blockade's superb combination of fast action (don't be the one who crashes) and strategy (the key is accessible open space—maximize yours while minimizing your opponent's). Play against another person or the computer. New high resolution graphics lets you see how you filled in an area—or use reversibility to review a game in slow motion (or at top speed, if that's your style). This is a game that you won't soon get bored with! By Don Stone. Integer Basic (plus machine language); 32 K; \$9.95.

What is a REVERSIBLE GAME? You can stop the play at any point, back up and then do an "instant replay", analyzing your strategy. Or back up and resume the game at an earlier point, trying out a different strategy. Reversibility makes learning a challenging new game more fun. And helps you become a skilled player sooner.

WORLD OF ODYSSEY—a new adventure game utilizing the full power of Disk II, which enables the player to explore 353 rooms on 6 different levels full of dragons, dwarfs, orcs, goblins, gold and jewels. Applesoft II 48K; \$19.95 includes diskette.

PERQUACKEY—an exciting vocabulary game which pits the player against the clock. The object of the game is to form words from a group of 10 letters which the computer chooses at random. The words must be 3 to 10 characters in length with no more than 5 words of any particular length. Each player has only 3 minutes per turn. The larger the words the higher the score. Applesoft II 16K; \$9.95.

APPLESHIP—is a naval game in which two players enter their ships in respective oceans. Players take turns trying to blast their opponent's ships out of the water. The first player to destroy their opponent's ships may win the game. A great low-res graphics game. Applesoft II 32K; \$14.95.

Available at your local computer store

Call or write for our free SOFTWARE CATALOG

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

DEALER INQUIRIES INVITED

#### POWERSOFT, INC.

P. O. BOX 157 PITMAN, NEW JERSEY 08071 (609) 589-5500 Programs Available on Diskette at \$5.00 Additional

- Check or Money Order
- Include \$1.00 for shipping and handling
- C.O.D. (\$1.00 add'tl. charge)
- Master Charge and VISA orders accepted
- New Jersey residents add 5% sales tax

## The APPLE Stripper

One of the classic dilemmas in BASIC has to do with REMarks. If you use them, they take up space and time. If you do not use them, the code is hard to understand. This program resolves the problem. It permits you to generously REMark your program for documentation purposes and then remove the REMarks for the run-time version.

Bill Crouch P.O. Box 926 Long Beach, CA 90801

As a writer of custom business software for the APPLE computer, I kept running into the same conflict; good programming style insisted that I document my programs with frequent REMark statements. My customers would have a hard time understanding or changing my programs if I did not.

On the other hand, large business programs use a great deal of memory and every byte is precious. The Applesoft manual tells us that the statement: 130 THIS IS A COMMENT uses up 24 bytes of memory. In a large program, a lot of memory will be taken by REMs, leaving less for arrays and program operation. It also means more frequent waits while the machine "housecleans" its string space.

The answer is obvious; write the program with REMarks and then remove them in the final working version. If changes are needed, make them on the version with REMarks and then remove the REMs again after the bugs have been corrected.

Removing REMs by hand took too long so I wrote a simple program to do it for me. It is disk based and will work on any APPLE with a disk drive.

#### **Program Requirements**

To use this program you need only observe a couple of simple rules. First, NEVER GOTO or GOSUB to a REmark. Always GOTO or GOSUB to the first line of code after the REMark.

Secondly, for maximum benefits, put your REMarks on a separate line rather than at the end of a line of code. This program only eliminates those lines where a REM is the first thing in the line.

```
10 REM
             REM KILLER
      REM BY BILL CROUCH
      REM PO BOX 926
      REM LONG BEACH CA 90801
      PRINT CHR$ (4); "MON I,O,C"
      DIM ARRAY (1000)
70
      ONERR GOTO 240
80 X = 0
      REM
             READ TEXT FILE
        HOME : REM CLEAR SCREEN
100
        PRINT CHR$ (4); "OPEN PROG.FILE"
PRINT CHR$ (4); "READ PROG.FILE"
110
        PRINT CHR$ (4); "READ PROG.FILE"
INPUT L$: REM GET A LINE FROM DISK
IF LEFT$ (L$,5) = "63000" GOTO 250: REM CHECK FOR END OF TEXT
IF L$ = "" GOTO 130: REM ELIMINATE NULL STRINGS
LN = VAL (L$):LN = INT (LN): REM SAVE LINE NUMBER
IF LEFT$ (L$,1) = "" THEN L$ = RIGHT$ (L$, (LEN (L$) - 1)): GOTO 1
140
               LEN (L$) < 2 GOTO 130: REM IF LINE USED UP GET ANOTHER ASC (L$) < 65 THEN L$ = RIGHT$ (L$,(LEN (L$) - 1)): GOTO 170 LEFT$ (L$,3) = "REM" THEN X = X + 1:ARRAY(X) = LN: REM KEEP TRAC
180
        IF
190
         K OF REMS
                   995 GOTO 250: REM STAY WITHIN ARRAY
 210
         IF X >
         GOTO 130: REM DO IT ALL AGAIN
 220
               WRITE STRIP FILE
        IF PEEK (222) < > 5 GOTO 130: REM CHECK FOR OUT OF DATA ERROR
 240
        PRINT CHR$ (4); "CLOSE"
POKE 216,0: REM CLEAR ONERR GOTO FLAG
IF X = 0 GOTO 340: REM NO REMS IN PROGRAM
 250
 260
 270
        PRINT CHR$ (4); "OPEN STRIP.FILE"
PRINT CHR$ (4); "WRITE STRIP.FILE"
 280
 290
         FOR Y = 1 TO X
         PRINT ARRAY(Y): REM SAVE LINE # OF REM
 310
         NEXT Y
         PRINT CHR$ (4); "CLOSE"
 330
 340
         END
```

]

] PR#0

#### How to Use the Programs

There are two separate programs. The first, XFILE.MAKER, must be appended to the end of your program. You could type it in yourself or, better still, use the merge routine on the DOS 3.2 Master. The only requirement is that line 63000 be after the last line of your program. It tells the next program that it is done.

You start the process with the command "RUN 63000"

You should have both programs on their own diskette with plenty of space for their text files. If REM KILLER is not on the same diskette with XFILE.MAKER, remove line #63130.

XFILE.MAKER will convert your program into a text fie and then run REM KILLER. REM KILLER then reads the text file, makes a list of REMs and then writes them off as STRIP.FILE.

By the way, certain characters in your program will cause the computer to say EXTRA IGNORED during the running of REM KILLER. You can ignore it too.

When it is done, load your original program and EXEC STRIP.FILE. Every line which is a REMark will be removed. Then save the stripped program.

Of course also save a copy of your original program. The first program I used this on was part of a trucking company

package. It saved me over 2400 bytes.

#### How it Works

XFILE.MAKER clears the screen with line 63050 and squashes the listing to suppress extra carriage returns with line 63060.

The rest of the program writes your program to the disk as a text file. Line 63130 calls REM killer.

(Note: CHR\$(4) is the same as CTRL D and is required before every APPLE disk command.)

REM KILLER: Line 60 sets up an array in which REMs are saved. It now allows for 1000 REMs which probanly is too many. If you have memory limitations, you may reduce this number and the corresponding one on line 210.

Line 140 checks for the end of your file and is the reason line 63000 is required in XFILE.MAKER.

Lines 150-190 get rid of null lines and all non-alpha characters. Line 200 then sees if the first alpha string is REM. If so, it saves the number in the array.

Lines 240-340 save the approximate line numbers as a text file called STRIP.FILE.

When you EXEC STRIP.FILE, the line numbers are prnted just as if you had typed them yourself. And the REMark lines are eliminated.

#### XFILE.MAKER

63010	REM
	BY BILL CROUCH
63020	REM PO BOX 926
63030	REM LONG BEACH CA 90801
63040	REM APPEND TO END OF PROGRAM
63050	CALL - 936
	(프로프 - 1975) - 프랑프 프로그램 - 프라토아 (네트 - 1985) - 프로그램 (데트 - 1985) - 프로그램 (데트 - 1985) - 프로그램 (데트 - 1985) - 프로그램
63060	POKE 33,33: REM FORMAT LISTING
63070	PRINT CHR\$ (4); "MON I,O,C": REM LET US SEE IT WORK
63080	PRINT CHR\$ (4); "OPEN PROG.FILE"
63090	PRINT CHR\$ (4); "WRITE PROG.FILE"
63100	LIST 0,63000
63110	PRINT CHR\$ (4); "CLOSE"
63120	TEXT
63130	PRINT CHR\$ (4); "RUN REM KILLER"
63140	END
63150	REM

CHANGE CHR\$ (4) TO CRTL D FOR INTEGER PROGRAMS

#### Classified Ads

SYM/KIM Appendix \$4.25 postpaid, (see MICRO, 19:68 for description). First bk. of KIM: \$10. Combo Appen. & 1st bk: \$13.50. SYM-1 Hardware Theory Manual supplements, SYM-1 Ref. Manual \$6.00. Order from:

Robert A. Peck P.O. Box 2231 Sunnyvale, CA 94087

Color Graphics for OSI C1P or Superboard II. Details on modifications enable operation in several color graphic modes. Easy installation. Includes colored-video game for demo, and checkout that runs on 4K RAM systems. Low-cost hardware not included. Send \$8.65 to:

Tom Hoffman 873 Dorset Drive Knoxville, TN 37919

100 percent PET disk oriented Macro Assembler/Text Editor (MAE) Development software. Includes a new version of our ASSM/TED written specifically for the 32K new ROM PET and 2040 disk drive. Features macros, conditional and interactive assembly, and includes a relocating loader program. \$169.95 includes diskette and manual. Send \$1.00 for details.

Eastern House Software 3239 Linda Drive Winston-Salem, NC 27106

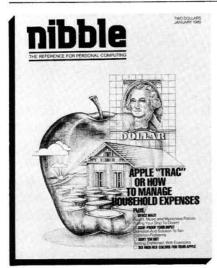
ATTENTION SYM-1 OWNERS: Modify your SYM to have 8K of 2114 RAM on board with the W7AAY piggyback RAM board. Bare board plus instructions is \$5.00 (plus a self addressed, stamped envelope per board. The new W7AAY RAE-½ board plugs into and extends ROM socket p3 so that the two chip version of RAE only needs one SYM socket. Completely assembled, with instructions for \$10.00 postpaid in the USA. Order from:

John M. Blalock P.O. Box 39356 Phoenix, AZ 85069

#### Advertise in Micro

An ad in this section may be run for \$10. per month. Should not exceed 6 lines. One ad per person (or co.) per month. Must relate to 6502 industry, be prepaid. Deadline: 25th of month.

# INTRODUCING . . . NIBBLE THE REFERENCE FOR APPLE COMPUTING



#### **NIBBLE IS:**

A SOFTWARE GUIDE for high quality Applications Programs for your Home and Business.

#### **NIBBLE IS:**

A REFERENCE GUIDE to new Programming Methods.

#### NIBBLE IS:

A BUYERS GUIDE for making purchase decisions on new products.

#### **NIBBLE IS:**

A CONSTRUCTION PROJECT COOKBOOK for adding function and value to the system you already own.

#### NIBBLE IS:

A COMMUNICATIONS CLEARING HOUSE for users, vendors, and associations.

Each issue of NIBBLE features at least one significant new application program of commercial quality. The programs in NIBBLE are surrounded with articles which show how to USE the programming methods in your OWN programs.

-	•	~					1
Examp	OC I	at 1	Inc	amir	100	ortic	ec.
LAMINI		,,,			12 6	$\mathbf{n}$	IL.J.

☐ Modeling and Forecasting Your Business
 ☐ Build a Two-Tape Controller for \$12
 ☐ Arcade Shooting Gallery — Save Your Quarters!
 ☐ Data Base Management
 System I, II, III

And many many more! NIBBLE will literally "Nibble Away" at the mysteries of your system to help you USE IT MORE. In 1980, the principal featured system is the Apple II.

#### **Try a NIBBLE**

nibb BOX 325 Li	incoln, Mass. 01773
l'II try N Enclosed	
Name	
Address_	- New York Control of the Control of
Address_ City	

© 1980 by micro-Software Publishing and Research Co.. Lincoln, Mass. 01773. All rights reserved. \*Apple II is a registered trademark of Apple Computer Company



"As manager of Personal Computers, here at Computerland of San Francisco, evaluating new software products is part of my job. With all the word processors on the market today, I -Karen Dexter Weiss choose EasyWriter for my business and personal use."

# **EasyWriter**...

#### 80 COLUMNS OF WORD PROCESSING POWER FOR YOUR APPLE II COMPUTER

Finally . . . INFORMATION UNLIMITED SOFTWARE is able to bring you a complete word processing system for the Apple II. The new EasyWriter system gives you 80 columns of upper and lower case characters for your Apple's video display, using the new SUP'R'TERM 1 board!

A long time ago, we decided to bring you the best simple-touse and understand tools for your computer system. Today we've taken another stride in that same direction. It took some doing, in both hardware and software, but we think you'll agree that for the buck, no one can touch us.



Check it out:

- · 80 Columns on the Screen!
- · Upper & Lower Case!
- · Global Search & Replace!
- Underlining!
- · Bidirectional Printing!
- · Incremental Spacing!
- · File Appending!
- . 50 Pages of Text Per Disk!

You can purchase the new EasyWriter 80 column word processing system as a complete hardware and software package directly from our new office in California, or from your local computer dealer.



Information Unlimited Software, Inc. 793 Vincente St. Berkeley, CA 94707 (415) 525-4046

- EasyWriter is a TM of Cap'n Software. Inc.
   Apple II is a TM of Apple Computers. Inc.

See The System at the 5th West Coast Computer Faire.

# Graphics and the Challenger C1P, Part 4

This continuing series on Graphics and the Challenger shows how to apply the material to create pictures and demonstrates how this may be used in Computer Aided Instruction.

William L. Taylor 246 Flora Road Leavittsburg, OH 44430

Computers are well suited for use in an educational environment, whether this is in a class room at a local high school, college, or in an industrial training seminar. The computer can aid the instructor or can be used as an individual instructor. With the introduction of the micro processor and the number of low cost personal computers that are owned and used by individuals as a hobby, the computer must be considered as a training tool for use in the home environment.

Children seem fascinated by computers and are equally fascinated by any device that has a keyboard. If the computer has any form of graphics display, either animated or still, they seem even more delighted to experiment with the device. This leads to the point that if children are drawn to the computer, then the computer, if programmed to be a teaching aid, can be a valuable tool in their education.

With this evidence I decided to try to develope a program that combines the elements that have the most attraction for children. Also, through this method, the program will at the same time be an educational tool.

The program, which I will call "Picture" was developed to be a teaching aid in the developement and spelling of English words. The program uses Graphics to draw a picture of several objects. Then the child is asked to spell the different parts of the picture that have been displayed. The child tries to spell the names of the objects displayed, and the computer displays the answer "Right" or "Wrong" on the screen in large letters.

In Part 3 of this series ("Graphics and the Challenger C1P"), we described the

features of the C1P. We developed some programs using Basic and Machine Language, in combination, to further explore the Graphics capabilities of the C1P. Many techniques were discussed and many Basic functions and statements were used in our example programs. This time let's continue with our graphics developmenmt and try a new programming approach.

This article has a two-fold purpose. First to continue our discussion of how to use the Graphics of the OSI Challenger C1P, and to secondly present a working program using the Graphics techniques

in a Computer Assisted Instruction program (CAI). The program in this part will be used as a CAI tool and will be treated as an example program. This program, by no means, is complete. That is, it can be expanded by the user. The program simply is a pure example of how to develop graphic plots: get these characters out to the monitor screen. Combining these Graphics with a program is a useful tool in the hands of the enterprising programmer. From the techniques that are presented in this example, the user will more fully understand how to develop such programs of his own.



#### **Program Description**

Let's start with a description of the "picture" program. First, what the program does is to generate a picture on the monitor screen. This picture is shown in the video memory map plotting chart in Figure 1. Notice that we have developed routines in the program that will POKE characters from the Graphics Set in the Character Generator ROM out to the monitor screen at the locations shown on the chart. In part 3 of this series, I gave a similar video memory chart. This time we will use the chart as in Figure 1. Notice that in the chart, we have drawn the picture that we wish to POKE out to the screen. All the memory locations can now easily be found, and routines written to accomplish the end task. Such a routine is located in the program between lines 10000 and 10420. This routine is used to draw the House, the Airplane, the Sun, the Man and the Car in the picture. All the parts of the picture were built from the Graphics elements in the Character Generator ROM. A list of these character elements appears in the upper left corner of Figure 1.

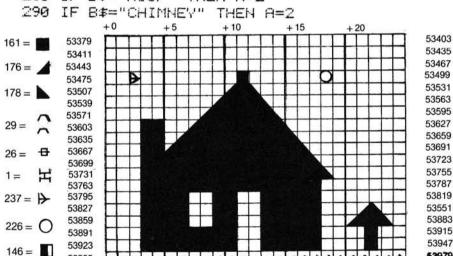
Please examine the program listing, starting at line 10000. Take the value in the statement line, For A = 53606 To 53926 Step 32. From these statement values find the corresponding value on the video memory map chart in Figure 1. It will be found that when the statement line at 10000 is compared to the memory map, you will be able to see just what the For-Next loop does. The Characters will be poke'd to these locations. Examine the program completely from line 10000 to 10420 to see how each unit works compared to the map. This example should give you a clear understanding of how to use the memory chart so that you can develop routines for your own program.

The program "Picture" contains two 176 = other Graphics Routines. These routines are used with the program to inform the user (or student) if he has identified and correctly spelled and element on the picture, or if he has identified and incorrectly spelled the object. These routines display the words: Right and Wrong, respectively. These words are in large graphic format at the top of the C1P's monitor screen. A video memory location map of these elements are in Figures 2 and 3. Please review these two figures for the memory locations. The subroutines for these graphic displays are located beginning at line 20000 for the word "Right" and at line 5000 for the word "Wrong".

These two subroutines were developed in the same manner as the one for the picture. That is, the video memory locations were plotted on the video memory plotting chart. Next, the graphics elements that were needed to generate the characters were selected from the list of graphics elements and finally, routines were written to do the task of POKEing the elements out to the screen. Analyze

OK

LIST 1 GOSUB 8000 35 PRINT"\*\*\*\* PICTURE \*\*\*\*":PRINT 50 PRINT 60 PRINT" HELLO IM A COMPUTER MY NAME IS CHALLENGER" 70 PRINT: PRINT 80 PRINT" WHAT IS YOUR NAME?" 90 INPUT A\$ 100 PRINT:PRINT"HELLO "A\$;" GLAD TO MEET YOU" 110 PRINT:PRINT" THIS IS A SPELLING GAME";A\$ 120 PRINT:PRINT" I WILL SHOW YOU A" YOU ARE TO" 130 PRINT" PICTURE. 140 PRINT"TELL ME THE PARTS" 150 PRINT" THAT YOU KNOW" 155 FOR Q=1 TO 10000:NEXT Q 160 POKE 11,232: POKE 12,15 170 X=USR(X) 180 GOSUB 10000 190 PRINT:PRINT" DID YOU SEE THE PICTURE?" 200 PRINT" TELL ME THE PARTS":PRINT 210 PRINT:PRINT" SPELL THE PARTS THAT MAKE THE PICTURE" INPUT B\$ 230 IF B\$="ROOF" THEN A=2 240 IF B\$="CHIMNEY" THEN A=2 250 IF B≰="WINDOW" THEN A=2 260 IF B\$="DOOR" THEN A=2 270 IF B\$="YARD" THEN A=2 280 IF B\$="ROOF" THEN A=2 290 IF B#="CHIMNEY" THEN A=2 +0 +5 +10 + 15 +20 53403 53379 53435 53411 53467



+10

+15

+20

+0

+5

53955

53987

54019

54051

54083

54115

54147

147 =

193 = ^

53979

54011

54043

54075

54107

54139

54171

5000 FOR A=53541 TO 53637 STEP 32 5010 POKE A,161:NEXT A 5020 FOR A=53544 TO 53640 STEP 32 5030 POKE A,161:NEXT A 5040 POKE 53638,175:POKE53639, 177:POKE 53606,176:POKE 53607,178 5050 FOR A=53546TO53642STEP32 5060 POKEA,161:NEXTA 5070 POKE 53547,151:POKE 53548,161: POKE 53579,150 5080 POKE 53580,175:POKE 53611,177: POKE 53612,178 5090 POKE 53580,175:POKE 53611,177: POKE 53612,178 5090 POKE A,161:NEXT A 5120 FOR A=53550 TO 53646 STEP 32 5110 POKE A,161:NEXT A 5120 FOR A=53552 TO 53648 STEP 32 5130 POKE A,161:NEXT A 5140 POKE A,161:NEXT A 5140 POKE A,161:NEXT A 5170 FOR A=53554 TO 53650 STEP32 5180 POKE A,161:NEXT A 5170 FOR A=53556 TO 53652 STEP 32 5180 POKE A,161:NEXT A 5190 POKE A,161:NEXT A 5190 POKE A,161:NEXT A 5190 POKE A,161:NEXT A 5200 FOR A=53590 TO 53654 STEP 32 5210 POKE A,161:NEXT A 5220 FOR A=53591 TO 53720 STEP 32 5210 POKE A,161:NEXT A 5220 FOR A=53591 TO 53719 STEP 64 5250 POKE A,161:NEXT A 5260 FOR T=1 TO 500:NEXT T 5270 X=USR(X) 5280 RETURN 8000 FOR Q=4072 TO 4095 8010 READ F:POKE Q,F 8020 NEXT Q 8030 DATA 169,32,160,8,162,0,157,0 8040 DATA 15,136,208,250,238,240 8050 DATA 15,136,208,250,238,240 8050 DATA 15,136,208,244,169,208 8060 DATA 141,240,15,96 8070 RETURN 10000 FOR A=53606 TO 53926 STEP 32 10010 POKE A,161:NEXT A	10130 FORE H.161:NEXT H 10140 FOR A=53580 TO 53584 10150 POKE A.161:NEXT A 10160 FOR A=53549 TO 53551 10170 POKE A.161:NEXT A 10180 POKE 53518,161 10190 POKE 53486,171 10191 FOR A=53704 TO 53716 10192 POKE A.161:NEXT A 10193 FOR A=53736 TO 53748 10194 POKE A.161:NEXT A 10195 FOR A=53768 TO 53780 10196 POKE A.161:NEXT A 10200 FOR A=53804 TO 53805 10205 POKEA.161:NEXT A 10207 FOR A=53804 TO 53805 10207 POKE A.161:NEXT A 10208 FOR A=53806 TO 53807 10240 POKE A.161:NEXT A 10250 FOR A=53896 TO 53901 10260 POKE A.161:NEXT A 10270 FOR A=53928 TO 53903 10280 POKE A.161:NEXT A 10270 FOR A=53937 TO 53940 10300 POKE A.161:NEXT A 10310 FOR A=53905 TO 53908 10320 POKE A.161:NEXT A 10330 POKE A.161:NEXT A 10330 POKE A.161:NEXT A 10350 FOR A=53801 TO 53806 10340 POKE A.161:NEXT A 10350 FOR A=53801 TO 53801 10400 POKE A.161:NEXT A 10370 FOR A=53800 TO 53801 10400 POKE A.161:NEXT A 10370 FOR A=53800 TO 53801 10400 POKE A.161:NEXT A 10370 FOR A=53800 TO 53801 10400 POKE A.161:NEXT A 10402 POKE53477,237 10405 POKE S3493,226 10410 FOR A=53955 TO 53979 10420 POKE A.193:NEXT A 10430 FOR D=1 TO 5000:NEXT D 10440 X=USR(X) 10450 RETURN 20000 FOR A=53509 TO 53637 STEP 32 20010 POKE A.161:NEXT A 20020 FOR A=53501 TO 53575 STEP 32
10030 POKE A,161:NEXT A	STEP 32
10040 FOR A=53517 TO 53672 STEP 31 :	20030 POKE A,161:NEXT A
10060 FOR A=53519 TO 53716 STEP 33 17	20040 POKE 53607,178:POKE53639,
10070 POKE A,178:NEXT A	7:POKE 53574,161:POKE 53510,161
10080 FOR A=53673 TO 53683	

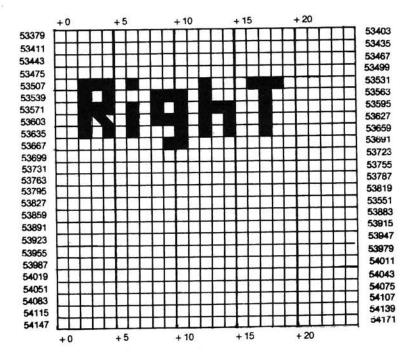
the video memory plotting charts of Figure 2 and Figure 3 along with the subroutines at lines 5000 and 20000 to see how the routines were plotted, written and used in the program.

The subroutine located between lines 8000 and 8070 is for loading the machine code routine into user memory. This routine is for the Fast Screen Erase routine used by the program to clear the screen whenever called. This subroutine will be called at line 1 in the Main Line Basic program. The routine for the fast screen erase has been included in the previous parts of this series. The reader should review these parts for a complete description fo this routine.

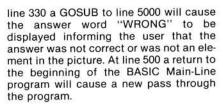
Now that I have described the Graphics generating routines and how they were developed, let's continue with the Mainline BASIC program that uses the subroutines. The program from line 35 through 500 forms the BASIC CAI user program. This program is a demonstration of he to develope programs in which the use can be taught such things as spellin 1 which is the purpose of this program, ombined with the graphics pres lation.

The For-Next loop at line 155 is used to give the user time to read the screen text just displayed. Statement line 160 sets the USR Vector to point to the Fast Screen Erase Machine Language routine located at OFE8 hex or 8168 decimal. Line 170 causes the program to jump to the machine language routine at OFE8 ot 8168 decimal where the screen will be erased.

Input from the user is accepted at state ment line 220. This data is stored in a



string variable, labeled B\$. The input string (B\$) is then compared in the string looking up table for a string match. If a match between the input and a table content is found, the information is then passed to the variable A as a decimal value. This value is then compared at Line 330 and Line 335 to check for a correct answer from the user. If a correct match was found in the string table, the A variable will force a GOSUB to Line 20000 where the answer word "Right" will be displayed for the user's answer. If a match did not occur in the string table, at



This program, as stated before, is not really complete, but an example to show how such a program can be constructed. This program can be expanded or modified by the reader. If you should desire to expand the picture display to include more ojects, then these object names should be included in the string table.

The complete program including the Graphics routines and the fast screen erase routine is located at the top of the first 4K of user memory. If you have more memory and wish to expand the program, you will have to relocate this routine. Listing 2, shows the modifications to the program which will allow the routine to be relocated to begin at 1FE8 Hex or 8168 decimal. The user must set memory size to 4050 decimal for a system with 4K of memory, and 8167 for a 8 K system.

memory, and 8167 for a 8 K system. 160 POKE 11,232 : POKE 12, 31 8000 FOR Q = 8168 TO 8191 8050 DATA 31,136,208,244,169,208 5410/ 8060 DATA 141,240,31,96 

+20

+10

+15

+0

+5

20045 POKE 53575,175:POKE 53606,162: POKE 53574,154 20050 FOR A=53513 TO 53641 STEP 32 20060 POKE A,161:NEXT A 20065 POKE 53545,32 20070 FOR A=53579 TO 53643 STEP 32 20080 POKE A, 161: NEXT A 20090 FOR A=53581 TO 53709 STEP 32 20100 POKE A, 161: NEXT A 20110 POKE 53580,161:POKE 53644,161: POKE 53708,161 20140 FOR A=53519 TO 53647 STEP 32 20150 POKE A,161:NEXT A 20160 FOR A=53585 TO 53649 STEP 32 20170 POKE A,161:NEXT A 20180 POKE 53584,161 20190 FOR A=53524 TO 53652 STEP 32 20200 POKEA, 161: NEXT A 20210 POKE 53523,161:POKE 53525,161 20230 FOR E=1 TO 500: NEXT E 20240 X=USR(X) 20250 RETURN 0K

#### OHIO SCIENTIFIC

HARDWARE: C1P VIDEO—Gives true 32 Chars/Line with guard bands. This is not a make-shift mod. It makes your video every bit as good as the 4P's plus you have switch selectable 1, 2 and 3 MHz. CPU clock as well as 300, 600 and 1200 baud for cassette and serial port all crystal controlled.

Complete Plans — \$18.95 or Send in your C1P to:

Kit — \$39.95

Personal & Business Computer Connection 38437 Grand River Farmington Hills, MI. 48018

and we will install the video mod. for \$79.95. Other mods available...Add sound: RS-232 port cassette motor control

**SOFTWARE:** (With Documentation) For C1, C2, 4P & 8P. Chess 1.9, Backgammon, Excellent card games, Arcade type games, Utility Programs, Mini Word Processor, Memory Maps, etc.

Catalog With Free Program (hard copy) and Memory Map for BASIC in ROM Models...\$1.00

Progressive Computing 3336 Avondale Court Windsor, Ontario, Canada N9E 1X6 (519) 969-2500

NOTE: Any Modification will VOID your OSI warranty.

#### WHAT'S THE ONE THING NO ONE HAS THOUGHT ABOUT DOING WITH COMPUTERS?

*&*&&&&&&&&&

We acknowledge that computers are the most valuable data processing devices ever conceived for business and education, and are the most creative toys on earth. However, the potential of computers has only begun to be explored. Avant-Garde Creations has discovered and developed a way to use computers in the areas of self-transformative experiences, life-awareness, making relationships work, and "getting your act together".

Previously, it was thought that such trips as est, Lifespring, Actualizations, and others were the only means of significantly dealing with the above areas. We acknowledge that they are indeed valuable experiences. But because one has to devote many full days and hundreds of dollars to such trips, all those people who aren't yet ready to get into all this that deeply are left with nowhere to turn for such awareness experiences.

The intention of Avant-Garde Creations is to change that. We have the knowledge and techniques, and now we have the programs. The first is called "The Life Dynamic Transformation Experience", and the second is "The Relationship Life Dynamic". Both are available on disk at this time. Over the next year 9 more life dynamic areas programs will be developed in the following areas: physical, creativity, environment, meaning, conditioning, sexuality, normalcy, aliveness, and responsibility.

Why spend hundreds of dollars to find out if you're into working on such areas of your life? Why not get something you can

use over and over, at your convenience?

One of the most exciting aspects of our programs is that in many program sections, it was discovered that "game-playing" was the most effective method of getting people to perceive a difficult truth. So from time to time you're given 100% original games to play, games specifically designed to give you the opportunity to "get" something challenging or elusive. So even if you don't "get" something, you'll have a great time!

Apple II with Applesoft and 48K required with single disk drive.

The Life Dynamic Transformation Experience. \$15.95 postpaid The Relationship Life Dynamic. \$15.95 postpaid Complete Program Listings available:

The Life Dynamic Transformation Experience...\$4.95 postpaid The Relationship Life Dynamic......\$4.95 postpaid



If you don't have an Apple II with Applesoft, there are approximately 5 Apple stores in every state. We'll send you the address of the one nearest you. Your Apple dealer will be glad to give or rent you (very cheaply) some time on one of their computers, and will be happy to get you started running the program. In case you think you need to understand computers to run our programs, we assure you that if you can read English and type your name, you won't have any problem.

#### AVANT-GARDE CREATIONS

PO Box 30161 Eugene, OR 97403 Dept. C1

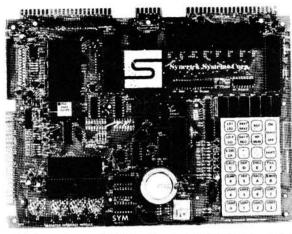


#### SYM-1, 6502-BASED MICROCOMPUTER

- . FULLY-ASSEMBLED AND COMPLETELY INTEGRATED SYSTEM that's ready-to-use
- ALL LSI IC'S ARE IN SOCKETS
- 28 DOUBLE-FUNCTION KEYPAD INCLUDING UP TO 24 "SPECIAL" **FUNCTIONS**
- EASY-TO-VIEW 6-DIGIT HEX LED DISPLAY
- KIM-1\* HARDWARE COMPATIBILITY

The powerful 6502 8-Bit MICROPROCESSOR whose advanced architectural features have made it one of the largest selling "micros' on the market today.

- THREE ON-BOARD PROGRAMMABLE INTERVAL TIMERS available to the user, expandable to five on-board.
- 4K BYTE ROM RESIDENT MONITOR and Operating Programs.
- Single 5 Volt power supply is all that is required.
- 1K BYTES OF 2114 STATIC RAM onboard with sockets provided for immediate expansion to 4K bytes onboard, with total memory expansion to 65, 536 bytes.
- USER PROM/ROM: The system is equipped with 3 PROM/ROM expansion sockets for 2316/2332 ROMs or 2716 EPROMs
- ENHANCED SOFTWARE with simplified user interface
- STANDARD INTERFACES INCLUDE:
  - -Audio Cassette Recorder Interface with Remote Control (Two modes: 135 Baud KIM-1\* compatible, Hi-Speed 1500 Baud)
  - -Full duplex 20mA Teletype Interface
  - —System Expansion Bus Interface
  - -TV Controller Board Interface
  - -CRT Compatible Interface (RS-232)
- APPLICATION PORT: 15 Bi-directional TTL Lines for user applications with expansion capability for added lines
- EXPANSION PORT FOR ADD-ON MODULES (51 I/O Lines included in the basic system)
- SEPARATE POWER SUPPLY connector for easy disconnect of the d-c
- AUDIBLE RESPONSE KEYPAD



Synertek has enhanced KIM-1\* software as well as the hardware. The software has simplified the user interface. The basic SYM-1 system is programmed in machine language. Monitor status is easily accessible, and the monitor gives the keypad user the same full functional capability of the TTY user. The SYM-1 has everything the KIM-1\* has to offer, plus so much more that we cannot begin to tell you here. So, if you want to know more, the SYM-1 User Manual is available, separately

SYM-1 Complete w/manuals \$229.00 7.00 SYM-1 User Manual Only 60.00 SYM-1 Expansion Kit

Expansion includes 3K of 2114 RAM chips and 1-6522 I/O chip.

SYM-1 Manuals: The well organized documentation package is complete and easy-to-understand.

SYM-1 CAN GROW AS YOU GROW. Its the system to BUILD-ON. Expansion features that are soon to be offered:

\*BAS-1 8K Basic ROM (Microsoft)

\$89.00

\*KTM-2 TV Interface Board

319.00

\*We do honor Synertek discount coupons

#### QUALITY EXPANSION BOARDS DESIGNED SPECIFICALLY FOR KIM-1, SYM-1 & AIM 65

These boards are set up for use with a regulated power supply such as the one below, but, provisions have been made so that you can add onboard regulators for use with an unregulated power supply. But, because of unreliability, we do not recommend the use of onboard regulators. All I.C.'s are socketed for ease of maintenance. All boards carry full 90-day warranty.

All products that we manufacture are designed to meet or exceed industrial standards. All components are first quality and meet full manufacturer's specifications. All this and an extended burn-in is done to reduce the normal percentage of field failures by up to 75%. To you, this means the chance of inconvenience and lost time due to a failure is very rare; but, if it should happen, we guarantee a turn-around time of less than forty-eight hours for repair.

Our money back guarantee: If, for any reason you wish to return any board that you have purchased directly from us within ten (10) days after receipt, complete, in original condition, and in original shipping carton; we will give you a complete credit or refund less a \$10.00 restocking charge per board.

#### **VAK-1 8-SLOT MOTHERBOARD**

This motherboard uses the KIM-4\* bus structure. It provides eight (8) expansion board sockets with rigid card cage. Separate jacks for audio cassette, TTY and power supply are provided. Fully buffered bus. \$139.00

VAK-1 Motherboard

#### VAK-2/4 16K STATIC RAM BOARD

This board using 2114 RAMs is configured in two (2) separately addressable 8K blocks with individual write-protect switches

\$239.00 VAK-2 16K RAM Board with only 8K of RAM ( ½ populated) \$125.00 VAK-3 Complete set of chips to expand above board to 16K

#### VAK-4 Fully populated 16K RAM VAK-5 2708 EPROM PROGRAMMER

This board requires a +5 VDC and +12 VDC, but has a DC to DC

multiplyer so there is no need for an additional power supply. All software is resident in on-board ROM, and has a zero-insertion socket.

VAK-5 2708 EPROM Programmer

\$249.00

#### **VAK-6 EPROM BOARD**

This board will hold 8K of 2708 or 2758, or 16K of 2716 or 2516 EPROMs. EPROMs not included.

VAK-6 EPROM Board

\$119.00

VAK-7 COMPLETE FLOPPY-DISK SYSTEM (See March Issue of Micro)

#### **VAK-8 PROTYPING BOARD**

This board allows you to create your own interfaces to plug into the motherboard. Etched circuitry is provided for regulators, address and data bus drivers; with a large area for either wire-wrapped or soldered IC circuitry

**VAK-8 Protyping Board** 

\$39.00

#### **POWER SUPPLIES**

ALL POWER SUPPLIES are totally enclosed with grounded enclosures for safety, AC power cord, and carry a full 2-year warranty.

#### **FULL SYSTEM POWER SUPPLY**

This power supply will handle a microcomputer and up to 65K of our VAK-4 RAM. ADDITIONAL FEATURES ARE: Over voltage Protection on 5 volts, fused, AC on/off switch. Equivalent to units selling for \$225.00 or more. Provides +5 VDC @ 10 Amps & ±12 VDC @ 1 Amp \$119.00

**VAK-EPS Power Supply** 

\$325.00

KIM-1\* Custom P.S. provides 5 VDC @ 1.2 Amps and +12 VDC @ .1 Amps KCP-1 Power Supply

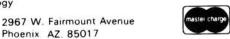
SYM-1 Custom P.S. provides 5 VDC @ 1.4 Amps

(602)265-7564

\$39.00 \$39.00

VCP-1 Power Supply

\*KIM is a product of MOS Technology







## SYMple BASIC Data Files

The SYM-1 has a Microsoft BASIC available in ROM. Data Save and Data Load via the cassette are NOT supported by this version. The routines required to implement these two important functions are presented here.

John M. Blalock 3054 West Evans Drive Phoenix, AZ 85023

If you've read "A SYMple Memory Expansion" in the August 1979 issue of MICRO and "Another KIM Expansion" in the September 1979 issue of *Kilobaud Microcomputing*, then you know that I like Micro-Z's BASIC for the KIM. You will also know that I have the Synertek BAS-1 BASIC for the SYM. Both versions were written by Microsoft, have 9-digit decimal accuracy, etc. but differ in some of their functions.

#### Comparing the Micro-Z Synertek BASICS

Synertek BASIC has a more convenient USR function and a &"hex" function that are definite improvements over the original BASIC. Their ROM version has no GET function like Micro-Z's. Another difference is that a response of a carriage return only to an INPUT statement will cause a break in program execution with Synertek's BASIC. Micro-Z has supplied a patch to defeat this break. The Synertek ROM does not include any trig functions, but they have recently released Technical Note #53-SSC that gives you full trig capability using only 313 bytes of RAM.

The main difference between the two BASICs, then, is the data save/data load feature added to his version by Bob Kurtz of Micro-Z. This is a very valuable feature that Microsoft left out. BASIC can not be used to maintain any types of files such as mailing lists, inventory records, or financial records without this feature. Perhpaps you could enter the data via DATA statements, but that would be a very trying task indeed! This feature is the major reason that I have preferred Micro-Z's BASIC over Synertek's.

#### Data Save/Data Load for Synertek BASIC

Listings 1, 2, and 3 are my first attempts to provide the same data save/data load functionality for the SYM with BAS-1. Listing 1 is just BASIC initialization, program loading, and a LIST of the program. All terminal input has been underlined for clarity. The little crooked arrows represent a carriage return typed in.

Listing 2 is a RUN of the program showing the means used to save the data. Three separate records are saved; the page zero pointers, the numeric data and string pointers, and the string data itself. To reload this data, BASIC must be initialized with the same memory size and the program can not have been modified.

Listing 3 is another RUN of the program after memory was cleared and the program reloaded. The data saved in listing 2 was restored, as can be seen. No, it is not as convenient as Bob Kurtz's method, but it works! Bob packs all the data together with a machine language subroutine and save it as one record. Another subroutine loads the combined record and then unpacks it, moving the data back to its original locations.

#### Machine Language Version of Data Save/Data Load

Listing 4 is a machine language subroutine that will save and load BASIC data files without having to turn control over to the SYM monitor. The data is still saved in three separate records, but they are recorded/loaded one right after another by the routine. An extra few seconds for each save or load (for sync, etc.) shouldn't hurt anyone, should it?

Listing 5 is a VERIFY dump of the subroutine. Load it in, VERIFY between the same addresses, and if you check sums match mine then you keyed it in correctly. Now we know why Synertek put those check sums on the VERIFY dumps! The rest of listing 5 shows BASIC initialization and the loading of the revised BASIC program.

Listing 6 is just a LIST of the revised program. Note the memory size was specified to allow room for the machine language subroutine which is called by statements 100 and 400. With either of the two methods, put the call to the load routine after any DIM statements and before the main program body. The call to the save routine should be at the very end of the program, as shown. Any changes to the program that increases the memory size needed for it will prevent data saved by a prior version from being loaded correctly.

Listing 7 is a RUN of the revised program wherein the data that is entered is saved at the end of the RUN. Listing 8 shows memory being cleared, BASIC initialization identical to that used in listing 7, and then the BASIC program being reloaded. The RUN of the program loads the data saved in listing 7.

If you plan on saving and loading data files very often, dedicating 148 bytes of memory to this subroutine should pay for itself in convenience over the method given earlier.

#### SYMple Memory Expansion Update

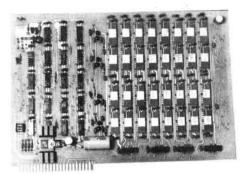
Regular readers of MICRO will recognize from the listings that my SYMple memory expansion board is still work-

INC ID ; third record  JGR DUHFT ; save it  JGR DUHFT ; save it  JGR DUHFT ; save it  JGR SAVER ; save all registers  JGR SAVER ; saver registers	Listing 5  .V IF&C-1FFF  1F&C 20 88 81 8D 4E A6 A9 00,53 1F74 8D 4D A6 8D 4B A6 A9 &5,5F  1F72 8D 4C A6 A9 EA B A6 A9 &5,5F  1F72 8D 4C A6 A9 EA B A6 A9 &65,5F  1F84 20 87 8E 20 FA 1F A6 A5,08  1F94 4D A6 A5 81 8D 4A A6 A5,08  1F94 4D A6 A5 81 8D 4A A6 A5,08  1F94 A7 8E 20 FA 1F A0 80 A5,1D  1FA4 87 8E 20 FA 1F A0 80 A5,1D  1FA5 8D 46 A5 8D 4A A6 A5 8B,9E  1FB4 A6 A5 B7 8D 4A A6 A5 8B,9E  1FB4 B7 A6 A5 B7 B9 A6 A5 B9,7A  1FC4 BF AC CA 81 20 88 81 8D,7A  1FC4 BF AC CA 81 20 88 81 8D,7A  1FC6 AF A6 A5 B7 8C C9  1FA A6 A5 B7 8C C9 A7,02  1FA A6 A5 B7 B7 B7,02  1FA A6 A5 B7 B7 B7,03  1FA A6 A5 B7 B7 B7,03  1FA A7 B7 B7 A7 B7,03  1FA A7 B7 A7 B7 A7 B7,03  1FA A7 B7 A7 B7 A7 B7,03  1FC 20 A7 B7 A7 B7,04  4CB5
1FBC:8D 48 A6 1FC2:20 87 8E 1FC2:20 87 8E 1FC5:4C C4 81 1FC8:80 4E A6 1FC8:80 4E A6 1FC8:80 4E A6 1FC8:80 4E A6 1FC8:80 6A 1F 1FD0:A5 D4 1FD0:A5 D4 1FD0:A5 D4 1FD0:A5 D4 1FD0:A5 D4 1FD0:A5 D4 1FC8:B5 D5 1FE7:A5 B6 1FF7:A5 B7 1FF7:A	
### SAUER ***    Foutine to save and load SYM BASIC data tables.   Foutine to save and load SYM BASIC data tables.   Initialize BASIC with MEMORY SIZE? = 8043 for an 8K SYM.   Call data load routine with 'R = USR(81636.384)* after   Call data save routine with 'R = USR(8044.384)* after   Call data save routine with 'R = USR(8044.384)* after   All data save routine with 'R = USR(8044.384)* after   Sall data save routine with 'R = USR(8044.384)* after   Sall data save routine with 'R = USR(8044.384)* after   Sall data save routine   Sall data save routine with 'R = USR(8044.384)* after   Sall data save routine with 'R = USR(8044.384)* after   Sall data save routine and load sall data     Sall data save routine     S	39; ore \$1F6C ; 8044 decimal 41 SAVE: JSR SAVER ; save all resisters 42
	1F6C:20 88 81 1F6F:8D 4E 46 1F72:8D 4D 46 1F77:8D 4B 46 1F77:8D 4B 46 1F77:8D 4C 46 1F77:8D 4C 46 1F77:4D 6A 46 1F87:4D 6A 46 1F87:4D 6A 46 1F87:4D 70 1F88:4D 4C 46 1F98:4D 4C 46 1F98:4D 4C 46 1F98:4D 4D 46 1F89:4D 4D 4

•	0
	_
	9
ì	=
ì	₽
ì	S
,	_

	DO YOU WANT TO SAVE THIS DATA? YES START RECORDER ON RECORD, THEN ENTER 'G CR'. G ** DATA SAVED. PROGRAM ENDS.	F 00-00-FO FF 00-200-1FFF L.2 01 J 0 MEMORY SIZE? 8043 WIDTH?	BASIC V1.1 COPYRIGHT 1978 SYNERTEK SYSTEMS CORP.  OK LOADT LOADED OK RIJN DO YOU WANT TO RESTORE PRIOR SAVE/LOAD DEMO DO YOU WANT TO RESTORE PRIOR SAVED DATA? YES START RECORDER ON PLAYBACK.  **  DATA LOADED.  ENTER PAY NUMBER AND NAME, SEPARATED BY A COMMA: FOR EXAMPLE; '12345, JOHN SMITH'	? 44444,JAMES BOND ? 44444,ALLEN SMITH ? -1, THE TABLE NOW CONTAINS THE FOLLOWING DATA: * PAY NUMBER NAME 1 23456 JOHN SMITH 2 34567 MARY JOHNSON 4 44444 ALLEN SMITH DO YOU WANT TO SAVE THIS DATA? NO PROGRAM ENDS.
LIST LIST	WWOAA HHA	000000000	260 INPUT A(1), 58.(1) 270 IF A(1) 270 IF A(1) 280 NEXT I;PRINT"TABLE IS FULL!" 290 N = I 300 PRINT"THE TABLE NOW CONTAINS THE FOLLOWING DATA:" 310 PRINT" # PAY NUMBER NAME" 320 FOR I = 1 TO N 330 PRINT;PRINT;PRINT 340 NEXT I 350 PRINT;PRINT;PRINT 350 INPUT"DO YOU WANT TO SAVE THIS DATA? ";Q\$ 350 INPUT"DO YOU WANT TO SAVE THEN PRINT;PRINT"PROGRAM ENDS.":END 350 INPUT"START RECORDER ON RECORD, THEN ENTER 'G CR', ";Q\$ 400 Q = USR(8044,384) 410 PRINT;PRINT"DATA SAVED. ";PRINT;PRINT"PROGRAM ENDS.":END OK	FUN  DATA SAVE/LOAD DEMO  DO YOU WANT TO RESTORE PRIOR SAVED DATA? NO  ENTER PAY NUMBER AND NAME, SEPARATED BY A COMMA: FOR EXAMPLE; 12345, JOHN SMITH ? 12345. JOHN SMITH ? 23456, JANE JONES ? 34567, MARY JOHNSON ? -1.  THE TABLE NOW CONTAINS THE FOLLOWING DATA:  THE TABLE NOW CONTAINS THE FOLLOWING DATA:  12345 JOHNSON 3 34567 MARY JOHNSON 3 34567 MARY JOHNSON 3

#### **16K MEMORY**



#### K-1016

- ADDRESSED AS CONTIGUOUS 16K STARTING AT ANY 8K BOUNDARY
- LOW POWER 1.6 WATTS TOTAL
- K-1016A \$340 6 MONTH WARRANTY

#### SYSTEM EXPANSION



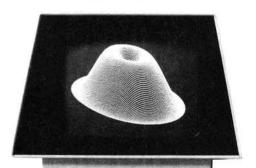
K-1012

- 12 PROM SOCKETS 2708/TMS 2716, USES THE POWER OF ONLY 1 PROM.
- 32 BIDIRECTIONAL I/O LINES
- FULL RS-232 ASYNC SERIAL COMMUNICATIONS, 75-4800 BAUD
- PROM PROGRAMMER
- K-1012A \$295

# EXPANSION

**FOR YOUR 6502 COMPUTER** 

# HIGH RESOLUTION GRAPHICS



- 320 x 200 BIT MAPPED GRAPHICS
- 8K RAM AVAILABLE FOR USE
- EACH POINT INDIVIDUALLY ADDRESSABLE
- K-1008A \$240, PET \$243 (PLUS PET INTERFACE)

#### MULTI-HARMONIC 4 VOICE MUSIC



K-1002-2



K-1002

MODEL FOR ALL PETS MODEL FOR KIM, AIM, SYM

- FORIER SYNTHESIZED WAVEFORMS
   UP TO 16 HARMONICS
- 4 VOICES PLAY SIMULTANEOUSLY
- QUALITY D/A CONVERTER, 6 POLE FILTER AND AMPLIFIER
- HARDWARE \$40-50, SOFTWARE — \$20



ALL MTU PRODUCTS ARE SUPPLIED WITH FULL DOCUMENTA-TION CLASSED AS "BEST IN THE INDUSTRY". MANUALS MAY BE PURCHASED SEPARATELY. Micro Technology Unlimited

P.O. Box 4596, 841 Galaxy Way Manchester, N.H. 03108

603-627-1464

Call Or Write For Our Full Line Catalog

# A Perpetual Calendar Printer for the AIM

If you know the proper tricks, a Perpetual Calander is quite easy to program. Here it is presented for the AIM 65. In addition to being an interesting demonstration, it points out a few programming tricks required when using integer numbers in BASIC.

Mel Evans 1027 Redeemer Ann Arbor, MI 48103

Another calendar printer? Yes, but with a couple of new twists. First, it puts out to the AIM printer. So the next time someone asks, "Okay, but what can it actually do?," you can give him an answer he can put in his pocket and take home with him.

Second, it has a built-in perpetualcalander algorithm that finds the starting day-of-the-week for any month of any year from 1583 AD (the start of the Gregorian calendar) to 999999999 AD (or until we change the calendar, or until the world ends, whichever comes first.) The algorithm is fairly simple, but the results can be impressive. For example:

RUN HOW MANY MONTHS? 1 MONTH #? 7 YEAR? 1776

***	k option	JUL	_₩	177	76 ×	***	
S	M	T	14	T	F	5	
	1	2	3	4	5	6	
7	8	9	10	4 11	12	13	
14	15	16	17	18	19	20	
21	22	23	24	25	26	27	
	29						

"So, Independence Day happened on a Thursday."

"You mean it figured out all those leap years clear back to 1776?"

"Well, the equivalent of that, yes."

"How do I know it's right?"

"You don't."

"Okay, print me December, 1941. I know what day Pearl Harbor happened on."

```
RUN
HOW MANY MONTHS? 1
MONTH #? 12
YEAR? 1941
```

4	いこし			72.	1 1 1	. 4. 4.
5	M	Т	М	Т	F	5
	1	2	3	4	5	6
7	8	9	10	4 11	12	13
14	15	16	17	18	19	20
21	22 29	23	24	25	26	27
28	29	30	31			

"So December 7th was a Sunday."
"Hey, that's right! Okay, print me the start of year 2000."

```
RUN
HOW MANY MONTHS?
FIRST MONTH #?
YEAR? 2000
*** JANUARY
               2000
 5
                T
                    F
            И
                6
                       15
 9
           12
               13
    10
       11
                   14
                   21
                       22
           19
               20
16
    17
       18
   24
       25
           26
30
    31
**
   FEBRUARY
               2000 ***
            14
                T
                    F
                        5
 5
                3
                    4
                      12
         8
            9
               10 11
 6
           15
               17
                   18
                       19
       15
13
   14
    24
       22
               24
20
    28
        29
```

"How about that! It got February right. Century years aren't normally leap years, but every fourth century is, and there it is."

"Right. Want a calendar of this month, and may; be the rest of the year?"

"Sure, but make it through next February. Why do all calendars end at December?"

"I don't know, but this one won't."
RUN
HOW MANY MONTHS? 5
FIRST MONTH #? 10
YEAR? 1979

```
1979
*** OCTOBER
                       ***
                         5
             M
     1
         2
     8
         9
                11
            10
       16
14
                18
                    19
                        20
                    26
21
        23
            24
                25
28
    29
        30
                1979
    NOVEMBER
                      * * *
                         5
                     F
 Ξ
     M
         T
                 T
             M
                          3
                 1
                 8
                     9
                        18
                    16
                15
    12
        13
            14
                        17
11
        20
            21
                    23
18
    19
                22
            28
                29
                    30
                1979
    DECEMBER
                          5
             М
                          1
```

*** 5 6 13 20 27	JF	MUF	184	198	)Ø ×	* * *
5	M	Ţ	Ы	T	F	5
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
29	21	22	27	24	25	26
27	29	29	30	31		-
1000000						
**	FEE	RUA	HR"	198 T 7 14 21	30 :	÷ :4: :4:
5	M	T	44	T	F	5
					1	2
3	4	5 12	. 5	7	8	9
1.0	11	12	13	14	15	18
17	18	.+ 0	20	21	22	23
17 24	25	26	27	$\bar{28}$	29	

The day-of-the-week algorithm appeared in BYTE (Day of Week and Elapsed Time Programs," W. B. Agocs, BYTE, September, 1979, p. 126.). I read it, thought "That's neat," and forgot it. Then a calendar printing program for Teletype came out in Kilobaud ("Calendar Program," Steve Tabler, Kilobaud Microcomputing, October 1979, p. 102.). Can the AIM do that on its printer? Sure it can! Can I build in that day-of-week algorithm so that it doesn't need starting instructions? Sure I can! The resulting AIM BASIC program is listed in Figure 1.

The starting day-of-week algorithm is in lines 85 through 150. It uses "Zeller's congruence," as explained in Agoc's article. Zeller first does some juggling of month and year numbers before getting down to the main computation of the day-of-week (variable DW in line 150).

The algorithm packs more power than I needed here; it works for any year, month, and day-of-month (day-of-month is variable DM in line 130). Since I only needed the beginning day-of-week of each month to be printed, I set DM = 1 in line 129. To restore the algorithm to its full power, just delete that one statement, and use DM as an input.

AIM BASIC (like most BASICs) does not allow much format flexibility in printing numbers, so to squeeze those date-lines onto the 20-column printer, a string variable, L\$, is used to build each line before printing. L\$ is first nulled (e.g., line 290), and is then built up, character by character, as in line 350:

L\$ = L\$ + CHR\$(48 + D2)

This statement adds D2, the second (units) digit of a two-digit date number, to line L\$. As shown in Appendix E of the

AIM BASIC manual, CHR\$(48) is ASCII "0" (zero), and the other digits follow. So, if D2 = 5, say, ASCII "5" is added to the string. After the last character has been added, the line is printed (e.g., line 380).

If you are fussy about format, the above technique gives you total control over each column of each line. If numbers don't print to suit you; don't print numbers, print characters.

AIM BASIC has one quirk which I haven't noticed in others (but if you're running a different BASIC, you might like to check it out). If X evaluates internally as less than an integer, but is sufficiently close to that integer, it will print as the integer, but INT(X) will truncate down to the next-lower integer; e.g., if X = 4.99999...,you get:

PRINT X 5 PRINT INT (X) 4

Don't believe it? Try this:

RUN X= 5 INT(X)= 4

To prevent this from happening, add a dab to X before doing INT(X). How much is a dab? Anything less than the smallest meaningful increment in X. The first equation in line 258, for example, is computing the century from the year:

C = INT(Y/100 + .005)

If year Y increases by 1, Y/100 increases by .01, so the added dab is half that. This assures that it will work for the year 2000, and is small enough so it will also work for 1999.

Another example is on Line 262: INT(YC/4 + .1).

When YC increments by one, YC/4 increases by .25, and the added dab is less than half that. The previous .005 would work fine here, too, but .1 costs fewer bytes.

A final note of minor interest. Line 80 sends two line-feeds to the printer before starting the calendar, and line 430 sends it five line-feeds, so you can tear off the finished calendar without having to pump th "LF" key. And PRINT TAB (100) is sure neater than a string of five PRINT statements, isn't it?

```
LIST
 4 REM
         PERPETUAL-
 5
  REM
    CALENDAR PRINTER
 6
  FEM
 10 DIM A(12), R$(12)
 29 FOR I=1 TO 12:RE
AD A/ID:NE//T I
 30 FOR I=1 TO 12:RE
AD R$(I):NEXT I
 40 INPUT "HOW MANY
MONTHS" N
 50 IF N=1 THEN INPU
T "MONTH #"; M
 60 IF NO1 THEN INPU
T "FIRST MONTH #"; M
 70 INPUT "YEAR"; Y
 89 PRINT TAB(40)
         CONVERT TO
 85 REM
 ZELLER MONTH & YEAR
 90 MZ=M-2:YZ=Y
 100 IF M=1 THEN MZ=
11:YZ=Y-1
 110 IF M=2 THEN MZ=
12:47=4-1
 115 REM
            FIND
STARTING DAY-OF-WEEK
 120 CZ=INT(YZ/100+.
005):YZ=YZ-100*CZ:DM
 130 D1=INT(2.6*MZ-.
1)+DM+YZ
 140 D1=D1+INT(YZ/4+
 1)+INT(CZ/4+.1)-2*C
 150 DW=D1-7*INT(D1/
7+.01)+1
 155 REM PRINT HEADE
 160 PRINT R$(M);:PR
INT Y; :PRINT"***"
 170 PRINT" 5
       F
 1.1
 175 REM BUILD FIRST
   DATE-LINE & PRINT
 180 L#="":D1=DW-.5
 190 FOR I=1 TO 7
 200 DT=I-DW+1
 210 IF IKD1 THEN L#
=[$+"
 220 IF IDD1 THEN L$
=L$+" "+CHR$(48+DT)
 230 IF IC6.5 THEN L
$=[$+" "
```

240 NEXT I 250 PRINT L# 255 REM CHECK FOR LEAP-YEAR 258 C=INT(Y/100+.00 5):YC=Y-100\*C 260 A(2)=28 262 IF YC=4\*INT(YC/ 4+, 1) THEN A(2)=29 264 IF YCK. 5 THEN A (2) = 28270 IF YCK. 5 AND C= 4\*INT(C/4+. 1) THEN A (2) = 29275 REM BUILD REMAINING DATE-LINES AND PRINT 280 EN=0 290 L#="" 300 FOR I=1 TO 7

310 DT=DT+1:IF DT>A (M)+.5 THEN EN=1:GOT 0 380 320 D1=INT(DT/10+.0 5):D2=DT-10\*D1 330 IF D16.5 THEN-L \$=<u>\</u>\$+" " 340 IF D10.5 THEN L \$=L\$+CHR\$(48+D1) 350 L\$=L\$+CHR\$(48+D 2) 360 IF I(6.5 THEN L \$=<u>L</u>\$+" " 370 NEXT I 380 PRINT L≇ 390 IF ENC.5 THEN 2 90 400 PRINT" " 405 REM DO AGAIN FOR NEXT MONTH

410 M=M+1:IF MD12.5 THEN M=1:Y=Y+1 420 N=N-1:IF ND.5 T **HEN 90** 430 PRINT TAB(100) 440 END 450 REM DATA: MONTH LENGTHS AND NAMES 460 DATA 31,28,31,3 0,31,30,31,31,30,31, 30,31 470 DATA \*\*\* JANUAR Y,\*\* FEBRUARY,\*\*\*\* M ARCH 480 DATA \*\*\*\* APRIL \*\*\*\*\* MAY , \*\*\*\* 490 DATA \*\*\*\* JULY /\*\*\*\* AUGUST,\* SEPT EMBER 500 DATA \*\*\* OCTOBE R,\*\* NOVEMBER,\*\* DEC EMBER

# AIM 65 Software



#### \* DISCOVER 6502 POWER \*

#### HELP!

HELP!!

9 Super utility programs for all AIM 65 programmers. HEX INPUT: Long and short-versions, used for entering hex bytes into memory. DUMP & HEXOUT: Print out your memory in two formats for easy checking or location of individual bytes. FIELD SORT: A field sorting routine that finds usage in many tasks including helping you organize your programming. RESTORE: A program which automatically restores your editor after you've -re-entered it improperly. This has been a real time saver for us. ONE STEP: Allows you to step thru the disassembly (K listing) one line at a time. SYMBOL TABLE: is for use with the assembler ROM (How can you do without one?) It prints the beginning and ending address, all in a handy format. RELOCATE: Is a powerful program which allows you to move or relocate programs or data in memory. All who write, adapt or prizele programs or subroutines will appreciate this. It allows you to place them wherever you'd like. You can even open up spaces right in the middle of a program for inserting missing, new, or additional data or instructions. A programmers dream.

GAIMS PAK I

SExciting games of skill for 1 or several players, using the full capabilities of the AIM 55 keyboard, display, and printer. HANGMAN: A challenging word game for 2 players. The AIM does the work and keeps score. SCORE 4: A challenging game in 3 dimensions. The printer shows the positions of the 2 players after each move. REACT: Your reflexes are tested in thousandths of a second. The display and keyboard are turned into a reaction timer. GOL-LUMS CAVERNS: Places you into the underground kingdom of the evil Wizard. You must move thru secret tunnels and cavern rooms avoiding traps. Wisterious mist, and the Wizard's spell. To capture the Wizard'you have only a few poison darts and your Magic computer to warn you when Evil is near. BINHEX: Teaches and tests your ability to convert Binary numbers into their Hexadecimal equivalents. Fun for budding programmers, and helps you to perfect a needed skill for fast and efficient programming.

Hexadecimal equivalents. Fun for budding programmers, and helps you to perfect a needed skill for fast and efficient programming.

MATH WHIZ

6 Programs dealing with numbers & math & the AIM 65. ADD & SUBTRACT: This powerful utility program turns your AIM 65 into a multiple precision calculator. TOTAL: Adds up to four decimal or hexadecimal numbers at a time. TEST MEMORY: Lets you really check out your RAM memory. FIBBONACCI: You learn about these important numbers as your AIM generates them in a series. DEC TO HEX: A multi-use program and algorithm for changing decimal numbers into their hex equivalents. TIMER: Makes your AIM 65 into a timer or a 12 or 24 hour clock, displaying or printing hours, minutes and seconds. A super demo of the power of the AIM 65.

YOU CAN NOW DEMON-STRATE THE POWER OF YOUR AIM 65 WITH CYBER-DYNE'S DYNAMIC ACTION SOFTWARE. ALL OF OUR PROGRAMS RUN ON 1K OR 4K AIMS. ALL SOFTWARE ON EZ-LOAD TAPE CASSETTES. COMPLETE TEXT, PROGRAM, & LOADING INSTRUCTIONS ARE INCLUDED.

"TAKE AIM" MANUAL, VOL I by JAMES HOYT CLARK of CYBERDYNE'S staff, coming soon-watch for it or write for info. A guide for all. Master AIM 65 hardware & software. A lab and learning manual, extension, clarification, & index to AIM 65 documentation. Over 30 programs. Explained & fully documented (games, math, utility, printer, display, & more).

> HOT LINE!! [801] 224-2745

#### **GAIMS PAK II**

GAIMS PAK II

6 Value packed games of skill and chance at less than \$1.75 each. Created for maximum enjoyment of the AIM 65 by you and others BRICKS: This program is unique because it learns from your mistakes and successes, while you play. It actually becomes "smarter" as you and the computer compete in a series of games. A real challenge to your skills of logic, deduction, and memory. TIC-TAC-TOE: Need we say more than the AIM 65 is a fair and impartial scorekeeper. CARDS: Gives you practice in when to hold-em as your AIM 65 deals 5 CARD STUD from an unmarked and randomly shuffled deck. LOGICAL ORDER: Tests your skills as a Master-Mind of reason and logic as you try to deduce a random 4 number sequence in the fewest number of tries. STARWAY 090: Places you at the controls of a crippled spacecraft. You must successfully pilot your craft back to the crippled spacecraft. You must successfully pilot your craft back to the mothership for a soft rendezvous. Your supply of fuel is limited and must be used with care to avoid disaster. ESP: Even computers can have ESP (or seem to). You mentally pick a number, answer a few questions (without disclosing the number), and your AIM 65 will guess the number correctly every time.

#### SHOW OFF

SHOW OFF

7 Programs (less than \$1.50 each) which show off all the features of the AIM 65. SIGNS: Lets you print 2 sizes of letters edgewise on the AIM 65 printer & make large banners. ROTATING BILLBOARD: Shows your messages as they rotate along the display. PRINTER WAVE: A good demo of user control of the AIM 65 printer. Starts you into graphics printing. PAPER ADVANCE: Gives you software control of the printer paper advance. See those last tew lines printed for a change LINE FEED: This time this hardware control of the paper advance. CURSOR DEMO: You can light all 16 segments of the displays or type in different display patterns. KEYBOARD INTERRUPT: Gives you hardware control of the keyboard. Scans the keyboard for key closures without interrupting program execution. ALL PROGRAMS in this section may be used in your own programs as subroutines or run on their own as demos. Full instructions included.

#### AIM 65 & 6502 RELATED PRODUCTS

I/O-TTY-CASSETTE connector board for the AIM 65 Plugs directly to AIM 65 app connector (J1). Includes AIM connector, TTY and recorder jacks, cable set for 1 recorder, PC board with traces & holes for LED indicators, switch input sensors, optoisolators, drivers, relays, audio amp, & AIM 65 I/O training course with software. AIM I/O BOARD-Kit \$19.75; Assembled \$22.75

CYBERDYNES 1980 CATALOG. HARDWARE, SOFTWARE, BOOKS, TRAINING ITEMS, GOODIES & R&D Services (FULL DESCRIPT \$1.00 (REFUNDABLE ON FIRST ORDER)—FREE WITH ORDER.



ORDER BY ITEM NAME AND PRICE. SOFTWARE CASSETTES \$9.75 EACH. POSTAGE IN U.S.A. & CANADA ORDER ORDER ORDER STITEM NAME AND PHICE. SOFT WARE CASSETTES \$3.70 EACH. TO ORDERS PAY BY

ORDER BY ITEM NAME AND PHICE. SOFT WARE CASSETTES \$3.70 EACH. TO ORDERS PAY BY

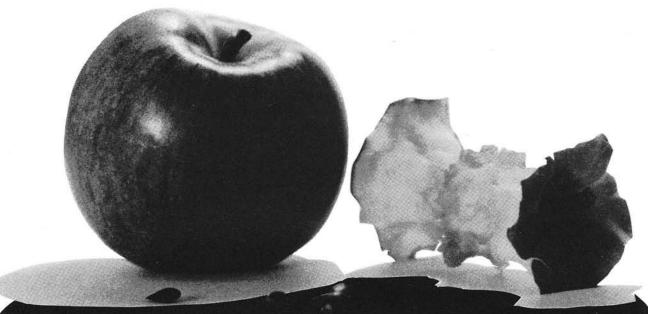
ORDER BY ITEM NAME AND PHICE. SOFT WARE CASSETTES \$3.70 EACH. TO ORDERS PAY BY ■ POSTAL MONEY ORDER. 10% DISCOUNT ON ORDERS OF 3 OR MORE ITEMS.

CUSTOM INDUSTRIAL MICRO-SYSTEMS OUR SPECIALTY. DEALER INQUIRIES INVITED.

#### SATISFACTION GUARANTEED \*



P.O. Box 1285 Orem, Utah 84057



# Introducing AppleSeed, our newest publication to whet your Apple\* appetite!

We invite you to subscribe to AppleSeed - the magazine that is to the Apple II\* what SoftSide is to the TRS-80\*\*. It offers the newest in software programming hints and ideas tailored especially for your computer. AppleSeed features challenging programs for both the do-it-yourselfer and the individual interested in pre-packaged programs and games... your own preview of the best available on the market today. A typical slice of AppleSeed consists of one major (new 16K) commercial level program (completely listed for your keying pleasure), accompanied by two or three applications for practical use or fun, supplemented by informative articles to polish your Apple\*. Get right to the core of your Apple\* needs and order AppleSeed today! 12 issues, 1 year, \$15.00. AppleSeed is the newest member of ...



**PUBLICATIONS** 

6 South Street, Milford, NH 03055 (603) 673-5144

\*A registered trademark of Apple Computers. \*\*A registered trademark of Radio Shack and Tandy Corp.

# **Bi-Directional Scrolling**

Everyone knows that a teletype only moves the paper in one direction - up. Likewise, the Apple display only scrolls one way - up. Now you can have scrolling in both directions - up and down - with these routines..

Roger Wagner SW Data Systems P.O. Box 582 Santee, CA 92071

By using the machine language routines given below, it is possible to scroll either text/gr page in either direction.

The up-scroll routine is derived from APPLE computer's red Reference Manual with the difference being that a zero-page location is referred to determine which page to scroll. The down scroll routine makes similar use of the same zero-page byte.

To use the routine a few entry conditions must be met:

- Load the binary routine into the \$300 page of memory starting at \$300.
  - Set pointers 6,7, and 8,9. If you want to bring new information onto the screen from RAM as you scroll 6,7 must point to the location in memory where the data to be loaded onto the top line of the screen will come from when you scroll the screen page down. Similarly 8,9 point to the place in memory to get the data for the bottom line when you scroll up.

If you want to use this routine to directly view memory, the easiest way to set the pointers 6,7 and 8,9 is to set 8 and 9 to the address you want to start viewing at. Put the low order byte in 8 and the high order in 9. (The screen height plus 1.) Then set 6,7 to the same value as 8,9 were originally, i.e., the low and high byte bring the starting address. Last of all, scroll back down one line to bring the starting address line into position as the first line of text visible at the top of the screen.

If you do not want new data brought onto the screen, then 6,7 and 8,9 will have

```
10 LOMEN: 3072
```

20 REM OR SET LOMEM: MANUALLY

BEFORE RUNNING.

30 CALL -936: INPUT "PAGE 1 OR 2?", PAGE

40 PRINT "INPUT ADDRESS ( < 32767) TO START AT: ": INPUT A

50 REM TO SCROLL WITHOUT BRINGING IN NEW DATA ENTER '0' FOR ADDRESS.

60 IF A#0 THEN 100: TEXT : CALL -936: POKE 34/1: REM FREEZE ONE BLANK LINE AT TOP OF SCREEN

70 YTAB 12: PRINT "(SAMPLE PG. 1 SCREEN DATA)"

80 POKE 6, 0: POKE 7, 4: POKE 8, 0: POKE 9, 4: REM BRING NEW SCREEN DATA FROM THAT BLANK LINE

90 GOTO 150

100 LB=R MOD 256:HB=A/256

110 POKE 5, PAGE\*4: IF PAGE=2 THEN POKE -16299, 0

120 POKE 8, LB: POKE 9, HB

130 FOR I=1 TO 25: CALL 768: NEXT I

140 POKE 6, LB: POKE 7, HB

150 KEY= PEEK (-16384): POKE -16368, 0

160 IF KEY=149 THEN CALL 768: REM RT. ARROW KEY TO SCROLL UP

178 IF KEY=136 THEN CALL 845: REM LFT. ARROW KEY TO SCROLL DOWN

180 IF KEY#136 AND KEY#149 OR A#0 THEN 190: POKE 6.0: POKE 7.4: POKE 8.0: POKE 9.4: REM RESET 6.7 & 8.9 TO POINT AT B LANK LINE

190 IF KEY#177 THEN 200: POKE 5,4: POKE -16300,0: REM '1' FOR PAGE 1

200 IF KEY#178 THEN 210: POKE 5.8: POKE -16299.0: REM '2' FOR PAGE 2

210 IF KEY#216 THEN 150: POKE -16300, 0: TEXT : CALL -868: PRINT "BYE.": END

1 **********	0300 R5 22	44 SCROLL	LDR WNDTOP	0362 E9 00	99 SBC	#\$99
2 * *	0300 H3 22 0302 48		PHR	0364 C5 22	100 CMP	WINDTOP
3 * APPLE SCROLLING ROUTINE *	0303 20 9E 03		isr vtabz	9366 39 9D	101 BMI	LDTOP
4 * *	9396 R5 28	47 NXTLN L		9368 48	102 PHA	
5 * BY *	0308 85 2A		STA BAS2L	0369 20 9E 03	103 JSR	YTABZ
6 * ROGER WAGNER *	939A A5 29		DA BRSH	036C B1 28	104 NXTCHR2 LDF	(BRSL), Y
7 *	030C 85 2B		STR BRS2H		105 STR	(BAS2L), Y
8 * THIS WILL LET EITHER PAGE *	030E R4 21		DY WINDWITH	0370 88	106 DEY	
9 * SCROLL IN EITHER DIRECTION. *	9319 88		EY	0371 10 F9	107 BPL	NXTCHR2
10 * IT IS PRIMARILY DESIGNED *	0311 68		PLA	0373 30 E1	108 BMI	NXTLN2
11 * TO FEED NEW SCREEN DATA IN *	0312 69 01		RDC #\$01	0375 A0 00	109 LDTOP LDY	#\$99
12 * FROM A GIVEN RANGE OF RAM. *	9314 C5 -23		CMP WNOBTH	0377 B1 06	110 LT2 LDR	(SCRNTP), Y
13 * *	0316 B0 0D		BCS LOBTM	0379 91 28	111 STA	(BRSL), Y
14 **********	0318 48	57 F	PHR	937B C8	112 INY	
15 *	0319 20 9E 03	58 3	JSR YTABZ	037C C4 21	113 CPY	WNDWDTH
16 *	031C B1 28	59 NXTCHR	LDA (BASL), Y	037E 90 F7	114 BCC	LT2
17 *	031E 91 2A	60 9	STA (BAS2L), Y	<b>938</b> 9 38	115 CRRCT2 SEC	
18 0BJ \$399	0320 88		ΣEΥ	0381 R5 06	116 LDA	SCRNTP
19 ORG \$300	9321 10 F9		BPL NIXTCHR	0383 E5 21	117 580	WNDWDTH
20 WNDLFT EOU \$20	8323 39 E1	63 E	BMI NXTLN	0385 85 06	118 STA	SCRNTP
21 WNDWDTH EQU \$21	0325 A0 00	64 LDBTM L	DY #00	0387 A5 07	119 LDA	SCRNTP+1
22 WNDTOP EQU \$22	0327 B1 08	65 LD2 L	DA (SCRNBTM), Y	0389 E9 00	129 SBC	#00
23 WNDBTM EOU \$23	0329 91 28	66 9	STA (BASL), Y	038B 85 07	121 STA	SCRNTP+1
24 CH EQU \$24	032B C8	67 1	INY	938D 38	122 SEC	CONVICTM
25 CV EQU \$25	032C C4 21		CPY WINDWOTH	038E A5 08	123 LDR	SCRNBTM
26 BASL EQU \$28	032E 90 F7		BCC LD2	9390 E5 21	124 SBC	MNOWOTH
27 BASH EQU \$29	0330 18	70 CRRCT (	CLC	0392 85 08	125 STR	SCRNBTM
28 BAS2L EOU \$2R	0331 A5 06		LDA SCRNTP	0394 A5 09	126 LDA	SCRNBTM+1
29 BR52H E0U \$28	0333 65 21		ADC WINDWOTH	9396 E9 00	127 SBC	#99
30 PAGE EOU \$05	0335 85 06		STA SCRNTP	0398 85 09	128 STR	SCRNBTM+1
31 * FOR APPLESOFT USE PAGE EOU \$1F	0337 A5 07		_DA SCRNTP+1	939A 68	129 RTS 130 BRK	
32 * PRIGE MUST HOLD \$04 FOR PG. 1	0339 69 00		ADC #00	039B 00	130 BRK 131 *	
33 * \$08 FOR PG. 2	0338 85 07		STA SCRNTP+1		132 *	
34 SCRNTP EQU \$06	933D 18		CLC LDA SCRNBTM	0390 A5 25	133 VTAB LDA	CV
35 * \$06, \$07 = LO/HI BYTES	933E A5 98			039E 20 A6 03	134 VTRBZ JSR	BASCALC
36 * OF START OF LINE JUST BEFORE	0340 65 21		ADC WINDWOTH STA SCRIBTH	93A1 65 20	135 ADC	WNOLFT
37 * TOP LINE	0342 85 08			93R3 85 28	136 STA	BASL
38 SCRNBTM EQU \$08	0344 A5 09		_DR SCRNBTM+1 RDC #00	93R5 60	137 RTS	CHOL
39 * \$08,\$09= LO/HI BYTES	9346 69 99			0310 00	138 *	
40 * OF START OF LINE JUST AFTER	0348 85 09		STA SCRNBTM+1 JMP VTAB		139 *	
41 * BOTTOM LINE	034A 4C 9C 03	7.4	AUL ALUO	03A6 48	140 BASCALC PH	IA
42 *		85 * 86 *		03A7 4A	141 LSR	-
43 *	934D 38	87 SCROLLI	ON SEC	03A8 29 03	142 AND	#\$03
	934D 30 934E A5 23		LDR WND8TM	038A 05 05	143 ORA	PAGE
	0350 E9 01		SBC #\$01	93AC 85 29	144 STA	BRSH
Change of Address?	9352 48		900 ##01 PHA	03AE 68	145 PLR	
We are still having problems with those of you who are moving around. Please	9353 20 9E 93		JSR YTABZ	93AF 29 18	146 RND	<b>#\$1</b> 8
notify us of any change of address so	0356 A5 28	92 NXTLN2		9381 90 92	147 BCC	BSCLC2
that you will not miss any issues. The	0358 85 2A		STA BRS2L	03B3 69 7F	148 RDC	#\$7F
Post Office does <b>not</b> return the undelivered copies, so we lose both the	035A A5 29		LDA BRSH	0385 85 28	149 BSCLC2 ST	
postage and the magazines. Send ad-	9350 85 28		STA BAS2H	9387 98	150 ASL	
dress changes to MICRO, c/o Carol Stark,	935E A4 21		LDY WNOWDTH	9388 9A	151 RSL	
Box 6502, Chelmsford, MA, 01824. Please include your old label or your subscrip-	9360 88		DEY	0389 05 28	152 ORR	BASL
tion number.	• 9361 68		PLR	9388 85 28	153 STA	BRSL
	920T 00	70	to Seal of Seal	9380 69	154 END RTS	j state eosti I J
				ENU BECENBI		

to point to a part of memory that contains 40 blank space characters. One way to do this is to freeze on blank line on either page 1 or 2, and then set 6, 7 and 8, 9 must be reset to that value each time the scroll is done. This is because normally the scroll routine updates 6,7 and 8,9 by thee screen width so as to remain synchronized with the screen display another technique is to just clear the top or bot-

Symbol Table						
WNDLFT	9629					
MNOMOTH	0021					
WHOTOP	9922					
WNDBTM	0023					
CH	0024					
CV	9925					
BASL	0028					
BASH	0029					
BRS2L	002A					
BRS2H	9928					
PAGE	9995					
SCRNTP	0006					
SCRNBTA	9998					
SCROLL	9398					
NXTLN	0306					
NXTCHR	031C					
LDBTM	0325					
LD2	0327					
CRRCT	0330					
SCROLLDN	034D					
NXTLN2	0356					
NXTCHR2	. 0360					
LDTOP	0375					
LT2	9377					
CRRCT2	0380					
VTAB	039C					
YTABZ	939E					
BASCALC	93 <b>8</b> 6					
BSCLC2	0385					
END	0380					

tom line to blanks each time a scroll is done.

- Location 5 must hold a 4 for page 1 scrolling, and an 8 for page 2.
- That's all. Now when you want the screen to scroll just 'CALL 768' to scroll up, and '845' to scroll down.

#### **Special Notes:**

If you are going to use page 2 of text/gr in Integer Basic, be sure to protect the variables with a 'LOMEM': 3072. This may be done before running the program, or if you know how, put as an early line in the program.

yram. *300 3	PE							
9399-		22	49	20	9F	83.	85	28
9398-					_	_		
9310-								
9318-								
9320-								
0328-								
9339-								
0338-								
9349-								
9348-								
9350-								
6358-								
8360-								
9368-				++				
0370-								
0378-								
0380-								
0388-								
0390-								
0398-								
93A0-								
03R8-	_							
03R0-								
93B8-								
				_				-

To use page 2 in Applesoft is more difficult, but can be done. First, location \$3AB in the machine code must be changed from \$05 to \$1F. Also, you must POKE 31 with a 4 or 8 as compared to the POKE 5 in Interger.

The real rub is that Applesoft programs normally begin in memory at \$800 (hex) which conflicts with page 2 use. The way around this is to do a 'POKE 104, 12: POKE 3072, 0' before loading your program. After loading do a 'CALL 54514' (unnecessary with DOS 32.). Unless you do a 'RESET', 'Control-B' other programs. Unfortunately, use of page 2 with the RAM version of Applesoft is to my knowledge impossible. (Sorry...)

If you wish to move the scrolling routine for some reason, the only location-dependent aspects of the code are 5 'JSR's and 1 'JMP' within it. Since these operations always reference absolute addresses they will have to be rewritten. Of course, if you have a relocate utility, it is that much easier.

For further enlightenment, see the sample Integer Basic program which makes use of the scrolling routine. Have Fun!

#### Location dependent:

\$303:	<b>JSR</b>	\$39E
319:	<b>JSR</b>	39E
34A:	JMP	39C
353:	<b>JSR</b>	39E
369:	<b>JSR</b>	39E
39E:	JSR	3A6

If page 2 of TEXT/GR is to be used, it must be protected by a 'LOMEM:3072' for integer BASIC, or a 'special Load' (as described in article) when using Applesoft.

Note: \$3AB must be changed from \$05 to \$1F for Applesoft.

#### SYNERGISTIC SOFTWARE

hesenls

THE MODIFIABLE DATABASE by Chris Anson & Robert Clardy

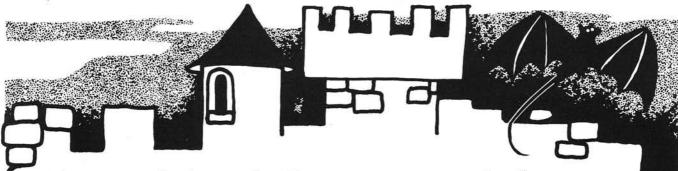
The Modifiable Database is a general purpose, user oriented database program that can be easily customized for your specific data management application. Create any number of application programs such as mailing lists, bibliography files, inventory controls, personnel files, accounting programs, etc. The only limitation is your own imagination.

The program uses fast and flexible machine language search and sort routines, provides for easy record editing, and can search or print up to 2 disks of records with a single command. All commands are invocable by a few keystrokes. There's never been an easier to use or more flexible data management program.

Applesoft program requires 48K and disk
Modifier Module 1 lets you add accounting and
numeric processing features to your program
Modifier Module 2 lets you format your output
in any way desired (columnar, standard forms,
suppressed fields, etc.)
\$49.50



AVAILABLE AT YOUR LOCAL DEALER OR SEND CHECK OR INQUIRY TO SYNERGISTIC SOFTWARE, 5221 120 AVE. S.E., BELLEVUE, WA 98006. WA RESIDENTS ADD 5.3% SALES TAX



Did you read about the Dungeonmaster who became so enchanted playing a real life version of Dungeons and

Dragons that he disappeared for a month?

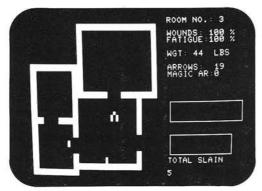
You'll be able to hold on to reality just a little better when you play the Dunionquest<sup>TM</sup> computer version, the greatest of all the role-playing fantasies.

But don't bet on it.

Sit at your computer. You're the hero. Enter the Dunjonquest "Temple of Apshai" and into the greatest fantasy adventure you've ever experienced. The Temple has over 200 rooms and catacombs in which lurk more than 30 kinds of monsters and beasts ready to do you in - in real time - before you can reach any of the 70 or so treasures waiting for the hero. You may spend days, weeks, months...the rest of your life ... striking at the forces of evil, or running from them, or calling on powers you can never completely understand. Always, always demonstrating in varying degrees your strength, constitution, dexterity, intelligence, intuition, the force of your ego.



Unlike chess or bridge or monopoly, this role-playing game - like other good role-playing games - is an experience rather than a game: it is not played so much as it's lived or experienced. Your alter ego goes forth into the world of demons and darkness, dragons and dwarves. Your character will do whatever you want him (or her or it) to do.



Actual photo of screen during a Dunjonquest game. In Room 3 in the Temple of Apshai, our hero observes two treasures unattended by dragons, monsters or demons...for the moment. He is completely free of wounds; he is not at all fatigued. He carries 44 pounds of armor and 19 arrows in his quiver. He has already slain five demons. Will he capture the treasures before moving on...or before the forces of darkness intercept him?

"The Temple..." comes complete with a superbly illustrated 56-page rule book and cassette program, designed to operate with the Level II 16K TRS 80, the PET 32K or the Apple II 48K (Applesoft) computer. Only \$24.95 complete, including shipping and handling on orders placed within the next 30 days. (Apple or TRS 80 disk available for \$29.95).

Dunjonquest's "The Temple of Apshai" is guaranteed to be the best version of Dungeons and Dragons/ Dragons and Dungeons. It's a product of the two guys who are Automated Simulations: Jim Connelley and Jon Freeman. Jim is a Dungeon Master, running continuous D & D campaigns. He's been a data processing professional with Westinghouse, GTE Sylvania, Logisticon...an expert in computer-based math-modeling and in simulation of complex phenomena. Jon is a game player, designer and author. He's a frequent contributor to Games magazine; his book, "The Playboy Winner's Guide to Board Games" is a paperback best-seller.

As we said, guaranteed: Guaranteed to be the best version; guaranteed that you'll be happy with it. Order now, use it for two weeks. If you don't enjoy completely this fantasy adventure experience that goes beyond all others, send it back to us. We'll refund your money in full; no questions asked.

Master Charge or Visa card holders: charge "The Temple of Apshai'' to your credit card. Just call our toll free number: (800) 824-7888, operator 861 (In California, call operator 861 (800) 852-7777. In Hawaii and Alaska, operator 861 (800) 824-7919) and you can begin enjoying your Dunjonquest game in days. Or send your check for \$24.95\* (or \$29.95)\* to



#### **Automated** Simulations

Dept. LW P.O. Box 4232 Mountain View,



# Software for the PET



DOMINOES

\$ 6.95



OTHELLO

\$ 9.95



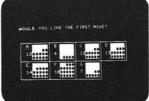
**LETTER SQUARES** 

\$ 6.95



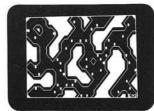
RPN MATHPACK

\$19.95



SUPER NIM

\$ 6.95



DIR/REF

\$ 6.95



SPACE WARS

\$ 9.95



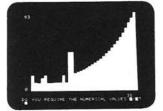
**GRAND PRIX** 

\$ 6.95



E.S.P.

\$ 9.95



FORECAST

\$ 9.95



\$ 6.95



BLOCKADE

\$ 9.95

Software



**DEPTH CHARGE** 

\$ 9.95



BASKETBALL

\$ 9.95



SLOT MACHINE

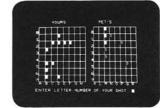
\$ 6.95



HOME ACCOUNTING

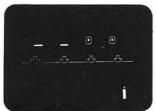
\$ 9.95

All orders include 3% postage and handling with a minimum of \$1.00. California residents include 6% Sales Tax. MASTERCHARGE VISA



BATTLE SHIP

\$ 9.95



SHOOTING GALLERY \$ 9.95

PET IS A TRADEMARK OF COMMODORE BUSINESS MACHINES, INC.

**PROGRAMMA** INTERNATIONAL, Inc. 3400 Wilshire Blvd. Los Angeles, CA 90010 (213) 384-0579 384-1116 384-1117

Dealer Inquiries Invited



ZAP

\$ 6.95

#### The SY6516 Pseudo-16 Bit Processor

While the 6502 is a great microprocessor as it stands, advances are being considered to make it even better. One of the approaches is to add some new capabilities such as some 16 bit operations, improved addressing, and more.

Randall Hyde 12804 Magnolia Chino, CA 91710

For those of you who may have wondered what the 6502 equivalent of the MC6809 would be, wonder no longer. Synertek is almost ready to ship the SY6516.

Synertek announced the 6516 almost a year ago, but due to production problems, it never quite made it. The 6516 was designed by Atari Inc. (back then it was to be called the 6509) for use with the Atari 400 and 800 computer systems. Unfortunately, Synertek was unable to deliver the chip in time for Atari to use it in their computers.

#### What is a Pseudo 16-bit Computer?

A pseudo 16-bit computer uses an internal 16-bit register arrangement, but externally it uses an eight bit bus. Sixteen bit data is multiplexed in, much like the Alpha Micro computer on the S-100 bus. In addition to the new 16-bit instructions, the 6516 maintains all of the 8-bit instructions of the 6502. You may reassemble your source files currently on the 6502 and run them directly on the 6516. All the information that I have recieved says that the 6516 is SOURCE code compatable with the 6502 and that it is OBJECT code incompatable with the 6502. I have heard rumors that Synertek is attempting to make the 6516 object code compatable, but quite honestly, I don't believe there is much chance of it happening.

Unlike the Motorola MC6809, which has a distinct set of 8-bit instructions and a distinct set of 16-bit instructions, the SY6516 contains a special register (the "Q" register) which toggles the system back and forth between 8-bit operation and 16-bit operation. In addition, all registers in the 6516 (A, X, Y, and SP) are

now 16-bits wide. The "Q" register contains four bits which may be programmed to put the accumulator in the 16-bit mode, the X-register in the 16-bit mode, the Y-register in the 16-bit mode, and memory in the 16-bit mode (for use with INC, DEC, ASL, ROL, ROR, LSR, etc.). If the accumulator is programmed to be in the 16-bit mode, then LDA will load the accumulator with 16-bits, the low order byte coming from the specified address and the high order byte coming from the specified address plus one. If the accumulator is in the 8-bit mode, then the LDA instruction behaves identically to the LDA on the 6502. The other registers (X, Y, and Memory) behave identically.

It does not take twice as long to perform a 16-bit instruction compared to the equivalent 8-bit instruction, as you might expect. Usually only one additional clock cycle is required. This means that 6516 code will run as much as 3 times faster than 6502 code performing the same operation.

In addition, several instructions have been "speeded up" over the 6502 equivalent. For instance, implied instructions now only require one cycle for complete execution (the 6502 requires 2). Several other instructions have been speeded up as well (see Table One).

Variety of addressing modes is what makes the 6502 as flexible as it is. The 6516 includes many more addressing modes in its instruction set. In particular, indirect addressing (without the indexed by Y or preindexed by X), 16-bit relative addressing (there is now a jump relative, so your code can be relocatable), and direct page addressing.

Direct page addressing is something

really special. It is available on the 6502 in a restricted form; on the 6502 it is called zero page addressing. Direct page addressing is different, in that any of the 256 pages in the 6516 address space may be used. The particular page is selected by the 8-bit direct page register "Z". The direct page facility should clear up many problems associated with zero page conflicts occuring in the 6502.

#### The New Instructions

The 6516 has a total of 114 instructions (compared to the 6502's 56). This gives a total of 255 different opcodes. Some of the new instructions are listed on the next page.

#### The User Flag

Bit 5 of the P register has been undefined to this point in the 6502. The 6516 utilizes this bit as a user defined flag. Included in the instruction set are instructions to set and clear this flag, as well as branch if set, and branch if clear. This user defined flag will prove to be a great help to users who are writing a boolean function. Up till now, the 6502 programmer had to use the carry or overflow flag. The user defined flag will help allieviate problems associated with the use of the aforementioned flags.

The 6516 instruction set was defined to allow maximum capability with the minimum number of instructions possible. For those of you who would really like to have seen an instruction of the form:

JMP (LBL,X) you may simulate this by:

LDY LBL,X YPC The instruction sequence still requires only 3 bytes (assuming LBL is a direct page reference) and the timing is 7 cycles which is only two cycles more than a straight jump indirect. This would execute just as fast as a JMP (LBL,X) instruction were it included directly in the instruction set.

For those of you who would like to have seen the auto-increment and auto-decrement instructions of the MC6809, once again they can be simulated by the 6516. For instance, the sequence LAX, INX simulates a post increment and INX, LAX simulates a pre-increment. These instructions require two bytes (the same as the 6809) and execute in 3 to 4 cycles (depending on whether you are in the eight-bit or 16-bit mode). This speed is comparable to the 6809.

The only advantage of the 6809 over the 6516 is the 6809 multiply instruction. However, a software multiply on the 6516 should execute fast enough so that it won't make that big a difference.

The addition of two stacks in the 6809 is no real advantage since you can simulate 2, 3 or even n stacks with one 16-bit stack pointer. Those of you writing machine interpreters (such as the UCSD Pascal Pcode interpreter) will be able to simulate a stack machine quite easily on the 6516.

In my opinion, Synertek has taken everything wrong with the 6502 and fixed it, in addition to adding several features which I had not even previously considered. The 6516 is easily the most powerful 8-bit processor available (with due respects to the Intel 8088 which I would rate "almost there"). This opinion, incidently, is not just my own. EDN rated the 6516 above all the 8-bit processors and even some 16-bit processors, several months ago. If Synertek does indeed make the 6516 processor object code compatable with the 6502, it will definitely make the 6516 something you shouldn't scoff at. Why? Because oncethis happen, 50,000 APPLE II computers will be upgradeable directly to a 16-bit processor and maintain software compatability with existing software. Likewise, the 70,000 or so PETs will be upgradeable and the OSI, and the KIM, and of course, the SYM, etc. etc.

The only fault I find with the 6516 is the assembly language mnemonics chosen by Synertek. They should have followed the example laid down by Motorola and used mneomics which specify the action, leaving the decision of where the data is coming from to the operand field.

I am currently writing a version of LISA (an interactive 6502 assembler for the AP-PLE II) for the 6516. I will maintain Synertek's syntax, however I will add several extensions to the syntax and in-

struction set to allow a much more regular syntax. This should prove to be a

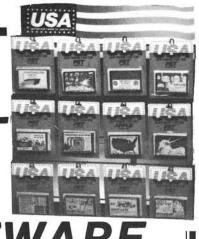
little more pleasant to the die-hard computer scientist.

#### The New Instructions

LDS LHA LHX LHY LAX LAY SAY	M-)S M-)AH M-)XH M-)YH M(X)-)A M(Y)-)A	(LOAD STACK POINTER FROM MEMORY) (LOAD HIGH ORDER ACL FROM MEMORY) (LOAD HIGH ORDER X-REG FROM MEMORY) (LOAD HIGH URLER Y-REG FROM MEMORY) (LOAD ACC INDIRECT THROUGH X REG) (STORE ACC INDIRECT THROUGH Y REG)
AXA AYA	A-M->A A+X->A A+Y->A	(ADD W/G CARRY) (SUBTRACT W/G CARRY) (ADD X REG TO ACC) (ADD Y REG TO ACC) (ADD ACC TO X REG) (ADD ACC TO Y REG) (ADD MEMORY TO X REG) (ADD MEMORY TO Y REG) (2'S COMPLIMENT ACC)
RLT RRT ASR RHL RHR RXL RXR RYL RYR		(ROTATE LEFT ACC) (ROTATE RIGHT ACC) (ARITHIMETIC SHIFT RIGHT ACC) (ROTATE AH LEFT THROUGH CARRY) (ROTATE AH RIGHT THROUGH CARRY) (ROTATE X REG LEFT THROUGH CARRY) (ROTATE X REG RIGHT THROUGH CARRY) (ROTATE Y REG LEFT THROUGH CARRY) (ROTATE Y REG RIGHT THROUGH CARRY)
TZA YFC PCY XHA XHY XHX XXY	Z-)AL Y-)PC PC-)Y AL(-)AH YL(-)YH XL(-)XH X(-)Y Qx(-)Qy	(TRANSFER Z TO ACC LOW) (TRANSFER Y REG TO PC) (TRANSFER PC TO Y REG) (EXCHANGE ACC BYTES) (EXCHANGE Y REG BYTES) (EXCHANGE X REG BYTES) (EXCHANGE X WITH Y REGISTAR)
SEF CLF LDQ SEV	1->F O->F M->G 1->V	(SET USER DEFINABLE FLAG) (CLEAR USER DEFINABLE FLAG) (LOAD G REGISTAR FROM MEMORY) (SET OVERFLOW FLAG)
BFS BFC JNE JEQ		(BRANCH IF FLAG SET) (BRANCH IF FLAG CLEAR) (JUMP IF NOT EQUAL TO ZERO 16-BIT RELATIVE) (JUMP IF EQUAL TO ZERO, 16-BIT RELATIVE)
PHD PLD PLX PLX PHY PHZ PHZ PHZ PHR PLR	A-)(S) (S)-)A X-)(S) (S)-)X Y-)(S) (S)-)Y Z-)(S), Q-)(S) (S)-)Q (S)-)Z	(16-BIT ACC PUSH) (16-BIT ACC PULL) (16-BIT X REG PUSH) (16-BIT X REG PUSH) (16-BIT Y REG PUSH) (16-BIT Y REG PUSH) (16-BIT Y REG PULL) (PUSH Z REG ONTO STACK, PUSH Q REG ONTO STACK) (PULL G FROM STACK, PULL Z FROM STACK) (COMBINATION OF PHD, PHX, PHY, AND PHZ) (COMBINATION OF PLB, PLX, PLY, AND PLZ)
BR1 BR2 BR3 BR4		(PERFORMS A JSR (\$FFF0)) (PERFORMS A JSR (\$FFF4)) (PERFORMS A JSR (\$FFF6))

(PERFORMS A JSR (\$FFF8) )

BR5



"Precise, humanized, well documented an excellent value" are the applauds now being given to United Software's line of software. These are sophisticated programs designed to meet the most stringent needs of individuals and business professionals. Every package is fully documented and includes easy to understand operator instructions.

DATABASE MANAGEMENT SYSTEM - A comprehensive, interactive system like those run on mainframes! Six modules comprising 42K of programming allow you to; create, edit, delete, display, print, sort, merge, etc., etc. - databases of up to 10,000 records. Printer routines automatically generate reports and labels on demand. 60 pages of concise documentation are included. Requirements - 16-32K PET and 2040 Dual Disk (printer optional). . . . Cost \$125

ACCOUNTS RECEIVABLE/PAYABLE - A complete, yet simple to use accounting system designed with the small businessman in mind. The United Software system generates and tracks purchase orders and invoices all the way through posting "controlled" accounts payable and accounts receivable subsystems.

Keyed Random Access file methods makes data access almost instantaneous. The low-cost solution for the first time computer user with up to 500 active accounts. Requirements - 32K PET, Dual Disk, any 80-column printer. . . . Cost \$175

CASH RECEIPTS & DISBURSEMENTS - Makes it a breeze to track all outgoing payments made by any type of business operation. Checks are tracked by number and categorized by type of expense. Sorting, summary, and audit trails make it easy to post to general ledger. This system also categorizes incoming receipts. Uses KRAM file access method. Requirements - 32K PET, Dual Disk (printer optional)....Cost

KRAM - Keyed Random Access Method - The new, ultra-fast access method for the PET Disk, provides keyed retrieval/storage of data, in either direct or sequential mode, by either full or partial key values. Written by United Software in 6502 machine code, and designed with the PET in mind, it exploits all the benefits of the PET Disk, allowing full optimization of your system. Eliminates the need for "Sort" routines! KRAM provides flexibility never seen on a micro before. KRAM is modeled after a very powerful access method used on large-scale IBM Virtual Storage mainframes. So "KRAM" all you can into your PET - it will love you for it. . . . Cost \$79.95

(Sublicenses available to software houses.)

PROGRAMS FOR	Super Startrek 14.95
ENTERTAINMENT	PET Music Box 29.95
Space Intruders ("Best Game of 1979")\$19.95 Jury/Hostage 12.50	UNITED SOFTWARE PROGRAMS FOR BUSINESS
Kentucky Derby/Roulette 9.95	Checkbook\$15.95
Alien I.Q./Tank 9.95	Mortgage 15.95
Tunnelvision/Maze Chase 14.95	Finance 12.95
Submarine Attack 9.95	Bonds 12.95
Battle of Midway 7.95	Stock Analyzer 22.95
Laser Tank Battle 9.95	Stock Options 24.95
Swarm 14.95	6502 Macro Assembler 49.95

Look for the RED-WHITE-BLUE United Software Display at your local computer dealer, or send check or moneyorder, plus \$1.00 shipping to:

#### UNITED SOFTWARE OF AMERICA

750 Third Ave. New York, N.Y. 10017

Dealer inquiries invited

#### DAKINS UTILITIES

TODAY'S DATE 89/18/79

1. THE LISTER

- 2. THE PEEKER
- 3. THE CRUNCHER
- 4. THE TEXT FILE COPY
- S. THE PROMPTER
- 6. THE CALCULATOR
- 7. THE DISKETTE COPY
- R. CHANGE TODAY'S DATE

ENTER YOUR SELECTION -> .

TODAY'S DATE 12/17/79

- 1. THE SCREEN PRINTER
- 2. THE ARRAY EDITOR
- 3. THE COPIER 4. THE PATCHER
- 5. THE LINE CROSS REFERENCE
- 6. THE VARIABLE CROSS REFERENCE
- 7. THE CALCULATOR II
- B. CHANGE TODAY'S DATE

ENTER YOUR SELECTION >.

The Cruncher removes REM statements and compresses code in Applesoft programs. The Prompter is a powerful data entry subroutine that can handle both string and numeric data. Options include using commas, decimal points, and leading zeros with right-justified numerics. A A maximum field length can be specified to prevent overflow in both numeric and alphanumeric fields. The Diskette Copy formats an output disk, copies each track, and verifies that the output matches the input. PLUS FOUR MORE UTILITIES TO AID YOUR OWN PROGRAMMING. Suggested Retail Price for Dakin5 Programming Aids is \$39.95.

The Copier copies absolutely any kind of file or program from one diskette to another. The Variable Cross Reference produces a cross-reference for all variable names used in an Applesoft BASIC program. The Line Cross Reference creates a cross-reference for an Applesoft BASIC program, showing where a given line is referenced by GOTO, GOSUB, THEN, or LIST statements. The Patcher allows you to display any sector of a diskette, and then to update any data within that sector. PLUS THREE MORE UTILITIES TO FACILITATE YOUR OWN PROGRAMM-ING. Suggested Retail Price for Dakin5 Programming Aids II is \$49.95.

Each utility package includes a program diskette and very complete documentation. The hardware requirements are an Apple II, 48K of memory, 2 Disk II's, and a printer. Languages are Applesoft/Assembler.

See your Apple dealer or contact Dakin5 Corporation, P.O. Box 21187, Denver, Colorado 80221. Telephone: (303) 426-6090 Dakin5 developed the business application software The Controller of Apple Computer Inc.

#### DISK DRIVE WOES? PRINTER INTERACTION? MEMORY LOSS? ERRATIC OPERATION? DON'T BLAME THE SOFTWARE!







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, \*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 . . . . . . . \$54.95 \*SUPER ISOLATOR (ISO-3), similar to ISO-1A except double filtering & Suppression . . . . \$79.95 \*ISOLATOR (ISO-4), similar to ISO-1A except unit has 6 individually filtered sockets . . . . \$93.95 \*ISOLATOR (ISO-5), similar to ISO-2 except unit has 3 socket banks, 9 sockets total . . . \$76.95 \*CIRCUIT BREAKER, any model (add-CB) Add \$ 6.00 \*CKT BRKR/SWITCH/PILOT any model ..... Add \$11.00

PHONE ORDERS 1-617-655-1532

Electronic Specialists, Inc. 171 South Main Street, Natick, Mass. 01760

Dept.mi

VISA

#### Announcing...

# OPTIMIZED SYSTEMS SOFTWARE UPGRADE YOUR APPLE II® WITH A NEW SYSTEMS SOFTWARE PACKAGE

- Unified Operating System
- Disk File Manager
- Commercial Basic
- Editor/Assembler/Debugger
- Data Base Manager

Optimized Systems Software does not use Apple DOS<sup>®</sup>. OSS is a unified and complete systems software package with its own Operating System and File Manager. The Operating System, the File Manager and the Basic combined use only slightly more RAM than Apple DOS<sup>®</sup> alone. Requires 48K Apple II <sup>®</sup> with Disk II.

#### **Operating System**

- Byte and Block I/O
- Simple User Interface
- Simple Device Interface (create your own)

#### Basic

- Nine Digit Precision DECIMAL Floating Point
- 32K Byte Strings
- Variable Names to 256 significant characters
- I/O Interface Statements (no PRINT "control-D...")

#### File Manager

- Open, Read, Write, Delete, Lock, etc.
- Random Access via Note & Point
- File Names of Primary. Ext type

#### Editor/Assembler/Debugger

- Line Editor
   (Edits Basic programs, too)
- Mini Assembler
- Maxi Assembler
- Disassembler
- Step, Trace, etc.

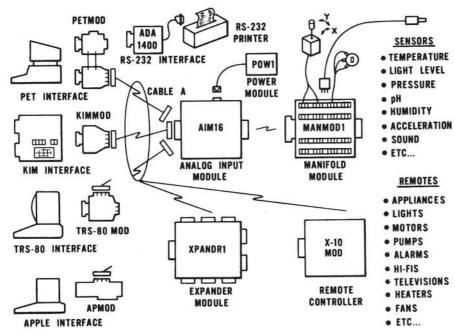
Available NOW at Special Introductory Prices	April 15	April 15
<ul> <li>Operating System + File Manager + Basic</li> </ul>	\$49.95	\$69.95
<ul> <li>Operating System + File Manager + ASM</li> </ul>	\$49.95	\$69.95
<ul> <li>Operating System + File Manager + Basic + ASM</li> </ul>	\$89.95	\$109.95
<ul> <li>Operating System + Data Base Manager</li> </ul>	(2nd Q)	

Order today. Add \$2.00 for shipping & handling. California residents add 6% sales tax. Visa/Mastercharge welcome. Personal checks require 2 weeks to clear.

Note: Apple II®, Apple DOS® are trademarks of Apple Computer, Inc.

Optimized Systems Software Shepardson Microsystems, Inc. 20395 Pacifica Dr., Suite 108 Cupertino, CA 95014 (408) 257-9900

# MICROCOMPUTER MEASUREMENT and



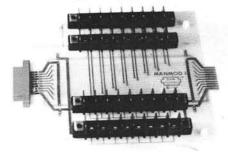
The world we live in is full of variables we want to measure. These include weight, temperature, pressure, humidity, speed and fluid level. These variables are continuous and their values may be represented by a voltage. This voltage is the analog of the physical variable. A device which converts a physical,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

Computers do not understand voltages: They understand bits. Bits are digital signals. A device which converts voltages to bits is an analog-to-digital converter.

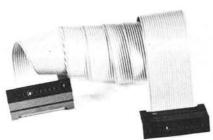
Our AIM 16 (Analog Input Module) is a 16 input analog-to-digital converter.

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

#### Connectors







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

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

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

#### Analog Input Module



The AIM 16 is a 16 channel analog to digital converter designed to work with most microcomputers. The AIM16 is connected to the host computer through the computer's 8 bit input port and 8 bit output port, or through one of the uMAC SYSTEMS special interfaces.

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

Power requirements are 12 volts DC at 60 ma.

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

#### **TEMPSENS**



This module provides two temperature probes for use by the AIM16. This module should be used with the MANMOD1 for ease of hookup. The MANMOD1 will support up to 16 probes (eight TEMP-SENS modules).

Resolution for each probe is 1ºF.

#### XPANDR1

The XPANDR1 allows up to eight Input/
Output modules to be connected to a computer at one time. The XPANDR1 is connected to the computer in place of the AIM16. Up to eight AIM16 modules are then connected to each of the eight ports provided using a CABLE A24 for each module. Power for the XPANDR1 is derived from the AIM16 connected to the first port.



# CONTROL for PET, Apple, KIM, and AIM



# Computer Interfaces and Sets





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

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

#### INTRODUCING SUPER X-10 MODULE

Open a door or window and turn on a light, tape recorder, alarm!

Control lab equipment. CLOSE THE LOOP on the real world.

# AN INEXPENSIVE CONTROL SOLUTION FOR

HOME SECURITY
ENERGY CONSERVATION
GREENHOUSES
ENVIRONMENTAL CONTROL
INDUSTRIAL CONTROL
LABORATORIES

#### SUPER X-10 MOD SPECS

- Remote controller
   Controls up to 256 different remote
   devices by sending signals over the
   house wiring to remote modules. Uses
   BSR remote modules available all over
   the USA (Sears, Radio Shack, etc.).
   Does not require BSR control module.
   Does not use sonic link.
- Clock/calendar
   Time of day hours, minutes, seconds
   Date month, day automatically corrects for 28,29,30 and 31 day months.

   Day of the week.
- Digital inputs/outputs
   inputs TTL levels or switch closures.
   Can be used as triggers for stored
  - 8 outputs TTL levels

sequences.



PLUS: CLOCK, CALENDAR, REMOTE SEQUENCE TRIGGERS

- 4. Computer interfaces S-100: Requires one 8-bit input port and one 8-bit output port. Requires cable assembly. PET, APPLE, TRS-80, KIM, SYM, AIM65: Plug-in sets available - no cable assembly required. Other: same as S-100
- Self-contained module in metal case with its own power supply. Physical size approximately 5X6X2.

Price (until April 30, 1980): \$199.00 (S-100), \$249.00 (other)

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

AIM16 (16 channel-8 bit Analog	
Input Module)	179.00
POW1a (POWer module-110 VAC)	14.95
POW1e (POWer module-230 VAC)	24.95
ICON (Input CONnector)	9.95
OCON (Output CONnector)	9.95
MANMOD1 (MANifold MODule)	59.95
CABLE A24 (24 inch interconnect	
cable)	19.95
XPANDR1 (allows up to 8 Input or	
Output modules to be connected to	оа
computer at one time)	59.95
TEMPSENS2P1 (two temperature p	robes,
-10°F to 160°F)	49.95
LIGHTSENS1P1 (light level probe)	59.95

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

The following sets include one AIM16, one POW1, one OCON and one MANMOD1.

AIM16 Starter Set 2a (110 VAC) 239.00

AIM16 Starter Set 2e (230 VAC) 249.00

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



#### **Order Form**

CONNECTICUT microCOMPUTER, Inc. 150 POCONO ROAD BROOKFIELD, CONNECTICUT 06804 TEL: (203) 775-9659 TWX: 710-456-0052

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

QUANTITY	DESCRIPTION	PRICE	TOTAL
			-
		_	-
			-
			-
		_	
SUBTOTAL			
Handling and	shipping — add per ord	ler	\$3.00
Foreign order	s add 10% for AIR posta	ige	
	add 7% sales	tax	
TOTAL ENCL	OSED		
AME			
DDRESS			
ITY			
TATE		ZIP	
ISA O M/	C O Expiration da	ate	
ard number_			



#### MICRO SUBSCRIBERS SPECIAL

#### THANK YOU, COMMO DORE ...

Just about a year ago, Skyles Electric Works introduced a 15 inch wide keyboard with all PET functions on nice, plump, full-sized keytops with torsion spring action. S.E.W. knew that most owners of the PET 2001-8 couldn't remain happy with the undersized keytops and unsatisfactory spring action.

Now you have announced that, effective in January of 1980, all PETs and CBMs would feature full sized keyboards.

So, thank you, Commodore, for confirming that S.E.W. was right all along.

Meanwhile, S.E.W. hasn't been resting...and now offers with the Big an 18 inch ribbon cable at no extra charge...and a cassette tape containing BASIC and machine language programs that convert the Big to ASCII or typewriter operation, including lower case characters, and upper case through the shift key. Even a shift lock, quotations in the proper place, numbers across the top row.



All keytops double shot for lifetime durability. Switching action uses gold cross-point contacts; torsion springs are gold plated. Housing is black heavy gauge aluminum. Entire unit can be rinsed to clean under water and left to dry. 120 day warranty. Only \$125.00\*

MICRO SUBSCRIBERS SPECIAL TILL MAY 15, 1980 \$99.50\*

# Wouldn't You Love Your PET 2001-8 Even More With the Skyles Big KeyBoard?

Enclosed is my	MICRO address	label showing	subscription	number,	please	send
Big Keyboar	d(s).Enclosed is	a check \$	*. Charge my	VISA, N	lasterch	arge
(circle one) #		Expires				J

\*California residents: please add 6% or 6.5% sales tax as required.



Skyles Electric Works | Mountain View, CA 94041 (408) 735-7891

231E South Whisman Road

# **PET Keysort**

One of the most useful operations to perform on a real data base is a keysort. On the PET, due to some problems in the 'garbage collection' procedures, sorting string arrays can become very time consumming. A complete, general purpose keysorting program is presented which has many useful features and is efficient.

Rev. James Strasma 120 West King Street Decatur,IL 62521

One of the most needed features of any business database program is a good sort routine. On the PET computer, there is also a real need for a way to sort string arrays without changing the strings. This is due to a quirk in the PET's "garbage collection" routine. PET was designed so every time a string is changed, a new string is created. Old versions are erased only after memory is filled. Then it deletes all the unneeded strings at once. As string space increases, collection time increases dramatically. With 24K of strings in memory, it can take several minutes.

Until future ROM's speed this process, it is best to avoid unneeded string manipulations. This makes a different sort program essential. For example, in an attendance program I developed, a heapsort is used. The heapsort itself takes about 20 minutes to sort 500 records. However, garbage collection adds another 2 hours! Clearly this is unacceptable.

One solution would be to define another array of integer string pointers, and sort that array. This would avoid moving strings entirely. As it happens. BASIC already stores its strings that way. Each string array is a table or pointers to another array of the actual strings. The pointers are above the program in memory, at the end of the variables. The strings are usually at the top of memory, though they may be anywhere.

I wrote a 'pointer sort' using the pointer table. It worked, but took too much memory, and had to be part of each program using it. I decided to put it in machine language instead. In the final form, it uses just under 1K of memory, at the top of memory. It resets BASIC's top of memory pointer to protect itself from

BASIC, and saves a copy of PET's zerobase to protect basic from the program. The other main features of KEYSORT, are as follows:

- extreme speed
- 2. simple operation
- 3. has defaults for all options
- 4. works with BASIC arrays
- 5. Remains until PET is reset
- accepts any number of fields within a string
- sorts any specified string array in memory
- accepts any character as a field marker
- both strings and fields may individually vary in length
- extensive error ckecking

The two BASIC demonstration programs will illustrate these features. Listing #1 creates an array of random strings to sort. It does 3000 names in 28 seconds. Once you create an array to sort, merely enter 'sys(31841)' to sort it, either directly or from a program. Later, when you are ready to sort on an array other than the first in memory, try out listing #2. It uses all of KEYSORT's options at once. First, it selects the 'a\$' array as the one to sort, ignoring all other arrays. Second, it selects the '> character as the marker between fields. Using a marker allows one string to hold about 128 separate fields at once. The array may be instantly resorted on any of these fields, as shown in sample run #2, which sorts on field #4, actually the fifth field, since there is a field #0.) You may sort by name one minute, by birthdate the next and by zip code after that.

There is no need for strings to have a fixed length. Nor is there any need for fields within strings to be any special

length. This avoids any waste of array space. KEYSORT's default field marker is the [tab] character, chr\$(9). This is easily changed, as shown in listing #2. Also there need not be any end of field marker unless you select one. Listing #1 works fine without fields. If time is very important to you. Note that using fields doubles the sort time. In return, it allows you to maintain a single data base, for several programs, and sort only the fields needed by the particular program currently in use. That saves a lot of typing time.

When you study the asssembly source listing of KEYSORT, you will note a subroutine called 'spg'. This is a routine any 6502 owner can use to save up to half of zero base. By placing it at the end of the normal program flow, it only has to be called once, and its ending 'rts' then returns to BASIC.

After you assemble and save a copy of KEYSORT, call it without any arrays in memory. You will immediately see:

?array error ready

This is KEYSORT's error message. Here it means no array was found. However, in the process, it reset Himem to protect itself from BASIC. You should do this each time you load KEYSORT, before defining strings. Otherwise they will overwrite the program. Note that if another program has already moved Himem lower than KEYSORT needs, the program leaves it alone.

If you see the '?array error' message at other times, one of several things has gone wrong. Perhaps there is no array to sort, ie. you cleared the variables. Or maybe the array has more than one

dimension—only one is allowed. (Unsorted arrays may have all the dimensions you wish.

Then again, you may have erred in poking in KEYSORT: that becomes the default for future sorts. Note that at the end of listing #2, the seven command locations are reset to zero. Unless the next sort uses the same KEYSORT features or more, you will need to zero those functions not desired in the next sort.

Both the assembly listing and the hex dump of KEYSORT here are for a 32K PET. However, the program is easily relocatable. There is no data in the body of the program, and the program does not change itself. To relocate it, merely change all of the high order bytes of 3 byte instructions, except for the one that jumps to \$C357 at \$7 of 8. This is a call to the new ROM's error message printer. Table #1 shows all the locations to change for relocation at the top of all PET model's memory. If you have an 'old ROM' PET, (8K '79 or earlier vintage), you will need to make the changes listed in table -2. You will also be limited to 256 element arrays, as the old ROM's couldn't handle more elements than that at once.

Other 6502 users with Microsoft may be able to adapt KEYSORT to their needs. My local 6502 group is converting it to the Apple, which uses a similar memory structure. It may help you to know how PET stores arrays. Each array starts with 7 housekeeping bytes. The first byte of the first array's housekeeping is addressed by 'aras' in BASIC (\$2c-2d,) low and high.) The last aray ends just before the address in 'eara', (\$2e-2f). The first 2 housekeeping bytes in each array contain its name. If it is a string array, \$80 will be added to the second character of the name as a flag. Even if there is no second character, byte 2 will contain \$80. Bytes 3 and 4 are the low and high bytes of the offset from the start of the current array to the start of the next one. Byte 5 is the number of dimensions in the array, 1-3. Bytes 6 and 7 are the HIGH and low bytes respectively of the number of elements in the array. (This is backwards from the usual 6502 format.) There will be 1 more element than in the DIM statement, as the 0th element counts too. The 0th element begins immediately after the housekeeping bytes. Each element consists of 3 bytes. The first is the length of the string. The other 2 are the low and high bytes long. Also, when first dimensioned, all the length bytes and address bytes are set to zero.

I wont't try to fully explain the BASIC and assembly listings of KEYSORT; they are fully commented. The only unusual feature in the BASIC programs is the use of PET's built-in 60th of a second jiffy clock, TI. When entering the assembly source, save \$3500 for the text file and

\$0200 for labels. If you have less room available, delete some comments.

If you have questions about KEYSORT, or need help, write me at the above address. Please include a stamped reply envelope. If you want a custom tape copy of KEYSORT, please send along \$5 for my time. Also, specify the starting or ending address you wish, and which ROM set you have.

Table 1: Locations to change on relocation

\$7C is found at:	\$7C62
7EFF	7F3A
\$7D is found at:	\$7C75
7CF5	7D33
7EAD	7EDC
\$7E is found at:	\$7DF7
7E48	7E87
\$7F is found at:	\$7D44
7D8F	7DA4
7DAA	7DC7
7E0C	7E68
7E9A	7EB8
	7ECB

#### To relocate for:

100 REMD SORT DEMO #1

PET 4K, change 7s to 0s PET 8K, change 7s to 1s PET 16K, change 7s to 3s Code will reside at Himem.

#### Table 2: Changes for using old ROMs Source Changes:

Line 430 ARAS .DE \$7E
 Start of array space [650 & 670]
Line 440 EARA .DE \$80
 End of array space [1080 & 1120]
Line 450 HIM .DE \$86
 End of memory [560, 590, 610, & 630]
Line 460 ARER .DE \$85
 Offset into error table [1320]
Line 470 ERRP .DE \$C359
 Error msg. and stop [1330]

#### **Object Code Changes**

\$7C77 = \$7E	\$7C7B = \$7F
\$7CC7 = \$81	\$7CCF = \$80
\$7C64 = \$87	\$7C6A = \$87
\$7C6E = \$86	7C72 = 886
\$7CF7 = \$85	\$7CF9 = \$59

```
110 PRINT"SAMPLE RUN FOR LISTING #1":PRINT
120 SZ=10:REM> ARRAY SIZE
130 DIM A$(SZ)
140 REMO MAKE UP STRINGS TO SORT
150 FOR I=0 TO SZ
160 A$=""
170 : FOR J=1 TO 10*RND(0)+1
180 :
        A$=A$+CHR$(65+26*RND(0))
190
      NEXT
200 :
      A$(I)=A$
210 :
      PRINT I,A$
220 NEXT
230 T1=TI:REM> ZERO THE CLOCK
240 SYS(31845):REM> SORT
250 T2=TI:REM> STOP THE CLOCK
260 PRINT:PRINT"ORDER AFTER SORTING":PRINT
270 REM> PRINT THE SORTED STRINGS
280 FOR I=0 TO SZ
290 : PRINT I,A$(I)
300 NEXT
310 REM) BRAG ABOUT THE TIME REQUIRED
320 PRINT:PRINT"TIME TO SORT="(T2-T1)/60"SECONDS
```

READY.

```
100 REMO KEYSORT DEMO #2
 110 PRINT"SAMPLE RUN FOR LISTING #2":PRINT
 120 SZ=10:REMD ARRAY SIZE
 130 F1=4:REMD FIELD # TO SORT BY
 140 D1=ASC(">"):REM> FIELD DELIMITER
 150 S$="A$":REM> SORT ARRAY NAME
 160 ZC=32731:REMD START OF Z.P. COPY
 170 NMFL=ZC+2:REMD FLAGS GIVEN ARRAY
 180 DFLG=ZC+3:REM> FLAGS NEW DELIM.
 190 DLIM=ZC+4:REM> STORES DELIMITER
 200 FDFL=ZC+5:REMD FLAGS KEY FIELD
 210 FLDS=ZC+6:REM> STORES KEY FIELD #
 220 DIM B$(10,2):REM> GARBAGE
 230 DIM CX(10)
240 DIM D(10)
 250 DIM A$(SZ):REM> ACTUAL SORT ARRAY
 260 REMO MAKE UP STRINGS TO SORT
 270 FOR I=0 TO SZ
 280 : A≸=""
 290 : FOR K=1 TO 5:REM> # OF FIELDS
 300 : : FOR J=1 TO 10*RND(0)+1
310 : : : A$=A$+CHR$(65+26*RND(0))
 320 : : NEXT
 330 : : REMO FIELD DELIMITER
340 : : IF KC5 THEN A$=A$+CHR$(D1)
 350 : NEXT
360 : A$(I)=A$
 370 : PRINT I,A≸
380 NEXT
 390 REM> TELL SORT FIELD # IS GIVEN
400 POKE FDFL, ASC("#")
410 REMO TELL SORT WHICH FIELD TO USE
420 POKE FLDS,F1
430 REMO GIVE SORT NEW DELIMITER
440 POKE DLIM, D1
450 REMO TELL SORT TO CHANGE DELIMITERS
460 POKE DFLG, ASC("%")
470 REMO CHANGE SORT ARRAY NAME TO BASIC
480 REMO TELL SORT SETTING NAME
490 POKE NMFL, ASC("$")
500 POKE ZC,ASC(S$):REM> CHARACTER #1
510 S2=ASC(MID$(S$,2)):REMD & #2
520 IF S2=ASC("$") THEN S2=128
530 POKE ZC+1,S2
540 T1=TI:REM> ZERO THE CLOCK
550 SYS(31841): REMD SORT
560 T2=TI:REMD STOP THE CLOCK
570 REMO CANCEL SPECIAL OPTIONS
580 FOR I=ZC TO ZC+6
590 : POKE I/0
600 NEXT
610 PRINT:PRINT"SORTED ON FIELD #"F1:PRINT
620 REMO PRINT THE SORTED STRINGS
630 FOR I=0 TO SZ
640 : PRINT L.A$(I)
650 NEXT
660 REM> BRAG ABOUT THE TIME REQUIRED
670 PRINT:PRINT"TIME TO SORT="(T2-T1)/60"SECONDS
READY.
```

#### **Classified Ads**

C1P software-Carz/Chase real-time games. \$6.95; Graphics/Billboard \$4.95; two screen clears, one for BASIC programs, one for imm. mode \$7.95; Learning OSI Basic \$14.95. COD or Money order, SASE for catalog. Order from:

OMEĞAWARE SOFTWARE 1418 Hanson Drive. Normal, IL 61761 (309) 454-1507

ZIPSOFT Presents: Number Match. A program to stretch your memory. Match sequence of numbers, colors and sounds. Different speed levels. Requires Apple II with Integer Basic and 16K. Tape \$8.95, Disk (32K req.) \$15.00. Order from:

Z. Gittler 50 Remsen Avenue Monsey, N.Y. 10952

Apple II program to output to and input from ASR 33 or 35 Teleype. Gives multiple line feeds at end of each page and waits for you to tear off roll paper or insert new single sheet. Uses game connector. Listing and instructions, \$2.00. Order from:

Ken Ellis RD 8, Box 344 York, PA 17403

Ohio Scientific. Expansion Help for C1P, C2, C4, C8 and C3. Parts, cables, boards, IC's, books, schematics. Contact:

Silver Spur Electronic Communications Co. P.O.Box 365 Chino, CA 91710

Attention APPLE Owners! A complete General Ledger, allowing 9 departments and 10 levels of subtotals. Virtually unlimited capacity. Will process 1,000 postings into 70 accounts in less than 30 minutes. Contact:

Small Business Computer Systems 4140 Greenwood Lincoln, NE 68504

Brand New Commodore PET 2001-8, 32K, Video Display. Will sell at best offer; 216/543-7785.

Dr. John W. Cook 8670 Tanglewood Trail Chagrin Falls, OH 44022

#### Sample Run for Listing 2

#### Sorted on Field 4

### BCHRFEDAKKYTGFCHDDGDXTDNTUTHO

1 YKNJZBKTDNJVSMVIDUOFDNYCCINWGVGDYSCF

2 IHRQDWAGQWNEXDXDMPDQGCORK

3 QJDHQMWUEDUKQCDGVKPMFDVZ

4 IDGZTWMDRLHDXSBPDMLWDPWO

5 EDFLDQDRDTDSIVR

6 FDWODZNBZOHCJGDPTQDLIODZVGLAH

7 GDTFZDLKPNDNPCHNZSXMVDGSKCDBWIDSZ

8 NSDNDTKBODLPRJWBODVLCDDFIDOXU

9 SQPIFSBRDGKVSJCHDWDHCUQDWQQDPCDZSIGN

10 BLDQENODAYDZDLODZVGLAH

10 FDWODZNBZOHCJGDPTQDLIODZVGLAH

10 FDWODZNBZOHCJGDPTQDLIODZVGLAH

11 GDTTZDLKPNDNPCHNZSXMVDGSKCDBWIDSZ

12 YKNJZBKTDNJVSMVIDUOFDNYCCINWGVGDYSCF

13 GDTFZDLKPNDDHCUQDWQQDPCDZSIGN

14 SQPIFSBRDGKVSJCHDWDHCUQDWQQDPCDZSIGN

15 FDWODZNBZOHCJGDPTQDLIODZVGLAH

TIME TO SORT= .0833333333 SECONDS READY.

#### and the state of t

```
pet keysort
0010 ;
0020 ;
0030 ; a multi-key sort for pet basic arrays
0040 :
0050 ;
           by rev. james strama
0060 ;
            120 w. king st.
0070 ;
0080 ;
              decatur, il. 62521
0090 ;
0100 ;
          as of feb. 14, 1980
0110 ;
0120 ;
0130 sart .ba $7c61
                         ;sys(31841)
0140 ;
0150 ;
0160 /first 5 var.s poked from basic
0170 arnm .de $00
                                istores array name
             .de $02
0180 nmfl
                                 jarray selected flag
             .de $03
0190 dfla
                                 Jdelimiter set flag
             .de $04
0200 dlim
                                 jdelim. char.
             .de $05
0210 fdfl
                                jkey field set flag
0220 flds
            .de $06
                                 //sort field #
0230 ;
0240 ;
0250 ; most var.s as in basic heapsort
0260 i .de $07
             .de $⊍9
0270 j
             .de $0b
0280 k
             .de $0d
0290 1
             .de $0f
0300 ln
                                 ;1 by-≸ lengths
0310 ln1
           .de $10
                                 ; "
0320 ln2
             .de $11
0330 n
             .de $12
                                 Jelements in array
             .de $14
0340 r1
                                 3 by-temp. registers
0350 r2
             .de $17
                                 ; 11
0360 r3
            .de $1a
                                j^{(n)}
0370 r4
             .de $1d
                                1 11
0380 s
             .de $20
0390 v1
             .de $23
                                 Spointer start-3
0400 j
0410 ;
0420 (rom-dependent var.s
```

```
0430 aras.de $2c;start of array space0440 eara.de $2e;end of array space0450 him.de $34;end of memory0460 arer.de $80;offset w/i error table0470 errp.de $c357;error msg. & stop
                                                                                                             0480 ;
                                                                                                             0490 ;
                                                                                                             0500 jother labels
                                                                                                            0510 dch     de $09
0520 locs    de $24
                                                                                                                                                                                                                                                                                                                                                         itab char.
                                                                                                                                                                                                                                                                                                                                                        # of locations to flip
| 0538 | 0548 | 0548 | 0548 | 0548 | 0548 | 0548 | 0548 | 0563 | 0548 | 0563 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 | 0566 
                                                                                                            0530 ;
                                                                                                            9549 ;
                                                                                                                                                                                                                                                                                                         jsave z.p.
jset current array ptr.
```

```
1590 ;k=n
              1600
7D25- A5 12
                              lda *n
                              sta *k
              1610
7D27- 85 0B
                               lda *n+1
7D29- A5 13
              1620
7D2B- 85 0C
              1630
                              sta *k+1
               1640 ; if l⇔1 soto l=1-1
                              lda * l+1
               1650 main
7D2D- A5 0E
               1660
                              bed ndec
7D2F- F0 03
                              jmp decl
7D31- 4C B2 7D 1670 dec2
                              lda *1
              1680 ndec
7D34- A5 0D
                              CMP #1
7D36- C9 01
              1690
7D38- DØ F7
              1700
                              bne dec2
               1710 ;r1=k
                              lda *k
                                                   jaet k
              1720
7D3A- A5 0B
                              sta *r3+1
7D3C- 85 1B
              1730
                              lda *k+1
7D3E- A5 0C
              1740
                              sta *r3+2
7D40- 85 1C
               1750
                              jsr conv
                                                   Jele. # to ptr. addr.
              1760
7D42- 20 B3 7F
            . 1770 js=v(k)
                                                   jr(1) to s
                               ldy #0
              1780
7D45- A0 00
                              lda (r1+1),9
              1790
7D47- B1 15
                              sta *s
7D49-85 20
              1800
              1810
                              iny
7D4B- C8
7D4C- B1 15
7D4E- 85 21
            . 1820
                              lda (r1+1),9
             1830
                              sta *s+1
              1840
7D50- C8
                               iny
              1850
                              lda (r1+1),9
7D51- B1 15
7D53- 85 22
                              sta *s+2
              1860
               1870 ju(k)=u(1)
                                                  ;r(2)=v1+3
                               lda ∗v1
7D55- A5 23
               1880
              1890
                               c lo
7D57- 18
              1900
7D58- 69 03
                               adc #$03
                              sta *r2+1
              1910
7D5A- 85 18
                              lda *v1+1
7D5C- A5 24
              1920
                              adc #$00
7D5E- 69 00
              1930
                              sta *r2+2
lda (r2+1),y
              1940
7D60-85 19
              1950
                                                  ;(n(1))=(n(2))
7D62- B1 18
7064- 91 15
              1960
                              sta (r1+1),y
                              dea
               1970
7D66-88
                            - lda (r2+1),9
7D67- B1 18
               1980
                              sta (r1+1),y
7D69- 91 15
               1990
                               deu
               2000
7D6B-88
                               lda (r2+1),y
7D6C- B1 18
               2010
               2020
                               sta (r1+1),9
7D6E- 91 15
               2030 ;k=k-1
                               sec
7D70- 38
               2040
                               lda ∗k
7D71- A5 0B
               2050
                              sbc #1
                                                  ;subtract with borrow
               2060
7D73- E9 01
                               sta *k
              2070
7D75- 85 0B
                              lda *k+1
              2080
7D77- A5 0C
               2090
                               sbc #0
7D79- E9 00
                              sta *k+1
7D7B- 85 0C
               2100
               2110 ;if kOl soto jeal
                              cmp #0
7D7D- C9 00
                2120
7D7F- D0 57
               2130
                               bne jeal
               2140
                               lda ∗k
7D81- A5 0B
               2150
                              bne jeal
7D83- DØ 53
               2160 jr1=i
```

7D85- A5 07 7D87- 85 1B 7D89- A5 08 7D8B- 85 1C 7D8D- 20 B3 7F	2170 2180 2190 2200 2210	lda *i sta *r3+1 lda *i+1 sta *r3+2 jsr conv	jconverted i to r(1)
7D90- A5 20 7D92- A0 00 7D94- 91 15 7D96- C8 7D97- A5 21 7D99- 91 15	2220 ;v(i)=s 2230 2240 2250 2260 2270 2280	lda *s ldy #0 sta (r1+1),y iny lda *s+1 sta (r1+1),y	)s to (r(1))
7D9B- C8 7D9C- A5 22 7D9E- 91 15 7DA0- A2 24 7DA2- BD DB 7F	2290 2300 2310 2320 ;exchanse 2330 sps 2340 slop	ldx #locs	;flip z.p. locations
7DH2- 6D DB 7F 7DA5- 48 7DA6- B5 00 7DA8- 9D DB 7F 7DAB- 68 7DAC- 95 00	2340 5 100 2350 2360 2370 2380 2390	lda opg,x pha lda *0,x sta opg,x pla sta *0,x	
7DAE- CA 7DAF- 10 F1 7DB1- 60 7DB2- 38	2400 2410 2420 2430 ; l= l-1 2440 dec l	dex bpl slop rts sec	;\$7f max ;end or return
7DB3- A5 0D 7DB5- E9 01 7DB7- 85 0D 7DB9- A5 0E 7DBB- E9 00 7DBD- 85 0E	2450 2460 2470 2480 2490 2500	lda *l sbc #1 sta *l lda *l+1 sbc #0 sta *l+1	;-1
7DBF- 85 10 7DC1- A5 0D 7DC3- 85 1B 7DC5- 20 B3 7F	2510 ;r1=l 2520 2530 2540 2550 2560 ;s≈v(l)	sta *r3+2 lda *l sta *r3+1 jsr conv	;conv. l to r(1)
7DC8- A0 00 7DCA- B1 15 7DCC- 85 20 7DCE- C8 7DCF- B1 15 7DD1- 85 21	2570 2580 2590 2600 2610 2620	ldy #0 lda (r1+1),y sta *s iny lda (r1+1),y sta *s+1	;(r(1)) to s
7DD3- C8 7DD4- B1 15 7DD6- 85 22 7DD8- A5 0D	2630 2640 2650 2660 ;j=l 2670 jeql	iny lda (r1+1),y sta *s+2 lda *l	
7DDA- 85 09 7DDC- A5 0E 7DDE- 85 0A	2670 Jeqt 2680 2690 2700 2710 ;i=j	sta *j lda *l+1 sta *j+1	
7DE0- A5 09 7DE2- 85 07 7DE4- A5 0A	2720 ieqj 2730 2740	lda *j sta *i lda *j+1	

7DE6- 85 08	2750 sta *i+1 2760 ;j=j+j	
7DE8- 18 7DE9- 26 09 7DEB- 26 0A	2770 clc 2780 rol *j 2790 rol *j+1	;double j
7DED- A5 0A 7DEF- C5 0C 7DF1- F0 05 7DF3- 90 0D	2800 ;compare j & k 2810   Ida *j+1 2820   cmp *k+1 2830   bea hea 2840   bcc j <k 2850 toj&gt; jmp j&gt;k</k 	;hi first
7DF5- 4C 90 7E 7DF8- A5 09 7DFA- C5 0B 7DFC- 90 04 7DFE- F0 5E 7E00- B0 F3	2850 toj) jmp j)k 2860 heq lda *j 2870 cmp *k 2880 bcc j(k 2890 beq jeqk 2900 bcs toj)	;if hi=, then ck. lo
LEGG- DO LO	2910 /if j <k r1="j&lt;/td" then=""><td></td></k>	
7E02- A5 09	2920 jCk lda *j	;; to r(1)
7E04- 85 1B	2930 sta *r3+1	
7E06- A5 0A	2940 lda *j+1	
7E08- 85 10	2950 sta *r3+2 2960 jsr conv	
7E0A- 20 B3 7F	2960 jsr conv 2970 jr2=v(j)	
7E0D- A0 00	2980 ldy #0	)(r(1)) to r(2)
7E0F- B1 15	2990 lda (r1+1),9	
7E11- 85 17	3000 sta *r2	
7E13- C8	3010 in9	
7E14- B1 15	3020 lda (r1+1),9 3030 sta *r2+1	
7E16- 85 18 7E18- C8	3030 sta *r2+1 3040 in9	
7E19- B1 15	3050 lda (r1+1),9	
7E1B- 85 19	3060 sta *r2+2	
	3070 ;r1=v(j+1)	
7E1D- 18	3080 clc	
7E1E- A5 15	3090 lda *ri+1	
7E20- 69 03	3100 adc #3 3110 sta *r1+1	;3 by. betw. ptrs. ;up (r(1)) by 1 ele.
7E22- 85 15 7E24- A5 16	3110 sta *r1+1 3120 lda *r1+2	Jun VIVIN DE I ete.
7E26- 69 00	3130 adc #0	
7E28- 85 16	3140 sta *r1+2	
	3150 ;compare v(j+1) & v(j)	7 757 7 879
7E2A- A0 02	3160 ldy #2	joopy to r(3) & r(4)
7E2C- B1 15	3170 lda (r1+1).9 3180 sta *r3+2	;v(j+1)
7E2E- 85 1C 7E30- 88	3180 sta *r3+2 3190 de9	
7E31- B1 15	3200 lda (r1+1),9	
7E33- 85 1B	3210 sta *r3+1	
7E35- 88	3220 dea	
7E36-B1_15	3230 lda (r1+1),9	
7E38- 85 1A	3240 sta *r3 3250 lda *r2+2	jv(3)
7E3A- A5 19 7E3C- 85 1F	3250 lda *r2+2 3260 sta *r4+2	3 7 3 7 .
7E3E- A5 18	3270 lda *r2+1	
7E40- 85 1E	3280 sta *r4+1	
7E42- A5 17	3290 lda *r2	
7E44- 85 1D	3300 sta *r4	
7E46- 20 DF 7E	3310 jsr cmpr 3320 ;if v(j)>=v(j+1) goto je	jcompare actual \$ data:
	2058 111 A/21\2-0/241\ 2000 26	THE STATE OF THE S

7E49- A5 14	3330	lda *r1	;deciding char.s
7E4B- C5 17 7E4D- 90 0F 7E4F- F0 0D	3340 3350 3360	om≈ *r2 boo jeak bea jeak	;r(2)>=r(1)
7E51- 18 7E52- A5 09 7E54- 69 01 7E56- 85 09 7E58- A5 0A 7E5A- 69 00 7E5C- 85 0A	3370 ;j=j+1 3380 3390 3400 3410 3420 3430 3440	clc lda *j adc #1 sta *j lda *j+1 adc #0 sta *j+1	;+1
7E5E- A5 09 7E60- 85 1B 7E62- A5 0A 7E64- 85 1C 7E66- 20 B3 7F	3450 ;r1=j 3460 jeqk 3470 3480 3490 3500 3510 ;compare v	lda *j sta *r3+1 lda *j+1 sta *r3+2 jsr conv	;j =ed k ;conv. j to r(1)
7E69- A0 02 7E6B- B1 15	3520 3530	(J/ & S   ld= #2   ld= (r1+1),=	)(r(1)) =ed v(j)
7E6D- 85 10 7E6F- 88	3540 3550	sta *r3+2 deu	joopy for s∕r
7E70- B1 15 7E72- 85 1B 7E74- 88 7E75- B1 15	3560 3570 3580 3580	lda (r1+1),9 sta *r3+1 de9 lda (r1+1),9	10(3)
7E77- 85 1A 7E79- A5 22 7E7B- 85 1F 7E7D- A5 21 7E7F- 85 1E 7E81- A5 20 7E83- 85 1D	3600 3610 3620 3630 3640 3650 3660	sta *r3 lda *s+2 sta *r4+2 lda *s+1 sta *r4+1 lda *s sta *r4	Vs.
	3670 3680 jif s(v(j)	jsr omer	;compare ≸s
7E88- A5 14 7E8A- C5 17 7E8C- F0 02 7E8E- B0 1E	3690 3700 3710 3720	lda *r1 omp *r2 bed j)k bos s(vj	<pre>/results here /r(3)/s in r(2) /if= /if r(1)&gt;r(2)</pre>
7E90- A5 07 7E92- 85 1B 7E94- A5 08	3730 /r1=i 3740 j>k 3750 3760	lda *i sta *r3+1 lda *1+1	<pre>/v(j)'s(=s's /conv. i to r(1)</pre>
7E96- 85 10 7E98- 20 B3 7F	3770 3780 3790 /v(i)=s	sta *r3+2 jsr conv	, sonv. 1 to r(1)
7E9B- A0 00 7E9D- A5 20 7E9F- 91 15 7EA1- C8 7EA2- A5 21 7EA4- 91 15	3890 3810 3820 3830 3840 3850	ldy #0 lda #s sta (r1+1),y iny lda #s+1 sta (r1+1),y	;s to (r(1))
7EA6- C8 7EA7- A5 22 7EA9- 91 15	3860 3870 3880	ine   lda	
7EAB- 4C 2D 7D	3890 3900 /r2=1	jmp main	ito top of main loop

7EAE- A5 07	3910 s(v)	lda #1	;s/s(v(j)/s
7EB0- 85 1B	3920	sta *r3+1	72 237377 2
		lda #1+1	Joonv. i to r(2)
7EB2- A5 08	3930		) CONV. 1 (0 1 \Z/
7EB4- 85 10	3940	sta *∩3+2	
7EB6- 20 B3 7F	3950	JEC CONV	5-3-179 DWY
7EB9- A5 15	3960	lda ≭r1+1	imove to r(2)
7EBB- 85 18	3970	sta *r2+1	
7EBD- A5 16	3980	lda *r1+2	
7EBF- 85 19	3990	sta *r2+2	
	4000 ;r1=j	and a transfer of the state of	
7EC1- A5 09	4010	lda <b>∗</b> j	Joonv. j to r(1)
7EC3- 85 1B	4020	sta *r3+1	) CONV. 5 (0 1 (1)
	4030		
7EC5- A5 ØA		lda *j+1	
7EC7- 85 1C	4040	șta *r3+2	
7EC9- 20 B3 7F		jsr conv	
	4060 ;v(i)=v(j)		
7ECC- A0 00	4070	ldy #0	jj's indirect to i's
7ECE- B1 15	4080	lda (r1+1),y	
7ED0- 91 18	4090	sta (r2+1),y	
7ED2- C8	4100	iny	
7ED3- B1 15	4110	lda (r1+1),9	
7ED5 - 91 18	4120	sta (r2+1),y	
7ED7- C8	4130	iny	
7ED8- B1 15	4140	lda (r1+1),9	
7EDA- 91 18	4150	sta (r2+1),y	
7EDC- 4C E0 7D	4160	jmo ieaj	<pre>/back to middle</pre>
	4170 jomph \$s s	dr.	
7EDF- A0 00	4180 cmer	ldy #0	
7EE1- 84 0F	4190	sty * In	
7EE3- A6 06	4200	ldx *flds	
7EE5- DØ ØB	4210	bne notz	
7EE7- A5 1D	4220		
		lda *r4	;sort on field#0
7EE9- 85 10	4230	sta *ln1	AN ANNUAL DESIGNATION OF THE PARTY AND AN ANALYSIS OF THE PARTY AND AN
7EEB- A5 1A	4240	lda *r3	;1st. var. in r(3)
7EED- 85 11	4250	sta ∗ln2	)2nd. in r(4)
7EEF- 18	4260	e le	
and have been been been been been been been be		the state of the s	
7EF0- 90 70	4270	bcc fsh	;find shorter \$
7EF0- 90 70 7EF2- A5 04		1977	
7EF2- A5 04	4270 4280 notz	bee fsh lda *dlim	ifield delimiter
7EF2- A5 04 7EF4- D1 1E	4270 4280 notz 4290 cont	boo fsh lda *dlim omp (r4+1),y	;field delimiter ;cmp flag to \$ char.
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08	4270 4280 notz 4290 cont 4300	boo fsh lda *dlim cmp (r4+1),y beq fndd	ifield delimiter
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8	4270 4280 notz 4290 cont 4300 4310 cnt3	bcc fsh lda *dlim cmp (r4+1),9 beq fndd in9	;field delimiter ;cmp flas to \$ char. ;found delim.
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D	4270 4280 notz 4290 cont 4300 4310 cnt3 4320	bcc fsh lda *dlim cmp (r4+1),9 bea fndd in9 cps *r4	;field delimiter ;cmp flag to \$ char. ;found delim. ;end of \$?
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330	bcc fsh lda *dlim cmp (r4+1),9 bea fndd in9 cpg *r4 bcc cont	;field delimiter ;cmp flag to \$ char. ;found delim. ;end of \$? ;no
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7 7EFD- 4C F3 7C	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330 4340	bcc fsh lda *dlim cmp (r4+1),y bea fndd iny cpy *r4 bcc cont jmp oops	<pre>;field delimiter ;cmp flag to \$ char. ;found delim. ;end of \$? ;no ;on error</pre>
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7 7EFD- 4C F3 7C 7F00- 84 0F	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330 4340 4350 fndd	bcc fsh lda *dlim cmp (r4+1),9 bea fndd in9 cpg *r4 bcc cont	;field delimiter ;cmp flag to \$ char. ;found delim. ;end of \$? ;no
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7 7EFD- 4C F3 7C	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330 4340	bcc fsh lda *dlim cmp (r4+1),y bea fndd iny cpy *r4 bcc cont jmp oops	<pre>;field delimiter ;cmp flag to \$ char. ;found delim. ;end of \$? ;no ;on error</pre>
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7 7EFD- 4C F3 7C 7F00- 84 0F	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330 4340 4350 fndd	bcc fsh lda *dlim cmp (r4+1),y beq fndd iny cpy *r4 bcc cont jmp cops sty *ln dex	;field delimiter ;cmp flag to \$ char. ;found delim. ;end of \$? ;no ;on error ;mark current offset
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7 7EFD- 4C F3 7C 7F00- 84 0F 7F02- CA 7F03- F0 02	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330 4340 4350 fndd 4360	bcc fsh lda *dlim cmp (r4+1),y beq fndd iny cpy *r4 bcc cont jmp cops sty *ln dex beq sfld	;field delimiter ;cmp flag to \$ char. ;found delim. ;end of \$? ;no ;on error ;mark current offset ;sort field
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7 7EFD- 4C F3 7C 7F00- 84 0F 7F02- CA 7F03- F0 02 7F05- B0 F1	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330 4340 4350 fndd 4360 4370	bcc fsh lda *dlim cmp (r4+1),y bea fndd iny cpy *r4 bcc cont jmp cops sty *ln dex bea sfld bcs cnt3	<pre>;field delimiter ;cmp flag to \$ char. ;found delim. ;end of \$? ;no ;on error ;mark current offset ;sort field ;count on</pre>
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7 7EFD- 4C F3 7C 7F00- 84 0F 7F02- CA 7F03- F0 02 7F05- B0 F1 7F07- C8	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330 4350 fndd 4360 4370 4380 4390 sfld	bcc fsh lda *dlim cmp (r4+1),9 bea fndd in9 cp9 *r4 bcc cont jmp cops sty *ln dex bea sfld bcs cnt3 in9	;field delimiter ;cmp flam to \$ char. ;found delim. ;end of \$? ;no ;on error ;mark current offset ;sort field ;count on ;@ sort field
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7 7EFD- 4C F3 7C 7F00- 84 0F 7F02- CA 7F03- F0 02 7F05- B0 F1 7F07- C8 7F08- D1 1E	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330 4350 fndd 4350 fndd 4370 4380 4390 sfld 4400	bcc fsh lda *dlim cmp (r4+1),y bea fndd iny cpy *r4 bcc cont jmp cops sty *ln dex bea sfld bcs cnt3 iny cmp (r4+1),y	;field delimiter ;cmp flas to \$ char. ;found delim. ;end of \$? ;no ;on error ;mark current offset ;sort field ;count on ;@ sort field ;next char.
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7 7EFD- 4C F3 7C 7F00- 84 0F 7F02- CA 7F03- F0 02 7F05- B0 F1 7F07- C8 7F08- D1 1E 7F08- F0 05	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330 4340 4350 fndd 4360 4370 4380 4390 sfld 4400	bcc fsh lda *dlim cmp (r4+1),9 bea fndd in9 cps *r4 bcc cont jmp cops sty *ln dex bea sfld bcs cnt3 in9 cmp (r4+1),9 bea fnef	;field delimiter ;cmp flam to \$ char. ;found delim. ;end of \$? ;no ;on error ;mark current offset ;sort field ;count on ;@ sort field ;next char. ;field beyond sort
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7 7EFD- 4C F3 7C 7F00- 84 0F 7F02- CA 7F03- F0 02 7F05- B0 F1 7F07- C8 7F08- D1 1E 7F08- F0 05 7F0C- C4 1D	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330 4350 fndd 4350 fndd 4360 4370 4380 4390 sfld 4400 4420	bcc fsh lda *dlim cmp (r4+1),9 bea fndd in9 cm9 *r4 bcc cont jmp cops sty *ln dex bea sfld bcs cnt3 in9 cmp (r4+1),9 bea fnef cp9 *r4	;field delimiter ;cmp flas to \$ char. ;found delim. ;end of \$? ;no ;on error ;mark current offset ;sort field ;count on ;@ sort field ;next char.
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7 7EFD- 4C F3 7C 7F00- 84 0F 7F02- CA 7F03- F0 02 7F05- B0 F1 7F07- C8 7F08- D1 1E 7F08- F0 05 7F06- C4 1D 7F06- 90 F7	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330 4340 4350 fndd 4360 4370 4380 4390 sfld 4400 4420 4430	bcc fsh lda *dlim cmp (r4+1),9 bea fndd in9 cm9 *r4 bcc cont jmp cops sty *ln dex bea sfld bcs cnt3 in9 cmp (r4+1),9 bea fnef cm9 *r4 bcc sfld	;field delimiter ;cmp flam to \$ char. ;found delim. ;end of \$? ;no ;on error ;mark current offset ;sort field ;count on ;@ sort field ;next char. ;field beyond sort
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7 7EFD- 4C F3 7C 7F00- 84 0F 7F02- CA 7F03- F0 02 7F05- B0 F1 7F07- C8 7F08- D1 1E 7F08- F0 05 7F0C- C4 1D 7F0E- 90 F7 7F10- C8	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330 4340 4350 fndd 4360 4370 4380 4490 sfld 4400 4420 4430 4440	bcc fsh lda *dlim cmp (r4+1),9 bea fndd in9 cm9 *r4 bcc cont jmp cops sty *ln dex bea sfld bcs cnt3 in9 cmp (r4+1),9 bea fnef cp9 *r4 bcc sfld in9	;field delimiter ;cmp flam to \$ char. ;found delim. ;end of \$? ;no ;on error ;mark current offset ;sort field ;count on ;@ sort field ;next char. ;field beyond sort ;end of \$?
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7 7EFD- 4C F3 7C 7F00- 84 0F 7F02- CA 7F03- F0 02 7F05- B0 F1 7F07- C8 7F08- D1 1E 7F08- D1 1E 7F08- F0 05 7F0C- C4 1D 7F0E- 90 F7 7F10- C8 7F11- 88	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330 4340 4350 fndd 4360 4370 4380 4490 sfld 4400 4420 4430 4430 4450 fnef	bcc fsh lda *dlim cmp (r4+1),9 bea fndd in9 cm9 *r4 bcc cont jmp cops sty *ln dex bea sfld bcs cnt3 in9 cmp (r4+1),9 bea fnef cpy *r4 bcc sfld in9 dey	;field delimiter ;cmp flam to \$ char. ;found delim. ;end of \$? ;no ;on error ;mark current offset ;sort field ;count on ;@ sort field ;next char. ;field beyond sort
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7 7EFD- 4C F3 7C 7F00- 84 0F 7F02- CA 7F03- F0 02 7F05- B0 F1 7F07- C8 7F08- D1 1E 7F08- D1 1E 7F08- F0 05 7F06- C4 1D 7F06- C8 7F10- C8 7F10- C8 7F10- C8 7F11- 88 7F12- 98	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330 4340 4350 fndd 4360 4370 4380 4490 4420 4430 4450 fnef 4460	bcc fsh lda *dlim cmp (r4+1),9 bea fndd in9 cm9 *r4 bcc cont jmp cops sty *ln dex bea sfld bcs cnt3 in9 cmp (r4+1),9 bea fnef cp9 *r4 bcc sfld in9	;field delimiter ;cmp flam to \$ char. ;found delim. ;end of \$? ;no ;on error ;mark current offset ;sort field ;count on ;@ sort field ;next char. ;field beyond sort ;end of \$?
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7 7EFD- 4C F3 7C 7F00- 84 0F 7F02- CA 7F03- F0 02 7F05- B0 F1 7F07- C8 7F08- D1 1E 7F08- D1 1E 7F08- F0 05 7F0C- C4 1D 7F0E- 90 F7 7F10- C8 7F11- 88	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330 4340 4350 fndd 4360 4370 4380 4490 sfld 4400 4420 4430 4430 4450 fnef	bcc fsh lda *dlim cmp (r4+1),9 bea fndd in9 cm9 *r4 bcc cont jmp cops sty *ln dex bea sfld bcs cnt3 in9 cmp (r4+1),9 bea fnef cpy *r4 bcc sfld in9 dey	;field delimiter ;cmp flam to \$ char. ;found delim. ;end of \$? ;no ;on error ;mark current offset ;sort field ;count on ;@ sort field ;next char. ;field beyond sort ;end of \$?
7EF2- A5 04 7EF4- D1 1E 7EF6- F0 08 7EF8- C8 7EF9- C4 1D 7EFB- 90 F7 7EFD- 4C F3 7C 7F00- 84 0F 7F02- CA 7F03- F0 02 7F05- B0 F1 7F07- C8 7F08- D1 1E 7F08- D1 1E 7F08- F0 05 7F06- C4 1D 7F06- C8 7F10- C8 7F10- C8 7F10- C8 7F11- 88 7F12- 98	4270 4280 notz 4290 cont 4300 4310 cnt3 4320 4330 4340 4350 fndd 4360 4370 4380 4490 4420 4430 4450 fnef 4460	bcc fsh lda *dlim cmp (r4+1),y bea fndd iny cpy *r4 bcc cont jmp cops sty *ln dex bea sfld bcs cnt3 iny cmp (r4+1),y bea fnef cpy *r4 bcc sfld iny dey tya	;field delimiter ;cmp flam to \$ char. ;found delim. ;end of \$? ;no ;on error ;mark current offset ;sort field ;count on ;@ sort field ;next char. ;field beyond sort ;end of \$?

7F16- 85 10 7F18- E6 0F 7F18- 85 0F 7F1C- 18 7F1D- 65 1E 7F1F- 85 1E 7F21- A5 1F 7F23- 69 00 7F25- 85 1F 7F27- A0 00 7F29- 84 0F 7F2B- A6 06 7F2D- A5 04	4490 4500 4510 4520 4530 4540 4550 4560 4570 4580 4600 4610	sta * ln1 inc * ln lda * ln clc adc *r4+1 sta *r4+2 lda *r4+2 adc #\$00 sta *r4+2 ld9 #0 sty * ln ldx *f lds lda *d lim	; len sort field ;skip delim. ;offset from start ;start of sort field ;r(4)'s done ;now other \$ ;only rem.s differ
7F2F- D1 1B 7F31- F0 08 7F33- C8 7F34- C4 1A 7F36- 90 F7 7F38- 4C F3 7C 7F3B- 84 0F 7F3D- CA 7F3E- F0 02 7F40- B0 F1	4620 cnt2 4630 4640 cnt4 4650 4660 4670 4680 fnd2 4690 4700	cmp (r3+1),y beq fnd2 iny cpy *r3 bcc cnt2 jmp oops sty *ln dex beq sfd2 bcs cnt4	
7F42- C8 7F43- D1 1B 7F45- F0 05 7F47- C4 1A 7F49- 90 F7 7F4B- C8 7F4C- 88 7F4C- 88 7F4E- 38 7F4E- 38 7F4F- E5 0F 7F51- 85 11 7F53- E6 0F	4720 sfd2 4730 4740 4750 4760 4770 4780 fne2 4790 4800 4810 4830	iny cmp (r3+1),y beq fne2 cpy *r3 bcc sfd2 iny dey tya sec sbc *ln sta *ln2 inc *ln	
7F55- A5 ØF 7F57- 18 7F58- 65 1B 7F5A- 85 1B 7F5C- A5 1C 7F5E- 69 ØØ 7F6Ø- 85 1C 7F62- A5 1Ø 7F64- C5 11 7F66- FØ Ø8 7F68- BØ ØC	4840 4850 4860 4870 4880 4890 4900 4910 fsh 4920 4930	lda * in c lc adc *r3+1 sta *r3+1 lda *r3+2 adc #\$00 sta *r3+2 lda * in1 cmp * ln2 bea ea bos two<	;found shorter \$ ;r(4)'s in ln1 ;2nd. shorter?
7F6A- 85 0F 7F6C- A2 01 7F6E- D0 0C 7F70- 85 0F 7F72- A2 00 7F74- F0 06 7F76- A5 11 7F78- 85 0F 7F7A- A2 02	4950 ; which 4960 4970 4980 4990 ea 5000 5010 5020 two< 5030 5040 5060 beas	sta *ln ldx #1 bne begs sta *ln ldx #0 beg begs lda *ln2 sta *ln ldx #2	;store least ;1st. shorter ;jump ;same ;jump ;2nd. shorter ;ck. if \$ is null

```
7F7E- F0 0D 5070 bea null
7F80- A0 00 5080 ldy #8
                       5090 jompr next char.
                  5100 nex lda (r3+1),9
5110 cmp (r4+1),9
5120 bne dif
5130 in9
 7F82- B1 1B
                                               omp (r4+1),9
7F84- D1 1E
                                                                              Johan.s differ?
                                               bne dif
ing
7F86- DØ 24
 7F88- C8
                                                                                 ino
5140 ;beyond last char.?
7F89- C4 0F 5150 cpy *ln
7F8B- 90 F5 5160 bcc nex
                                                                                √Clen
                        5170 ;if so,which $ is longer?
                      5180 null crx #1
5190 bea one<<
5200 brl two<<
7F8D- E0 01
7F8F- F0 09
                                                                                /1st.?
                      5200
 7F91- 10 10
                                                                                 ino, 2nd.?
                                                                  ino
il rts below selected
ifrom d social
                        5210 /same
7F93- A9 00 5220
7F95- 85 14 5230
7F97- 85 17 5240
7F99- 60 5250
                                                lda #0
                                                sta *r1
                                                sta *r2
                                                rts
                  5260 ;one is (
5260 ;one is (
5270 one(( lda (r4+1),y
5280 sta *r1
5290 lda #0
7F9A- B1 1E
7F9C- 85 14
7F9E- A9 00
                      5300
5310
sta *n2
 7FA0--85 17
                                               lda (r3+1),9
5430 ; conversion from # to address w/i pointer array s/r
7FB3- A5 1B 5440 conv lda *r3+1
7FB5- 85 1E 5450 sta *r4+1
7FB7- A5 1C 5460 lda *r3+2
7FB9- 85 1F 5470 sta *r4+2
7FB8- 18 5480 clc
7FBC- 26 1B 5490 rol *r3+1
7FBE- 26 1C 5500 rol *r3+2
7FC0- A5 1B 5510 lda *r3+1
7FC2- 18 5520 clc
7FC3- 65 1E 5530 adc *r4+1
7FC5- 85 1B 5540 sta *r3+1
7FC7- A5 1C 5550 adc *r4+2
7FC9- 65 1F 5560 adc *r4+2
7FCB- 85 1C 5570 sta *r3+2
7FCB- 85 1C 5570 sta *r3+2
7FCB- 85 1B 5580 lda *r3+1
7FCF- 18 5590 clc
7FD0- 65 23 5600 adc *v1
                                                rol #r3+1
                                                                                 ;double it
                                           rol *r3+2
                                                                      ;+t1=t*3
                                                 lda *r3+1
                                                                                 distance from ar. start
                                           0:0
ado *v1
sta *r1+1
lda *r3+2
ado *v1+1
sta *r1+2
7FD0- 65 23 5600

7FD2- 85 15 5610

7FD4- A5 10 5620

7FD6- 65 24 5630

7FD8- 85 16 5640

7FD8- 60 5650
                                                                      iresult in r(1)
7FDA- 60
7FDB-
                                                rts
                                            .ds locs+1 /save z.m. here
                        5660 cpa
5670
                                                  .en
```

#### Label File

aras	=0020
	=7F7C
	=7F2F
	=7EF4
deh =	
	=0003
	=002E
ea =7	
	=0005
	=0006
	=7F4C
	=7CFB
hok =	
j =00	
	=7E5E
1 =00	
in2 =	
	=7D2D
	=7D34
	=7EF2
	=7CF3
r3 =0	
	=7EAE
	=7F42
S09 =	
	=7FA3

arer =0080
ckna =708A
ont3 =7EF8
conv =7FB3
dec2 =7D31
dif =7FAC
efnd =7CD6
errp =0357
flo2 =7CAC
fnd2 =7F3B
fnef =7F11
hea =7DF8
i =0007
j(k =7 <b>E</b> 02
jeal =7DD8
ln =000F
locs =0024
n =0012
nex =7F82
null =7F8D
r1 = 9914
r4 =001D
sart =7061
sfld =7F07
toj> =7DF5
v1 =0023
A1 -0050

arnm =0000
omer =7EDF
ont4 =7F33
cma =7FDB
decl =7DB2
dlim =0004
ehfd =7CCE
fana =7CEB
floh =7CB6
fndd =7F00
fsh =7F62
him =0034
ieqj =7DE0
j>k =7E90
k =000B
lni =0010
lok =7098
nare =7CDC
nmfl =0002
one(< =7F9A
r2 = 0017
s =0020
sav =7073
slop =7DA2
two< =7F76
wrnm =7CA0



Get rid of those bugs the way the experts do. Our TEMA program allows you to easily locate errors in any machine language program.

ASCII or hex strings can be located, registers and memory locations displayed, plusmany other features that are invaluable for debugging.

Available now on cassette or disk. 48K Apple required.

Cassette-\$19.95

Disk-\$24.95

#### APPLE ONLY

Also Available	le Cassette	Disk
One-Arm Bandit (32K)	9.95	14.95
Card Shark (INT)	7.95	9.95
High Roller (Applesoft)	7.95	9.95
Hi-Res Sub Game (32K)	14.95	19.95
Adult Games Pack	7.95	9.95
Trend Line Analysis (48K, Applesoft & ROM)	9.95	14.95

#### **CUSTOM PROGRAMMING**

We specialize in custom software. Modifications to existing programs or completely specialized to suit your needs. Write for details.

#### Decision Systems

//0000,8000,8000



Presenting the Other Side of the Apple II\*

ISAM-DS

An integrated set of routines for the creation and manipulation of indexed files. Retrieve records sequentially by key value or partial key value. Files never have to be reorganized.

Requires: Disk, Applesoft (32K ROM or 48K RAM)

PBASIC-DS A sophisticated preprocessor for structured BASIC. Gain the power of PASCAL-like logic structures at a fraction of the cost. Use all regular BASIC statements plus 14 commands and 9 new statements/structures (WHILE, UNTIL, CASE, etc.) Works with INTEGER or APPLESOFT programs

Requires: Disk, Applesoft (32K ROM or 48K RAM) \$35

UTIL-DS

An Applesoft utility package that includes improved error interrupt handling (return to the statement following the one in error), a routine that selectively clears array variables (for reDIM or chaining), an interface routine that provides a 'GOSUB' facility from machine language to Applesoft and an advanced formatting routine for printing numeric values. Works with negative values; inserts commas in value, etc. Also contains a loader to put the routines into RAM with your program. Requires: Disk, Applesoft (ROM of RAM)

\$35

(Texas residents add 5% tax)

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

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

# FREE! up to \$170. in merchandise with purchase of PET-CBM item!!!

PET 16K Large Keyboard \$ 995 \$130
rei 32k large keyodard \$1295 \$170
PET 8K Large Keyboard (New) \$ 795 \$100
PET 2040 Dual Disk (343K) \$1295 \$170
PET 2023 Printer (pres feed) \$ 695 \$ 70 /
PET 2022 Printer (trac feed) \$ 795 \$100 1
KIM-1 \$159 (Add \$30 for Power Supply) SYM-1 \$ 209.00
AXIOM EX-801 Printer-PET \$ 477.00
2114 L 450 ns 5.35 24/4.95 100/4.45
2716 EPROM (5 Volt)
6550 RAM (for 8K Pet)
PET 4 Voice Music System (KL-4M) 34.50
All Books and Software
Leedex Video 100 12" Monitor 119.00

#### ATARI — INTRODUCTORY SPECIAL

ATARI 400, Atari 800, and all Atari Modules 20% OFF.

Programmers Toolkit - PET ROM Utilities ...... 44.90 PET Word Processor - Machine Language ...... 24.00



3M "Scotch" 8" Disks ... 3M "Scotch" 5" Disks Verbatim 5" Disks ... Disk Storage Pages . . .

SALE 10/31.00 10/31.50 10/26.50 10/ 3.95

Cassettes (all tapes guaranteed) Premium quality, high output low noise in 5 screw housing with labels: AGFA PE 611 noise in 5 screw housing with labels: C-10 10/5.95 50/25.00 C-30 10/7.00 50/30.00 100/48.00

Add \$1 per order for UPS shipping. Ask for 6502, TRS-80, and S-100 Product List.

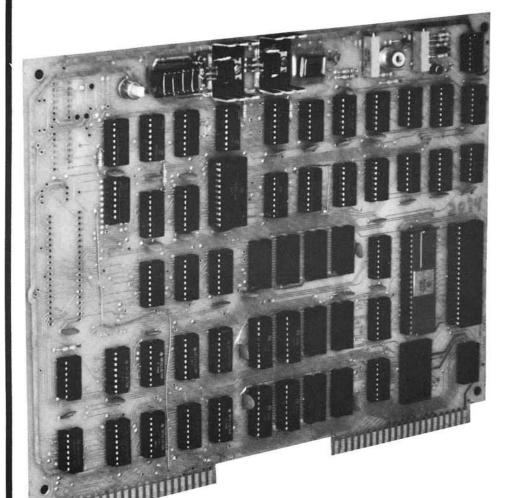
115 E. Stump Road B Computers Montgomeryville, PA 18936 (215) 699-8386

# IT'S EASY TO ADD VIDEO TO YOUR SYSTEM

# VIDEO PLUS

VIDEO PLUS is the most versatile expansion board offered for the ASK family of microcomputers. It has many important features:

- Programmable Display Format with up to 80 characters by 24 lines.
- UPPER and lower case ASCII characters in ROM.
- Optional Programmable Character Generator with up to 128 additional user defined characters.
- Onboard Display RAM adds up to 4K bytes of RAM to your system as display memory.
- Provision for 2K EPROM memory for the ASK VIDEO
   PLUS Software or other support programs.
- Keyboard Interface supports an ASCII keyboard.
- Smart Video Controller supports hardware scrolling, programmable cursor, light pen, programmable video format, ....
   ASK VIDEO PLUS Software provides a complete control package for the AIM, SYM, and KIM. Works
- ASK VIDEO PLUS Software provides a complete control package for the AIM, SYM, and KIM. Works with the AIM/SYM Monitor, Editors, BASIC, and any program using the I/O vectors.
- Single Voltage only +5V @ 1.5 Amps.



WE ARE LOOKING FOR A FEW GOOD COMPUTERISTS:

6502 PROGRAMMERS
TECHNICAL WRITERS
JUNIOR ENGINEERS
SENIOR TECHNICIANS

For details please call Randy at 617/256-3649 or send your resume to:

THE COMPUTERIST P.O. Box 3

So. Chelmsford, MA 01824

#### **DATA CAPTURE 3.0**

Is DATA CAPTURE just another smart terminal program for your Apple II<sup>®</sup> or Apple II Plus<sup>®</sup>?

NO. It is a GENIUS TERMINAL PROGRAM and is designed to be used with the Micromodem II<sup>179</sup>.

Tired of watching data and programs scroll off the screen forever? Then DATA CAPTURE is the program for you.

- ANYTHING that appears on the screen of your Apple II can be captured.
   Any program or data.
- You can then save what you have captured to disk, dump it to your printer or even do simple editing with DATA CAPTURE.
- You can use DATA CAPTURE to compose text offline for later transmission to another computer. Think of the timeshare charges this will save you.
- Use DATA CAPTURE with the Dan Paymar Lower Case Adapter and you can enter lower case from the keyboard for transmission to another computer or capture both upper and lower case.
- A program is also included to convert your programs to text files for transmission using DATA CAPTURE.
- You receive two versions of the program. One is for upper case only and one for both upper and lower case use with the above adapter.

DATA CAPTURE will save you money if you are using a timesharing system because you can compose messages offline for later transmission. You can also quickly capture data for later reading, printing or editing. Requires DISK II<sup>®</sup>, APPLESOFT II<sup>®</sup>.

#### Price \$29.95

If your local dealer does not have DATA CAPTURE then order directly. We ship DATA CAPTURE within 3 working days of receipt of order and welcome your personal check. We also accept Visa and Master Charge. Add \$49.95 if you would also like to order the Dan Paymar Lower Case Adapter at the same time.

#### Ask for a catalog of our software.

- \* Apple II, Apple II Plus, Disk II and APPLESOFT II are trademarks of Apple Computer Company.
- \* Micromodem II is a trademark of D.C. Hayes Associates, Inc.

SOUTHEASTERN SOFTWARE 7270 Culpepper Drive New Orleans, LA 70126

504/246-8438 504/246-7937

# KIM Scorekeeper

Always on the lookout for new applications for the basic KIM-1, a general purpose, multi-player scorekeeper is presented. The techniques can be readily modified for use on a SYM-1 or AIM 65, and the scorekeeping function can be included as part of larger game programs.

Joel Swank 4655 SW 142nd, 186 Beaverton, OR 97005

Ever have a problem getting someone to keep score for your friendly game of Hearts? Well KIM would like to be a volunteer. KIM will keep up to nine separate scores for you to display and update from the keyboard. Each player can have from 0 to 9999 points, sufficient for most card games or other games needing a scorekeeper. Bridge fans can drop the low order zero from their scores (150 points for a grand slam??). I must credit the idea to a hardware project in October Popular Electronics by Joseph Fortuna. He used decade counters and 7-segment LED drivers to two-digit scores. A telephone dial was used to increment to counters. I immediately saw a job that KIM could do with software. Naturally with all the power of KIM available I had to improve and expand the idea.

The KIM SCOREKEEPER uses nine 2-byte memory registers to save the players' scores. Normally one of the players' scores is displayed continuously in the KIM display. The high order digit of the display is the player number, 1 to 9. The next digit is blank and the four low order digits contain that player's score. To display another player's score the PC (Player Change) key is pushed and the display goes blank. Then a number from 1 to 9 is pushed to get that player's score in the display. After a player is selected, the score can be updated. A player's score can be increased by entering the number to be added to the score and pushing the 'E' (Enter) key. Up to four digits can be entered. During entry of a number, the display shows the number being entered in the four low order digits with the two high order digits blank. Digits are shifted through the display as they are entered. If more that four digits are entered, the high order digits are shifted out and lost as in the KIM monitor.

The player's score can be decreased by pushing the 'D' (Decrease) key to set subtract mode. When the subtract mode is in effect, any number entered wil be subtracted from the player's score when the 'E' key is pushed. The high order digit of the display will show a minus sign when the number being entered is to be subtracted. Subtract mode stays in effect until the '+' key is pushed to reset the program to add mode. The '+' and 'D' keys are effective anytime except when performing the player change function. If any key except 0 to 9, '+' or 'D' is entered during the update operation the display returns to the current player. The 'C' (Clear) key may be used to zero the current player's score.

As shown by the programs, SCORE-KEEPER has two main display loops. One displays the current player and his score while waiting for a command from the keyboard. The other displays the number being entered while inputting digits from the keyboard. The code is divided into subroutines for the sake of modularity and readability. The KIM subroutine GETKEY is used for communication from the keyboard, and the HEX to 7-segment conversion table in the KIM ROM is used to generate characters. The display is driven directly by the subroutine DISSEG. DISSEG is more flexible than the KIM subroutine SCANDS since it allows individual control of each segment of the KIM display. Thus any pattern can be displayed. DISSEG reads data from memory at SEGBUF and dumps it directly to the KIM display high order digit first. This subroutine could be used in a wide variety of games for KIM.

KIM SCOREKEEPER is an example of KIM's ability to replace and improve a hardware gadget. There is nothing I like more than finding a hardware function that KIM can replace with software. Someday I will calculate the weight of the hardware that my KIM has displaced.

```
0001:
0002:
                                    KIM SCOREKEEPER
0003:
                                VERSION 1 SEPTEMBER 1979
0004:
0005:
0006:
0007:
                      SCORER ORG
                                    $0200
0008: 0200
0009:
                       ZERO PAGE STORAGE
0010:
0011:
                                           PLAYER SCORE TABLE
                      PLAYER *
                                    $0080
0012: 0200
                                           Ø-ADD ELSE SUBTRACT
                      MODE
                                    $0094
0013: 0200
                      CURPLA *
                                    $0095
                                            INDEX TO CURRENT PLAYER
0014: 0200
                                            LAST KEY ENTERED
                      CURKEY *
                                    $0096
0015: 0200
                                            REGISTER SAVE AREA
                      TEMP
                                    $0097
0016: 0200
0017: 0200
                      INDEX
                             *
                                    $0098
                                            REGISTER SAVE AREA
                      SEGBUF *
                                    $0099
                                            DISPLAY BUFFER
0018: 0200
                      NUMBUF *
                                            NUMBER INPUT BUFFER
                                    $009F
0019: 0200
0020:
0021: 0200
                      ZERO
                                    $0000
```

				<b>:</b>			w	
	* * * * *			**** TO LIFT		2	A ICE	
	*	* * *	* -	* *	× *	* * *	BUFFER KEY T	
SCORE	* +	* * × * ш ээ	* * *	* KE *	INDEX **** TE	EY ***BUFFER	ш •	Щ
	****** CURRENT	****   INDE   BYTES	******* CURRENT	* I	**************************************	* *	FEH ARD SAME	EY? CALCULATE
UPDATE	* i	* * EB *	****** F NEW CU	******* KEYBOARD FOR LAST E LESS T	2 * DF	FIRST INPUT	R KEY INT AY BUFFE KEYBOABD F READ SA	EY?
	* 4	**************************************	* * * * * * * * * * * * * * * * * * *	* * *	AVE IT AVE IT WEEN AND CURRENT		LAY E	60 C
YES,	* * * * * * * * * * * * * * * * * * * *	**************************************	* * * * * * * * * * * * * * * * * * *	**************************************	MULT E SAVE 1 ** *** NUMBER OF CURF	**************************************	SHIFT KEY INTO DISPLAY BUFFER READ KEYBOARD DON'T READ SAMI IGNOHE 'NO KEY	ENTER KEY? YES GO CAL
	**************************************	* 4 G B Z B Z B Z	******* NUMBER C	*	M * NU :	* * LL LL		
UPDATE	******	****** CURPLA ZERO PLAYER	******* GET NUN	***** GETKEY CURKEY UPPLAY CURKEY UPPLAY UPPLAY	CURPLA * * * * INPUT SCORE	TEKP ZERO NUMBUF TEMP	SHFKEY DISNUM GETKEY CURKEY UDLOOP CURKEY \$15	SØE Addum
<b>–</b> 0	**************************************	•	HTS ************************************	* 3 2	<b>*</b>	*	# 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	
USB UMP	* no	LDAIM STAZX INX STAZX	# # K	**************************************	A # B T S A T E * *	* STA * STA * STA STA STA CTA	JSB JSB CKMP BEG STA CKMPI	CMPIN
	*****	. H	***** UPPLAY	*	ASL STA BTS RTS ****	*	<b>a</b>	ОШ
	* * * * * *	CLACUR	* * * * *	*****	* * * * *	**************************************	UPLUP	
22		۵		_	production of the control of	5		
60 60		waa a		# 2000 - E E F	10	V 81 . 8 V	25 ±	
20 6E 4C 16		A6 95 895 88 95 88	<b>15</b>	28 6A 65 96 67 96 67 87 67 87 68 67 88 67	8 9 9 5 9 5 9 5 9 5 9 5 9 5 9 5 9 5 9 5	5 97 5 98 5 97 5 97	8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	9 BE
					8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	E 85 2 85 4 85 6 A5	4440F	B C9
0249 0240		0254F 0253 0253 0255	20	0259 0250 0250 0262 0262 0264	026B 026B 026D	026E 0270 0272 0274	0278 0278 0276 0281 0283 0283	Ø288 Ø28D
0082:	888898 888898 888898 888898 888898 888898	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	20000000000000000000000000000000000000	20000000000000000000000000000000000000	114. 115. 117. 118. 118. 128. 122.	125 125 125 127 128 129	1322 1332 1332 1335 1336 1336	140:
	00000000							
22		2000000	22200000	200000000	222222222	2 2 2 2 2 2 2 2	000000000	200
22	22200000		2222222		222222222 LLLLLLLLLLL	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2		9 22 27
88				× ?	222222222 LLLLLLLLLLL	2 0 0 0 0 0 0	LAYER BB	9 22 27
22	E R OR T		<i>a</i>	× ?		MODE MODE	PLAYER BB	200
20	E R OR T		D 60	SO * * * * * * * * * * * * * * * * * * *		MODE	PLAYER BB	II B
888	E R OR T		RES TO B	SO * * * * * * * * * * * * * * * * * * *	MODE BEING	MODE	CURRENT PLAYER BE CURRENT PLAYER BE CURRENT PLAYER BE	II B
8 8	E R OR T		RES TO B	SO * * * * * * * * * * * * * * * * * * *	ADD MODE	SUBTHACT MODE 8	CURRENT PLAYER BE CURRENT PLAYER BE CURRENT PLAYER BE	II B
88	HE KEYBOAHD 7-SEG TABLE TON REGISTEH SELECT PORT IS OUTPUT PORT	* * * * * * * * * * * * * * * * * * *	ALL SCORES TO 0 CURRENT KEY	COND DIGIT BLANK T PLAYER IS #1  *******  ******  *******  *******  ****	CE CELL & GET NET 81 EY?	ET SUBTHACT MODE 0	CLEAR CURRENT PLAYER BY CLEAR CURRENT PLAYER BY CHANGE CURRENT PLAYER B	THEN IGNORE IT
5 5	HE KEYBOAHD 7-SEG TABLE TON REGISTEH SELECT PORT IS OUTPUT PORT	* * * * * * * * * * * * * * * * * * *	ALL SCORES TO 0 CURRENT KEY	COND DIGIT BLANK T PLAYER IS #1  *******  ******  *******  *******  ****	CE CELL & GET NET 81 EY?	KEY?  SET SUBTHACT MODE  MKEY?	CLEAH CURRENT PLAYER BE KEY?  CHANGE CURRENT PLAYER BE CHANGE CURRENT PLAYER PLAYE	THEN IGNORE IT
5 5	READ THE KEYBOAHD HEX TC 7-SEG TABLE DIRECTION REGISTER DIGIT SELECT PORT SEGMENTS OUTPUT PORT	* * * * * * * * * * * * * * * * * * *	INIT CURRENT KEY SET MODE TO ADD	COND DIGIT BLANK T PLAYER IS #1  *******  ******  *******  *******  ****	DEBOUNCE CONE & GET NET BIT OF BOUNCE BIT OF	D KEY? NO YES SET SUBTHACT MODE	YES, CLEAH CURRENT PLAYER BY PC KEY?	NUMERIC KEY? NO. THEN IGNORE IT
555	1F6A READ THE KEYBOARD 1FE7 HEX TC 7-SEG TABLE 1741 DIRECTION REGISTER 1742 DIGIT SELECT PORT 1740 SEGMENTS OUTPUT PORT	* * * * * * * * * * * * * * * * * * *	INIT CURRENT KEY SET MODE TO ADD	COND DIGIT BLANK T PLAYER IS #1  *******  ******  *******  *******  ****	DEBOUNCE CONE & GET NET BIT OF BOUNCE BIT OF	D KEY? NO YES SET SUBTHACT MODE	YES, CLEAH CURRENT PLAYER BY PC KEY?	NUMERIC KEY? NO. THEN IGNORE IT
5 5 5	READ THE KEYBOAHD HEX TC 7-SEG TABLE DIRECTION REGISTER DIGIT SELECT PORT SEGMENTS OUTPUT PORT	* * * * * * * * * * * * * * * * * * *	ZERO INIT ALL SCORES TO 0 PLAYER \$1A CLALUP CURKEY INIT CURRENT KEY WODE SET MODE TO ADD	SEGBUF +01 SECOND DIGIT BLANK  # \$02 CURHENT PLAYER IS #1 CURPLA  **********  *********  **********  ****	### SET ADD WODE GETLUP  ### SET ADD WODE  ### GETLUP  ### SET ADD WODE  ### GETLUP	IM \$800 D KEY? NOMNUS NO MCDE YES SET SUBTHACT MODE 8 GETLUP IM \$80	OCCLE NO CLEAR CURRENT PLAYER BE CETLUP STATE NO CLEAR CURRENT PLAYER BE NO CLEAR CURRENT PLAYER BE CETLUP YES, CHANGE CURRENT PLAYER BE CETLUP	\$0A NUMERIC KEY?
	\$1F6A READ THE KEYBOAHD \$1FE7 HEX TO 7-SEG TABLE \$1741 DIRECTION REGISTER \$1742 DIGIT SELECT PORT \$1740 SEGMENTS OUTPUT PORT	* * * * * * * * * * * * * * * * * * *	ZERO INIT ALL SCORES TO 0 PLAYER \$1A CLALUP CURKEY INIT CURRENT KEY WODE SET MODE TO ADD	SEGBUF +01 SECOND DIGIT BLANK  # \$02 CURHENT PLAYER IS #1 CURPLA  **********  *********  **********  ****	### SET ADD WODE GETLUP  ### SET ADD WODE  ### GETLUP  ### SET ADD WODE  ### GETLUP	IM \$800 D KEY? NOMNUS NO MCDE YES SET SUBTHACT MODE 8 GETLUP IM \$80	OCCLE NO CLEAR CURRENT PLAYER BE CETLUP STATE NO CLEAR CURRENT PLAYER BE NO CLEAR CURRENT PLAYER BE CETLUP YES, CHANGE CURRENT PLAYER BE CETLUP	NUMERIC KEY? NO. THEN IGNORE IT
	* \$1F6A READ THE KEYBOARD * \$1FE7 HEX TC 7-SEG TABLE * \$1741 DIRECTION REGISTEH * \$1742 DIGIT SELECT PORT * \$1740 SEGMENTS OUTPUT PORT	* * * * * * * * * * * * * * * * * * *	LOAIM ZERO INIT ALL SCORES TO 0 STAAY PLAYER INY CPYIM \$1A BCC CLRLUP STA CURREY INIT CURRENT KEY STA MODE SET MODE TO ADD	STA SEGBUF +01 SECOND DIGIT BLANK LDAIM \$P2 STA CURPLA  *********************  ************	USAN DISCENTING TO THE CONTROL OF THE TOTAL	CMPIM \$00 D KEY? BNE NOMNUS NO SIA MODE YES SET SUBTHACT MODE 0 EEQ GETLUP	UNE CETLUP  CMPIM \$14  DSR  USR  CMPIM \$14  DSR  UNP  CETLUP  UNP  CMPIM \$14  DC  CMPIM \$14  DC  CMPIM  CMP	CMPIN SOA NUMEHIC KEY? BCS DETLUP NO, THEN IGNORE IT
LABELS	* \$1F6A READ THE KEYBOARD * \$1FE7 HEX TC 7-SEG TABLE * \$1741 DIRECTION REGISTEH * \$1742 DIGIT SELECT PORT * \$1740 SEGMENTS OUTPUT PORT	*  INITIALIZATION  *  *  *  *  *  *  *  *  *  *  *  *  *	LOAIM ZERO INIT ALL SCORES TO 0 STAAY PLAYER INY CPYIM \$1A BCC CLRLUP STA CURREY INIT CURRENT KEY STA MODE SET MODE TO ADD	STA SEGBUF +01 SECOND DIGIT BLANK LDAIM \$P2 STA CURPLA  *********************  ************	USAN DISCENTING TO THE CONTROL OF THE TOTAL	CMPIM \$00 D KEY? BNE NOMNUS NO SIA MODE YES SET SUBTHACT MODE 0 EEQ GETLUP	UNE CETLUP  CMPIM \$14  DSR  USR  CMPIM \$14  DSR  UNP  CETLUP  UNP  CMPIM \$14  DC  CMPIM \$14  DC  CMPIM  CMP	CMPIN SOA NUMEHIC KEY? BCS DETLUP NO, THEN IGNORE IT
	\$1F6A READ THE KEYBOAHD \$1FE7 HEX TO 7-SEG TABLE \$1741 DIRECTION REGISTER \$1742 DIGIT SELECT PORT \$1740 SEGMENTS OUTPUT PORT	*  INITIALIZATION  *  *  *  *  *  *  *  *  *  *  *  *  *	ZERO INIT ALL SCORES TO 0 PLAYER \$1A CLALUP CURKEY INIT CURRENT KEY WODE SET MODE TO ADD	STA SEGBUF +01 SECOND DIGIT BLANK LDAIM \$02 CURRENT PLAYER IS #1  STA CURPLA  *****************  * MAINLINE  * * ********************************	S GELUP JSH DISKLAT SCURE & BET KET BIL CAP CURKEY DEBOUNCE  STA CURKEY  CMPIM \$12 PLUS KEY?  BNE NOPLUS NO  LDAIM ZERO YES, SET ADD MODE  STA MODE  BEQ GETLUP	CMPIM \$00 D KEY? BNE NOMNUS NO SIA MODE YES SET SUBTHACT MODE 0 BEG GETLUP CMPIM \$00 C KEY?	UNE CETLUP  CMPIM \$14  DSR  USR  CMPIM \$14  DSR  UNP  CETLUP  UNP  CMPIM \$14  DC  CMPIM \$14  DC  CMPIM  CMP	\$0A NUMERIC KEY?
	* \$1F6A READ THE KEYBOARD * \$1FE7 HEX TC 7-SEG TABLE * \$1741 DIRECTION REGISTEH * \$1742 DIGIT SELECT PORT * \$1740 SEGMENTS OUTPUT PORT	* * * * * * * * * * * * * * * * * * *	00 CLRLUP STAAY PLAYER  LDAIM ZERO INIT ALL SCORES TO 0  INY  CPYIM \$1A  SCC CLRLUP STA MODE SET MODE TO ADD	######################################	CAP IN \$12 PLUS KEY?  CAP TWEY  STA CURKEY  CAP IM \$12 PLUS KEY?  BNE NOPLUS NO  LDAIM ZERO YES, SET ADD MODE  STA MODE  BEQ GETLUP	NOPLUS CMPIM \$00 D KEY?  BNE NOMNUS NO STA MODE YES SET SUBTHACT MODE  BEQ GETLUP NOMNUS CMPIM \$20 C KEY?	02 JAP GETLUP  NOCLH CMPLM \$14 PC KEY?  08 JAP GETLUP  NOCLH CMPLM \$14 PC KEY?  08 JAP UPPLAY YES, CHANGE CURHENT PLAYER	NOPC CMPIN \$0A NUMERIC KEY?  BCS GET:UP NO, THEN IGNORE IT
	* \$1F6A READ THE KEYBOARD * \$1FE7 HEX TC 7-SEG TABLE * \$1741 DIRECTION REGISTEH * \$1742 DIGIT SELECT PORT * \$1740 SEGMENTS OUTPUT PORT	* INITIALIZATION * * ********************************	9 00 CLRLUP STAAY PLAYER INIT ALL SCORES TO 00 9 00 IN STAAY PLAYER IN STAAY PLAYER IN STA CHRLUP STA MODE SET MODE TO ADD	9A STA SEGBUF +01 SECOND DIGIT BLANK  82	WE WE SELLOT JSH DISPLAT SCURE & BEI KET BY BY BY BEINGE YER BEINGE YER BY BEINGE YER BY	00 NOPLUS CMPIM \$00 D KEY? 04 BNE NOMNUS NO 954 STA MODE YES SET SUBTHACT MODE 0 E5 EEQ GETLUP	45 02 JSR CLRCUR YES, CLEAR CURRENT PLAYER BY JSR CKPT NO CKRY?  14 NOCLH CMPIM \$14 PC KEY?  15 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER BY JSR UPPLAY BY JSR UPPLAY PLAYER BY JSR UPPLAY BY	9 BA NOPC CMPIN \$8A NUMERIC KEY? 8 CD BCS GET:UP NO, THEN IGNORE IT 8
	GETKEY * \$1F6A READ THE KEYBOARD TABLE * \$1FE7 HEX TO 7-SEG TABLE PADD * \$1741 DIRECTION REGISTER SBD * \$1742 DIGIT SELECT PORT SAD * \$1740 SEGMENTS OUTPUT PORT	* * INITIALIZATION * * ********************************	A9 00 CLALUP STAAY PLAYER 99 80 00 CLALUP STAAY PLAYER C0 1A CPYIM \$1A 90 FB STA CURKEY INIT CURRENT KEY 85 96 STA MODE SET MODE TO ADD	85 9A STA SEGBUF +01 SECOND DIGIT BLANK A9 02 LDAIM \$02 CURRENT PLAYER IS #1 85 95 **********************************	CS 96 CENTUREY DESCRIPTION OF SET NET BY CS 96 CENTUREY DEBOUNCE CONFIGUREY STA CURKEY DEBOUNCE CONFIGUREY STA CURKEY CS 96 CENTUREY CS 97	C9 8D NOPLUS CMPIM \$8D D KEY?  08 84 BNE NOMNUS NO F8 594 STA MODE YES SET SUBTHACT MODE 8 F8 E5 EEG GETLUP C9 8C NOMNUS CMPIM \$8C C KFY?	20 4F 02 USH CLRCUR YES, CLEAH CURHENT PLAYER B 20 4F 02 USH CLRCUR YES, CLEAH CURHENT PLAYER B 34C 16 02 USH S14 PC KEY?  C9 14 NOCLH CMPIM \$14 PC KEY?  D0 06 06 USH NOPC NO USPLAY YES, CHANGE CURHENT PLAYER B 34C 16 02 UMP GETLUP	C9 BA NOPC CMPIN \$BA NUMERIC KEY?  BB CD BCS GETIUP NO, THEN IGNORE IT B
	* \$1F6A READ THE KEYBOARD * \$1FE7 HEX TC 7-SEG TABLE * \$1741 DIRECTION REGISTEH * \$1742 DIGIT SELECT PORT * \$1740 SEGMENTS OUTPUT PORT	* * INITIALIZATION * * ********************************	9 00 CLRLUP STAAY PLAYER INIT ALL SCORES TO 00 9 00 IN STAAY PLAYER IN STAAY PLAYER IN STA CHRLUP STA MODE SET MODE TO ADD	## ## ## ## ## ## ## ## ## ## ## ## ##	#210 20 85 85 85 85 85 85 85 85 85 85 85 85 85	C9 8D NOPLUS CMPIM \$8D D KEY?  08 84 BNE NOMNUS NO FR E5 PEG GETLUP  C9 8C NOMNUS CMPIM \$8C C KEY?	235 20 4F 02 JSR CLRCUR YES, CLEAH CURRENT PLAYER 0235 20 4F 02 JSR CLRCUR YES, CLEAH CURRENT PLAYER 0235 20 4F 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 20 59 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAY YES, CHANGE CURRENT PLAYER 023F 24 24 16 02 JSR UPPLAYER 025F 24 24 16 02 JSR UPPLAYER 025F 24 24 16 02 JSR UPPLAYER 025F 24 24 16 02 JSR UPPLAYER 0	9 BA NOPC CMPIN \$8A NUMERIC KEY? 8 CD BCS GET:UP NO, THEN IGNORE IT 8
	8288   GETKEY * \$1F6A READ THE KEYBOAHD   8288   TABLE * \$1F67 HEX TC 7—SEG TABLE   81741 DIRECTION REGISTER   81742 DIGIT SELECT PORT   81748 SEGMENTS OUTPUT	* INITIALIZATION * * ********************************	0202 99 00   CLRLUP STAAY PLAYER   NIT ALL SCORES TO 0   0204 99 00 00 CLRLUP STAAY PLAYER   NIT CURRENT KEY   0208 99 66   STA MODE SET MODE TO ADD	### ##################################	210 65 96 CMP CURKEY DEBOUNCE 210 65 96 CMP CURKEY DEBOUNCE 210 85 96 STA CURKEY 221 DØ Ø6 CMPIM \$12 PLUS KEY? 223 A9 ØØ LDAIM ZERO YES, SET ADD MODE 225 85 94 STA MODE 227 FØ ED BEQ GETLUP	0229 C9 00 NOPLUS CMPIM \$00 D KEY?   0	71: 0233 00 06 06 06 07: 07: 0235 00 06 06 06 07: 07: 0235 00 06 06 06 07: 07: 0238 09 14 00 06 07: 07: 0238 09 14 00 06 07: 07: 0238 09 14 00 06 07: 07: 0238 09 08 06 07: 07: 0238 09 08 06 07: 07: 07: 07: 07: 07: 07: 07: 07: 07:	8: 8245 C9 8A NOPC CMPIK \$8A NUMEHIC KEY? 1: 8247 B8 CD BCS GETLUP NO, THEN IGNORE IT 8

			**	
CVTSEG CONVEHT TO SEGS IN NUMLUP QUIT AFTER 2 DISSEG GO DISPLAY ************************************	* × ×	TEMP RESTORE BYTE  M \$CF CLEAR HI NYE  V TABLE LCAD SEOMENT  INDEX CET BUFFER  INDEX NEXT BUFFER  INDEX NEXT BUFFER  **********************************	* DISGET : DISPLAY CURHENT PLAYER AND *  * HEAD KEYBOARD  * **********************************	CVISEC CVISEC CVISEC DISSEC DISSEC GETKEY ** CCDES F
0.0EY DEG DEG ******	**************************************	LDA ANDIM TAX LDAX LDAX LDAX STAZX INC HTS	* * DISSET * * * * * * * * * * * * * * * * * *	* * * * * * * * * * * * * * * * * * *
2 2	15	Ļ.	4	20 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
E7 29		99 98 98	67 99 98 98	88 87 87 88 88 88 88 88 88 88 88 88 88 8
1 88 1 88 4 4 4 4 C	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	
020E 02E1 02E2	02E7 02E7 02E8 02EE 02EC 02EC	02577 02578 02570 03877 0383	83388 83388 83388 83388 841114	88888888888888888888888888888888888888
	8214: 8214: 8214: 8214: 8214: 8228: 8222:	02255. 02226. 02227. 022329. 02332. 02332.	6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6	88888888888888888888888888888888888888
			#	
ODE		<b>*</b>	# 0F	* * * * * LI
DE CT MODE	UK ANS RE	# #	E OF #	* * * * * LT * * * * * * * * * * * * * *
) MODE.	UK ANS RE	# 40 E	3YTE OF # HI BYTE OF # AY	* * * * * LT * * * * * * * * * * * * * *
ADD	EY? IN BUFFEH NT PLAYEH ODE FOR HUMANS TE OF SCORE BTRACT?	YTE YTE HI BYTE OF # BINAHY	SCORE LO BYTE OF # TTE AACT HI BYTE OF # 3INARY	**************************************
1 1	KEY? I IN BUFFER I AENT PLAYER MODE FOR HUMANS SYTE OF SCORE SUBTRACT?	BYTE BYTE A I BYTE OF # IE S BINAHY	SCORE LO BYTE OF # TTE AACT HI BYTE OF # 3INARY	**************************************
1 1	WERIC KEY?  5, PUT IN BUFFER  1 CURRENT PLAYER  CIMAL MODE FOR HUMANS  CIMAL WODE FOR ECOME  D OH SUBTRACT?	SCORE D LO BYTE SAVE T HI BYTE 1 ADD HI BYTE OF # D SAVE CK TO BINAHY	CURRENT SCORE  JETHACT LO BYTE OF #  ST HI BYTE  11 SUBTRACT HI BYTE OF #  ND SAVE  ACK TO BINARY	**************************************
PLUS KEY? NO YES, SET OP D KEY? N NO YES, SET	NUMERIC KEY? YES, PUT IN BUFFER NO, EXIT A GET CURRENT PLAYER DECIMAL MODE FOR HUMANS ACT LO BYTE OF SCORE ADD OR SUBTRACT?	NT SCORE ADD LO BYTE S SAVE GET HI BYTE +Ø1 ADD HI BYTE OF # AND SAVE BACK TC BINAHY	SUBTRACT LO BYTE OF # AND SAVE GET HI BYTE +Ø1 SUBTRACT HI BYTE OF # AND SAVE BACK TO BINARY	**************************************
PLUS KEY? NO YES, SET OP D KEY? N NO YES, SET	NUMERIC KEY? YES, PUT IN BUFFER NO, EXIT A GET CURRENT PLAYER DECIMAL MODE FOR HUMANS ACT LO BYTE OF SCORE ADD OR SUBTRACT?	CURRENT SCORE JABUF ADD LO BYTE AYER & SAVE AYER GET HI BYTE JABUF +01 ADD HI BYTE OF # AYER AND SAVE BACK TO BINAHY	ER FROW CURRENT SCORE UMBUF SUBTHACT LO BYTE OF # LAYER AND SAVE UMBUF +Ø1 SUBTHACT HI BYTE OF # LAYER AND SAVE LAYER AND SAVE EACK TO BINARY	**************************************
\$12 PLUS KEY? CKD NO ZERO YES, SET WDDE UDLOOP SØD D KEY? CKNUM NO MODE YES, SET UDLOOP	\$0A NUMERIC KEY? UPLUP YES, PUT IN BUFFER NO, EXIT CURPLA GET CURRENT PLAYER DECIMAL MODE FOR HUMANS PLAYER GET LO BYTE OF SCORE MODE ADD OR SUBTRACT? SUBTRK SUBTRACT	TO CURRENT SCORE  NUMBUF ADD LO BYTE  X PLAYER & SAVE  X PLAYER GET HI BYTE  NUMBUF +Ø1 ADD HI BYTE OF #  X PLAYER AND SAVE  BACK TO BINARY	ER FROW CURRENT SCORE UMBUF SUBTHACT LO BYTE OF # LAYER AND SAVE UMBUF +Ø1 SUBTHACT HI BYTE OF # LAYER AND SAVE LAYER AND SAVE EACK TO BINARY	**************************************
\$12 PLUS KEY? CKD NO ZERO YES, SET WODE UDLOOP KODE CKNUM NO MODE YES, SET	NUMERIC KEY? YES, PUT IN BUFFER NO, EXIT A GET CURRENT PLAYER DECIMAL MODE FOR HUMANS ACT LO BYTE OF SCORE ADD OR SUBTRACT?	TO CURRENT SCORE  NUMBUF ADD LO BYTE  X PLAYER & SAVE  X PLAYER GET HI BYTE  NUMBUF +Ø1 ADD HI BYTE OF #  X PLAYER AND SAVE  BACK TO BINARY	ACT BUFFER FROW CURRENT SCORE SEC STAZX PLAYER AND SAVE INX LDAZX PLAYER GET HI BYTE OF # SEC NUMBUF +#1 SUBTRACT HI BYTE OF # SEC NUMBUF +#1 SUBTRACT HI BYTE OF # STAZX PLAYER AND SAVE CLD EACK TO BINARY	**************************************
CMPIM \$12 PLUS KEY? BNE CKD NO LDAIM ZERO YES, SET STA MODE BEQ UDLOOP CMPIM \$0D D KEY? BNE CKNUM NO STA MODE YES, SET BEQ UDLOOP	CWPIM \$0A NUMERIC KEY? BCC UPLUP YES, PUT IN BUFFER RTS NO, EXIT LDX CURPLA GET CURRENT PLAYER SED DECIMAL MODE FOR HUMANS LDAZX PLAYER GET LO BYTE OF SCORE LDY MODE ADD OR SUBTRACT? BNE SUBTRK SUBTRACT	BUFFER TO CURRENT SCORE  CLC ADC NUMBUF ADD LO BYTE STAZX PLAYER & SAVE INX LDAZX PLAYER GET HI BYTE ADC NUMBUF +#1 ADD HI BYTE OF # STAZX PLAYER AND SAVE CLD RACK TO BINAHY RTS	ACT BUFFER FROW CURRENT SCORE SEC STAZX PLAYER AND SAVE INX LDAZX PLAYER GET HI BYTE OF # SEC NUMBUF +#1 SUBTRACT HI BYTE OF # SEC NUMBUF +#1 SUBTRACT HI BYTE OF # STAZX PLAYER AND SAVE CLD EACK TO BINARY	**************************************
\$12 PLUS KEY? CKD NO ZERO YES, SET WDDE UDLOOP SØD D KEY? CKNUM NO MODE YES, SET UDLOOP	\$0A NUMERIC KEY? UPLUP YES, PUT IN BUFFER NO, EXIT CURPLA GET CURRENT PLAYER DECIMAL MODE FOR HUMANS PLAYER GET LO BYTE OF SCORE MODE ADD OR SUBTRACT? SUBTRK SUBTRACT	TO CURRENT SCORE  NUMBUF ADD LO BYTE  X PLAYER & SAVE  X PLAYER GET HI BYTE  NUMBUF +Ø1 ADD HI BYTE OF #  X PLAYER AND SAVE  BACK TO BINARY	FER FROW CURRENT SCORE NUMBUF SUBTRACT LO BYTE OF # PLAYER AND SAVE NUMBUF +01 SUBTRACT HI BYTE OF # PLAYER AND SAVE BACK TO BINARY	**************************************
CMPIM \$12 PLUS KEY? BNE CKD NO LDAIM ZERO YES, SET STA MODE BEQ UDLOOP CKD CMPIM \$00 D KEY? ENE CKNUM NO STA MODE YES, SET BEQ UDLOOP	CKNUM CMPIM. \$0A NUMERIC KEY? BCC UPLUP YES, PUT IN BUFFER ADDUM LDX CURPLA GET CURRENT PLAYER SED DECIMAL MODE FOR HUMANS LDAZX PLAYER GET LO BYTE OF SCORE LDY MODE ADD ON SUBTRACT? BNE SUBTRACT	ADD BUFFER TO CURRENT SCORE  CLC  ADC  NUMBUF ADD LO BYTE  STAZX PLAYER & SAVE  INX  LDAZX PLAYER GET HI BYTE  ADC  NUMBUF +01 ADD HI BYTE OF #  STAZX PLAYER AND SAVE  CLD  BACK TO BINAHY  RTS	SUBTRACT BUFFER FROW CURRENT SCORE  SUBTRY SEC  SRC NUMBUF SUBTRACT LO BYTE OF #  INX  LDAZX PLAYER AND SAVE  SRC NUMBUF +Ø1 SUBTRACT HI BYTE OF #  STAZX PLAYER AND SAVE  CLD  EACK TO BINARY  HTS	**************************************
9 12 CMPIM \$12 PLUS KEY? 8 86 LDAIM ZERO YES, SET 5 94 STA MODE 8 E2 BEQ UDLOOP 9 80 CKD CMPIM \$80 D KEY? 8 84 STA MODE YES, SET 8 DA BEQ UDLOOP	DA CKNUM CMPIM \$0A NUMERIC KEY?  D3 BCC UPLUP YES, PUT IN BUFFEH  RTS NO, EXIT  95 ADDUM LDX CURPLA GET CURRENT PLAYER  B0 DECIMAL MODE FOR HUMANS  LDAZX PLAYER GET LO BYTE OF SCORE  94 LDY MODE ADD OH SUBTRACT?  0E	ADD BUFFER TO CURRENT SCORE  CLC ADC NUMBUF ADD LO BYTE  STAZX PLAYER & SAVE INX BØ LDAZX PLAYER GET HI BYTE AØ ADC NUMBUF +Ø1 ADD HI BYTE OF #  STAZX PLAYER AND SAVE CLD BACK TO BINAHY RTS	SUBTRACT BUFFER FROW CURRENT SCORE  SUBTRK SEC  SEC NUMBUF SUBTRACT LO BYTE OF #  STAZX PLAYER AND SAVE  INX  BØ LDAZX PLAYER GET HI BYTE  AØ SPC NUMBUF +Ø1 SUBTRACT HI BYTE OF #  STAZX PLAYER AND SAVE  STAZX PLAYER AND SAVE  CLO BACK TO BINARY  HTS	##************************************
C9 12 CMPIM \$12 PLUS KEY? DØ Ø6 A9 ØØ LDAIM ZERO YES, SET 85 94 STA MODE FØ E2 BEG UDLOOP C9 ØD CKD CMPIM \$ØD D KEY? DØ Ø4 STA MODE YES, SET FØ DA BEG UDLOOP	C9 0A CKNUM CMPIM \$0A NUMERIC KEY? 60 BCC UPLUP YES, PUT IN BUFFER 60 ATS NO, EXIT A6 95 ADDUM LDX CURPLA GET CURRENT PLAYER F8 BC LOAZX PLAYER GET CURRENT PLAYER F8 CORE DECIMAL MODE FOR HUMANS F8 LOAZX PLAYER GET LO BYTE OF SCORE A4 94 LDY MODE ADD OH SUBTRACT? D0 0E SUBTRACT	ADD BUFFER TO CURRENT SCORE  CCC NUMBUF ADD LO BYTE  STAZX PLAYER & SAVE E8 INX E9 ADC NUMBUF +#1 ADD HI BYTE OF #  STAZX PLAYER AND SAVE  CCD BACK TO BINAHY  68 RTS	SUBTRACT BUFFER FROW CURRENT SCORE  SUBTRY SEC SEC NUMBUF SUBTRACT LO BYTE OF #  STAZX PLAYER AND SAVE E8 LDAZX PLAYER GET HI BYTE OF #  SEC NUMBUF +#1 SUBTRACT HI BYTE OF #	**************************************
### ### ### ### ### ### ### ### ### ##	### BCC UPLUP YES, PUT IN BUFFER BCC UPLUP YES, PUT IN BUFFER BCC UPLUP YES, PUT IN BUFFER BCAS 60 ADDUM LDX CURPLA GET CURRENT PLAYER BCAS FS CORE LOAZX PLAYER GET CURRENT PLAYER BCAS FS CORE LOAZX PLAYER GET LO BYTE OF SCORE BCAS A4 94 LDY WODE ADD OH SUBTRACT?	ADD BUFFER TO CURRENT SCORE  B288 65 9F	SUBTRACT BUFFER FROW CURRENT SCORE  ### SUBTRY SEC ### SEC NUMBUF SUBTRACT LO BYTE OF ### STAZX PLAYER AND SAVE ### STAZX PLAYER GET HI BYTE ### SEC NUMBUF +#1 SUBTRACT HI BYTE OF ### SEC NUMBUF HIS	##************************************

0265:				25/02/07	DISSEG			SET 6530 TO
0266:				17		STA	PADD	DUTPUT
0267:						LDYIM		SELECT DIGIT 1 FIRST
Ø268:						LDAIM		
0269:					to an area of the contract	STA	INDEX	
0270:					DISLUP	TO SEE SHOW A WORLDOOD	INDEX	GET BUFFER INDEX
0271:							SEGBUF	GET A DIGIT FROM BUFFER
Ø272:				5000		LDXIM		G. I. Sen. Louis Trans.
Ø273:				2000		STX	SAD	CLEAR_DISPLAY
0274:				17		STY	SED	SELECT DIGIT
Ø275:				17		STA	SAD	LITE DIGIT
Ø276:			7F			LDXIM	\$7F	Distributed that should be selected for
Ø277:					WAIT	DEX	20022000	LEAVE IT ON FOR A WHILE
Ø278:						BNE	WAIT	
0279:			98			INC	INDEX	NEXT BUFFER POSION
0280:						INY		ZIM CONTROLOGICAL IN CONTROLOGICA IN CONTROLOGICA
Ø281:			4000			INY	Decomple)	SELECT NEXT DIGIT
Ø282:						CPYIM		DUN YET?
0283:						BCC	DISLUP	NOPE
Ø284:						LDAIM	100000000000000000000000000000000000000	
Ø285:				0.00		STA	SED	TURN OFF SEGS
0286:			41	17		STA	PADD	TURN OFF 6530
0287:	Ø358	60				RTS		
		Ob						
0288:		OP						
Ø288: Ø289:		G D						
0288: 0289: 0290:		G.E						******
0288: 0289: 0290: 0291:		O.			*	*****		*
0288: 0289: 0290: 0291: 0292:		G.			*	*****		
0288: 0289: 0290: 0291: 0292: 0293:		Ge			* * SHF	******	SHIFT H	* <ey *="" *<="" into="" numbuf="" td=""></ey>
0288: 0289: 0290: 0291: 0292: 0293: 0294:					* * SHF *	****** FKEY :	SHIFT H	*
0288: 0289: 0290: 0291: 0292: 0293: 0294: 0295:	Ø359	ØA			* * SHF	***** FKEY : *****	SHIFT H	* <ey into="" numbuf<="" td=""></ey>
0288: 0289: 0290: 0291: 0292: 0293: 0294: 0295:	Ø359 Ø35A	ØA ØA			* * SHF *	****** FKEY : ****** ASLA ASLA	SHIFT H	*  KEY INTO NUMBUF
0288: 0289: 0290: 0291: 0292: 0293: 0294: 0295: 0297:	Ø359 Ø35A Ø35B	ØA ØA ØA			* * SHF *	****** FKEY : ****** ASLA ASLA ASLA	SHIFT H	* <ey into="" numbuf<="" td=""></ey>
0288: 0289: 0290: 0291: 0292: 0293: 0294: 0295: 0296: 0297:	Ø359 Ø35A Ø35B Ø35C	ØA ØA ØA			* * SHF *	****** FKEY : ****** ASLA ASLA ASLA ASLA	SHIFT :	*  KEY INTO NUMBUF *  *  *******************************
0288: 0289: 0290: 0291: 0292: 0293: 0294: 0295: 0297: 0297: 0298:	Ø359 Ø35A Ø35B Ø35C Ø35D	ØA ØA ØA ØA			*     * SHF     *     ****** SHFKEY	FKEY :  *****  ASLA  ASLA  ASLA  ASLA  ASLA  ASLA  ASLA	SHIFT :	*  *  *  *  *  *  *  *  *  *  *  *  *
0288: 0289: 0290: 0291: 0292: 0293: 0294: 0295: 02978: 0298: 0298:	0359 035A 035B 035C 035D 035F	ØA ØA ØA A2 2A	<b>Ø</b> 4		* * SHF *	KEY: ***** ASLA ASLA ASLA ASLA ASLA ASLA ASL	SHIFT :	*  KEY INTO NUMBUF *  **********************************
0288: 0289: 0290: 0291: 02934: 02956: 02978: 0298: 0298: 03001:	0359 035A 035B 035C 035C 035F 0360	ØA ØA ØA ØA 2A 2A	Ø4 9F		*     * SHF     *     ****** SHFKEY	FKEY: ASLA ASLA ASLA ASLA ASLA ASLA ASLA ASL	SHIFT : ***********************************	*  KEY INTO NUMBUF *  **********************************
0288: 02890: 0290: 0291: 02934: 02956: 029760: 02990: 02990: 02990: 0300: 0300:	0359 035A 035C 035C 035F 0360 0360	ØA ØA ØA A2 26 26	Ø4 9F		*     * SHF     *     ****** SHFKEY	FKEY: ASLA ASLA ASLA ASLA ASLA ASLA ASLA ASL	SHIFT :	*  KEY INTO NUMBUF *  **********************************
0288: 02890: 0291: 02923: 02940: 02999678: 0299901: 0299901: 029901: 0303: 0303: 0303:	Ø359 Ø358 Ø355 Ø355 Ø368 Ø364	ØA ØA ØA 2A 6 6 CA	Ø4 9F AØ		*     * SHF     *     ****** SHFKEY	FKEY: ASLA ASLA ASLA ASLA ASLA ASLA ASLA ASL	SHIFT N	*  KEY INTO NUMBUF *  **********************************
0288: 02890: 02991: 02994: 02996678: 0299901: 0299901: 0299901: 029901: 0293023030303030303030303030303030303030	Ø359 Ø358 Ø355 Ø355 Ø365 Ø364 Ø364	Ø Ø A A A A A A A A A A A A A A A A A A	Ø4 9F AØ		*     * SHF     *     ****** SHFKEY	ASLA ASLA ASLA ASLA ASLA ASLA BOL BOL BOL BOL BOL BOL BOL	SHIFT : ***********************************	*  KEY INTO NUMBUF *  **********************************
8:: 82899: 829994:: 82299567: 822999999: 82299999: 82333333333333333333333333333333333333	Ø359 Ø358 Ø355 Ø355 Ø365 Ø364 Ø364	Ø Ø A A A A A A A A A A A A A A A A A A	Ø4 9F AØ		*     * SHF     *     ****** SHFKEY	FKEY: ASLA ASLA ASLA ASLA ASLA ASLA ASLA ASL	SHIFT N	*  KEY INTO NUMBUF *  **********************************
8: 82899: 822991: 82299567: 822999999: 822999999: 83383383383383385: 83385:	Ø359 Ø358 Ø355 Ø355 Ø365 Ø364 Ø364	Ø Ø A A A A A A A A A A A A A A A A A A	Ø4 9F AØ		*     * SHF     *     ****** SHFKEY	ASLA ASLA ASLA ASLA ASLA ASLA BOL BOL BOL BOL BOL BOL BOL	SHIFT N	*  KEY INTO NUMBUF *  **********************************
8: 82899: 82991: 82299567: 8229999991: 8229999991: 83383383838383838383838383838383838383	Ø359 Ø358 Ø355 Ø355 Ø365 Ø364 Ø364	Ø Ø A A A A A A A A A A A A A A A A A A	Ø4 9F AØ		*     * SHF     *     ****** SHFKEY	ASLA ASLA ASLA ASLA ASLA ASLA BOL BOL BOL BOL BOL BOL BOL	SHIFT N	*  KEY INTO NUMBUF *  **********************************
8: 82899: 822991: 82299567: 822999999: 822999999: 83383383383383385: 83385:	Ø359 Ø358 Ø355 Ø355 Ø365 Ø364 Ø364	Ø Ø A A A A A A A A A A A A A A A A A A	Ø4 9F AØ		*     * SHF     *     ****** SHFKEY	ASLA ASLA ASLA ASLA ASLA ASLA BOL BOL BOL BOL BOL BOL BOL	SHIFT N	*  KEY INTO NUMBUF *  **********************************

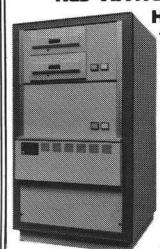
#### Symbol Table

ACDUM	Ø2A6	CKD	0299	CKNUM	Ø2A1	CLRCUR	Ø24F
CLALUP	0204	CURKEY	0096	CURPLA	0095	CVTSEG	Ø2E7
DISGET	0306	DISLUP	@334	CISMIN	Ø2D3	DISNUM	Ø2CB
DISSEC	0329	GETKEY	1F6A	GETLUP	0216	INDEX	0098
MODE	0094	NOCLR	@23B	NOMNUS	0231	NOPC	0245
NOPLUS	0229	NUMBUF	009F	NUMLUP	Ø208	PADD	1741
PLAYER	0080	SAD	1740	SED	1742	SCORER	0200
SEGBUF	0099	SHFKEY	0359	SHFLUP	Ø35F	SUBTRK	Ø2BD
TABLE	1FF7	TEMP	0097	UDLOOP	Ø278	UPDATE	Ø26E
UPLUP	Ø278	UPPLAY	0259	WAIT	0345	ZERO	0000

#### April Fools On Us

We fear that a second class mail bag full of issue 21 may have been lost by the US Postal Service. If you live in the Arkansas, Louisiana, or Georgia area and did not receive your copy of MICRO, 21, please let us know.

# The Age of Affordable Computing Has Arrived.



YOU ?

\$12,900

# The C3-B

# **OHIO SCIENTIFIC**Challenger Series

World's largest line of microcomputers for Personal and Business Use There's one for every budget. And it grows with you, too.

C1P: \$349!

THE **ALL-IN-ONE** Ohio Scientific's Challenger 1P is the easy-to-use home computer. It does a lot more for a lot less! Just connect it to your TV\* and a player. That's all there is to it. There's nothing to build. You can create your own personal programs. Or select from a whole library of programs on **EO GAME** low-cost cassettes. Everything from teaching arithmetic to spectacular video games to balancing your HE checkbook Something for every

WALLENCED ID



#### COMPUTERSHOP

Boston 590 Comm. Ave. (across from B.U.) 247-0700 Cambridge 288 Norfolk St. (near M.I.T.) 661-2670

# NOW YOUR APPLE II CAN PERFORM JUST LIKE THE BIG BOYS

If you're a businessman who demands ultimate performance from your Apple II, then take a look at this outstanding General Ledger Package from Small Business Computer Systems (SBCS).

#### It features

- . 6 digit account numbers
- · 31 character account name.
- Ten levels of subtotals giving you a more detailed income statement and balance sheet.
- . Departmentalizing . . . up to nine departments.
- Flexibility adaptable to any printer and either cash or accural accounting methods.
- · Cash Journal allows a 33 character transaction description and automatically generates the appropriate offsetting entry.
- . You can print the balance sheet and income statement for the current month, current quarter, or any of the previous three quarters. This year's or last year's totals are also included on the income statement. Or a special report that lists the current account balance for selected accounts.
- Higher number of entries from an external source as many as 1,000 per session.
- . No limit on entries giving you the opportunity to make your entries as many times or as often as you want.
- · With high speed printer routines and other special features of our conversion, processing performance does not decrease dramatically at the system limits.
- Look at these examples of times required to update the chart and print the audit trail. With 133 item chart of accounts, 700 postings into 70 regular accounts: less than 20 min. With 133 item chart of accounts, 1000 postings into 70 regular accounts: less than 30 min. With 210 item chart of accounts, 1000 postings into 125 regular accounts: less than 40 min.
- Coming early this year capability to archive up to 2,500 postings. The chart of accounts will also be archived to maintain the opening balance for the archive period.

In the final analysis, your financial statements are what this General Ledger is all about. And with this General Ledger Package you can format your own balance sheet and income statement. As well, department financial statements may be formated differently. You have complete freedom to place titles and headings where you want them, skip lines or pages between accounts and generate subtotals and totals throughout the reports - up to ten levels if you need them.

And coming early in 1980, SBCS will present the Accounts Payable/Accounts Receivable Package you have been waiting for.

Just compare these numbers against any package on the market today:\*

	5 inch	8 inch
	disc	disc
Vendors or customers	700	1,800
Payable Transactions	350	750
Payable Invoices	380	840
Receivable Transactions	600	1,300
Receivable Invoices	600	1,300

These are maximum numbers that you can put on a disc if you're using the disc only for these respective data files.

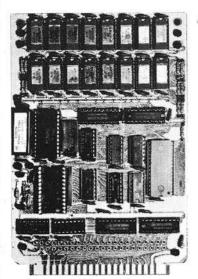
We are an authorized converter for Osborne/McGraw-Hill, providing you with business packages that will do everything the Osborne General Ledger will do in addition to many features we have added.

Call or write:

#### Small Business Computer Systems

4140 Greenwood Lincoln, Nebraska 68504 (402) 467-1878

#### KIM/SYM/AIM-65-32K EXPANDABLE RAM DYNAMIC RAM WITH ON BOARD TRANSPARANT REFRESH: THAT IS COMPATIBLE WITH KIM/SYM/AIM-65 OTHER 6502 BASED MICROCOMPUTERS.



ASSEMBLED/

### \$419.00 WITHOUT RAM CHIPS ### \$349.00 D TO GET DEST HARD TO GET PARTS ONLY (NO RAM CHIPS) .. \$109.00 BARE BOARD AND MANUAL

- PLUG COMPATIBLE WITH KIM/SYM/AIM-65.
- PLUG COMPATIBLE WITH KIM/SYM/AIM-65.
  MAY BE CONNECTED TO PET USING ADAPTOR
  CABLE. SS44-E BUS EDGE CONNECTOR.
  USES -5V ONLY. (SUPPLIED FROM HOST
  COMPUTER BUS). 4 WATTS MAXIMUM.
  BOARD ADDRESSABLE IN 4K BYTE BLOCKS
  WHICH CAN BEINDEPENDENTLY PLACED ON
  4K BYTE BOUNDARIES ANYWHERE IN A 64K
  BYTE ADDRESS SPACE.
  ASSEMBLED AND TESTED BOARDS ARE
  GUARANTEED FOR ONE YEAR. AND
  PURCHASE PRICE IS PULLY REFUNDABLE IF
  BOARD IS RETURNED UNDAMAGED WITHIN
  14 DAYS.
- \* FULL DOCUMENTATION

#### INTERFACE KIT

CONNECTS THE ABOVE 32K EXPANDABLE RAM TO A 4K OR 8K PET. CONTAINS EXPANSION INTERFACE CABLE, BOARD STANDOFFS. POWER SUPPLY MODIFICATION KIT AND COMPLETE INSTRUCTIONS.

6502, 64K BYTE RAM AND CONTROLLER SET 6502, 64K BYTE RAM AND CONTROLLER SET
MAKE 64K BYTE MEMORY FOR YOUR 6800 OR
6502. THIS CHIP SET INCLUDES:
\* 32 MSK 4116-3 16KX1, 200 NSEC RAMS.
\* 1 MC3480 MEMORY CONTROLLER.
\* 1 MC3242A MEMORY ADDRESS
MULTIPLEXER AND COUNTER.
DATA AND APPLICATION SHEETS. PARTS
TESTED AND GUARANTEED.

\$295.00 PER SET

**16K X I DYNAMIC RAM** 

THE MK4116-3 IS A 16,384 BIT HIGH SPEED NMOS, DYNAMIC RAM. THEY ARE EQUIVALENT TO THE MOSTEK. TEXAS INSTRUMENTS, OR MOTOROLA 4116-3.

- \* 200 NSEC ACCESS TIME, 375 NSEC CYCLE

- I PIN TTL COMPATIBLE.
  BURNED IN AND FULLY TESTED.
  PARTS REPLACEMENT GUARANTEED FOR
  ONE YEAR.

\$8.50 EACH IN QUANTITIES OF 8



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.

ALL ASSEMBLED BOARDS AND MEM-ORY CHIPS CARRY A FULL ONE YEAR REPLACEMENT WARANTY.

# -ACTION, STRATEGY, AND FANTASYfor the SERIOUS games player and his APPLE II

#### Brain Games - 1 demands ingenuity.

Two players bombard radioactive material with protons and electrons until it reaches critical mass and sets up a **Nuclear Reaction. Dodgem** requires you to outmaneuver another player to get your pieces across the board first. **Dueling Digits** and **Parrot** challenges your ability to replicate number and letter sequences. **Tones** lets you make music with your Apple (16K) CS-4004 \$7.95. **Strategy Games** and **Brain Games** are on one disk (16K) CS-4503 \$14.95.

Strategy Games - 1 keeps games players in suspense.

You and your opponent trail around the screen at a quickening pace attempting to trap each other in your **Blockade.** A 7 category quiz game will certify you as a **Genius** (or an errant knave!). Beginners will meet their master in **Checkers. Skunk** and **UFO** complete this classic collection (16K) CS-4003 \$7.95

#### Know Yourself through these valid self-tests.

Find out how your life style effects your Life Expectancy or explore the effects of Alcohol on your behavior. Sex Role helps you to examine your behavior and attitudes in light of society's concept of sex roles. Psychotherapy compares your feelings, actions, and phobias to the population's norms and Computer Literacy tests your microcomputer savvy. A fun and instructional package (16K) CS-4301 \$7.95. Know Yourself and CAI Programs are on one disk (16K) CS-4503 for \$14.95





#### You're in command in Space Games - 1.

Maneuver the TIE fighters into your blaster sights and zap them with your lasers to save the rebel base camp from annihilation in **Star Wars.** Rocket Pilot is an advanced real time take off and landing game. High resolution graphics, exploding saucers and sound effects add to the suspense as you repel the **Saucer Invasion.** Finally, a bonus graphics demonstration, **Dynamic Bouncer** (16K) CS-4001 \$7.95. **Space Games** and **Sports Games** are on one disk (16K) CS-4501 for \$14.95

# ACTION

Sports Games - 1 puts you in the Apple World Series

Take the field in the **Great American Computer Game.** Mix up your pitches to keep the batter off balance. Move your fielders to snag the ball before he gets to first. Balls and strikes, double plays, force outs, and errors let you play with a realistic strategy. Also in the line up—**Slalom**, a championship downhill ski race, **Torpedo Alley**, and **Darts** (16K) CS-4002 \$7.95. **Space Games** and **Sports Games** are on one disk (16K) CS-4501 for \$14.95

It's easy to order SENSATIONAL SOFTWARE for your Apple II.

Send payment plus \$1.00 shipping and handling in the U.S. (\$2.00 foreign) to Creative Computing Software, P.O. Box 789-M, Morristown, N.J. 07960. N.J. residents add \$1.00 sales tax. Visa, Master Charge and American Express orders may be called in toll free to 800-631-8112 (in N.J. 201-540-0445).

#### **OSI BASIC in ROM**

While the various Microsoft BASICs are easy to use, they are difficult to understand due to an intentional lack of documentation. To help understand your OSI BASIC, a table of the locations of the subroutines to service the main commands is presented. The program which generated the table is provided as a starting point for you to explore your BASIC.

E.D. Morris, Jr. 3200 Washington Midland, MI 48640

A previous article in Micro 18:9 by S.R. Murphy gave a peek into OSI BASIC in ROM by listing a number of scratch pad locations in page zero. In the present article, I wish to delve further into the inner workings of BASIC by explaining the dispatch table.

At the bottom of the BASIC ROMs, between \$A000 and \$A083, is a list of addresses known as the dispatch table. These are the starting addresses of all the machine subroutines needed to carry out the BASIC keywords such as END, FOR, NEXT etc. The addresses are in hexidecimal in the normal machine format of low byte first followed by the high order byte. For example, starting at \$A000 you find the data:

\$A000 39 \$A001 A6 \$A002 55 \$A003 A5

Thus the first two entries in the dispatch table are \$A639 and \$A555. These point to subroutines in the BASIC ROMs.

Now we need to know what each subroutine does. Conviently there is another table starting at \$A084 containing a list of all the BASIC keywords. The first entries in this table are:

\$A084 45 \$A085 4E \$A086 C4 \$A087 46 \$A088 4F \$A089 D2

Except for the C4 and D2, the data looks like ASCII code. If the high order bit

is removed from C4 and D2, then it is ASCII code for ENDFOR. You can demonstrate the list of keywords for yourself by running the program:

10 FOR X = 41092 TO 41315

20 Y = PEEK (X)

30 PRINT CHR\$ (Y);

40 NEXT

If you have the OSI graphics character generator, the last letter of each word will be a graphics character instead of a letter. The high bit being set is used to separate the entries in the word list. To convert these to letters and leave a space between key words, add the following line to the above program:

25 IF Y ➤ 127 THEN PRINT CHR\$(Y—128);;Y=32

Now we have two lists, one of addresses and one of functions. These can be combined to give an address for each function.

END \$A639 FOR \$A555

However things are not quite that simple. Unfortunately the two tables are not strictly in the same order. Also some of the address entries refer to the subroutine location and others to the location, less one. The address table is further complicated in the case of the arithmetic operators by a third entry which is the precedence value.

Following is a BASIC program that sorts out these quirks and outputs a list of BASIC KEYWORDS together with the hex address of the machine code

associated with that keyword. Notice that the program does not contain data statements, rather PEEK's directly at your BASIC ROM's. The program steps through the dispatch table printing out each address. The value of Q is added to each address and is either 1 or 0. The correct keyword is found by PEEKing at D until a character is found with the high bit set.

The subroutine at line 500 converts a binary word into ASCII digits for printing.

For those of you who have trouble with this program or for those who have a sore index finger from typing in that 24K game program, I am providing an output listing. However I urge you to run it yourself to prove all this stuff is really "in there." The BASIC program also contains information about the location and structure of the two tables.

Looking at the sample run, the addresses for END and FOR found earlier, are incorrect by one byte. Users of the USR function know that the subroutine address must be placed at \$000B and \$000C. The dispatch table associates location \$000A with the USR function. Location \$000A contains 4C or JMP which completes the three byte instruction.

It is interesting to note that the BASIC keyword table is identical to a numerical listing of the BASIC tokens(MICRO 15:20). The keywords TAB, TO, THEN, and STEP are missing from the dispatch table. However these commands are never used alone but always occur with another BASIC keyword (PRINT, FOR, IF and FOR-NEXT). The purists will note the absence of AND, OR, GREATER, LESS

and EQUALS. I must confess, these did not fit neatly into my BASIC program.

If you have ever tried to make sense of "that 8K block of data up there at \$A000," it looked like a hopeless task. With the dispatch table at hand, you can break it down and attack one function at a time.

Sample Run (Program output listing)

**END** A63A A556 FOR **AA40** NEXT A70C DATA A923 INPUT DIM AD01 READ A94F LET A7B9 **A6B9 GOTO** RUN A691 A73C IF RESTORE A61A **GOSUB** A69C RETURN A6E6 A74F REM A638 STOP A75F ON NULL A67B WAIT B432 LOAD FFF4 FFF7 SAVE **AFDE** DEF POKE B429 A82F PRINT CONT A661 **A4B5** LIST A68C CLEAR A461 NEW SGN **B7D8** B862 INT ABS B7F5 USR 000A **AFAD** FRE POS **AFCE** SQR BAAC RND BBC<sub>0</sub> LOG **B5BD** EXP BB1B **BBFC** COS SIN BC03 TAN BC4C ATN **BC99** PEEK **B41E B38C** LEN B08C STR\$ VAL B3BD ASC **B39B** CHR\$ B2FC LEFT\$ B310 RIGHT\$ **B33C B347** MID\$ B46F B458 **B5FE B6CD** BAB6

These subroutines are available to use if you are into machine code programing. Mr. Murphy is wrong: OSI users are not disinclined to explore their machines. The problem, until now, has been that too lit-

tle information was available. So let's dig into OSI's BASIC and publish a complete memory map similar to those already out for the PET and APPLE.

#### **BASIC Program**

10 0-1-D-41000
10 Q=1:D=41092 20 FOR C=40960 TO 41060 STEP 2
25 IF C=41016 THEN Q=0:D=41237
25 IF C=41016 IREN Q=0.D=41257
30 X=PEEK(C+1):GOSUB 500
40 X=Q+PEEK(C):GOSUB 500
50 PRINT" ";
60 X=PEEK(D)
70 D=D+1 80 IF X<128 THEN PRINT CHR\$(X);:GOTO60
90 X=X-128
100 PRINTCHR\$(X)
110 NEXT C
115 D=41224 120 FOR C=41062 TO 41074 STEP 3
120 FOR C=41062 TO 41074 SIEF 5
130 X=PEEK(C+2):GOSUB 500 140 X=1+PEEK(C+1):GOSUB 500
150 PRINT" ";
160 X=PEEK(D)
170 D=D+1 180 IF X<128 THEN PRINT CHR\$(X);:GOTO 160
190 X=X-128
200 PRINT CHR\$(X)
210 NEXT C
220 END
500 REM PRINT SUB
510 H=INT(X/16)
520 L=X-16*H
530 IF H<10 THEN H=H+48:GOTO 550
540 H=H+55
550 IF L<10 THEN L=L+48:GOTO 570
560 L=L+55
570 PRINT CHR\$(H); CHR\$(L);
580 RETURN
JOU IMICIAN

#### Would you like to become a Micro Dealer?

MICRO is a quality 6502 magazine. Our current dealers report that having MICRO available for sales in their stores actually helps sell 6502 based systems.

We require a minimum quantity of only 10 copies per month, and we offer a standard trade discount

Other items available are the Best of MICRO, Volumns 1 and 2, and All of MICRO.

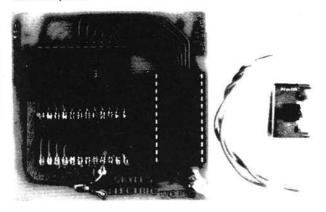
If you are interested in becoming a MICRO dealer, please write to:

MICRO Box 6502 Chelmsford, MA 01824



# Socket 2 Me™!"

From the original producer of peripherals exclusively for PET lovers everywhere . . . the device that allows you to select between the BASIC Programmer's Toolkit and the Commodore Word Processor II while they occupy the same address space.



The Socket 2 ME . . . doubles your memory expansion in a single socket. It's a 2.5" x 2.75" board that fits neatly into the Toolkit/Word Processor socket on the main logic board of all new PETs. Then both the Word Processor and the Toolkit plug into the Socket 2 ME.

A miniature slide switch - part of the kit - mounts with double-stick tape (supplied) to the front part of the right side of the PET base, almost hidden by the overhang of the top of the PET cabinet. The slide switch is connected to the Socket 2 ME by a special cable (also supplied) . . . and you're up and running.

Up and running; installation took only a minute or so. Flip the switch from Toolkit to Word Processor. And back. No need to open the PET.

Complete with the first-rate installation and operating instructions you've come to expect from all Skyles documentation.

#### YOU HAVE AN ORIGINAL 2001-8 PET?

No problem. The Socket 2 ME interfaces with the BASIC Programmer's Toolkit model TK 160E or TK 160S connector board, the Word Processor II interfaces with the Socket 2 ME, the slide switch is placed on the PET base. Then, as long as the PET 2001-8 has at least 8K of memory expansion, the system is up and running.

#### YOU HAVE A COMPUTHINK DISK AND YOU WANT A BASIC PROGRAMMER'S TOO LKIT?

How would you like to switch between the Computhink and the Toolkit with a single SYS command?

Just add two small jumpers to the Computhink system and a short program to the DOS diskette. Plug in the BASIC Toolkit TK 80E, enter the SYS command and your system is up and running.

#### NO ROOM ON THE PET 2 COMPUTHINK DISK BOARD?

All your sockets are booked? Fret not; Skyles comes to the rescue. Skyles Electric Works now has a modified EPROM board available with sockets for the Toolkit and Computhink ROM chips. Plug in the ROMs, add a jumper (supplied) to the PET 2 Computhink disk board, plug the new EPROM board into the Computhink disk board. Power up and enter a short switching program into the DOS diskette. Switch between the Computhink disk and the Toolkit-with a single SYS command.

#### ORDER NOW - with Skyles' 10-day money-back guarantee:

Socket 2 ME: \$\$22.50\*

Commodore Word Processor II: \$100.00\* Commodore Word Processor III: \$200.00\* BASIC Programmer's Toolkit

Model TK 80ED

\$85.00\*

Model TK 160ED

\$95.00\*†

\*Add \$2.50 to each for shipping and handling.

tNote: If Computhink EPROM board is returned, after purchase of TK 160ED, Skyles will refund \$20,00. "Socket 2 ME" is the trademark of Skyles Electric Works.

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



231E South Whisman Road Mountain View, CA 94041 (408) 735-7891

Skyles Electric Works

# Now You Can Have

# INPUT/OUTPUT

For Your Apple Computer

DUAL AXIS JOY-STICK AND PUSH-BUITTON MOD-ULE for Games, Graphics, and Experimental Program Input



microstick.

EXTRA LONG, HEAVY DUTY cables and connectors

MICROBOX AND MICRO-STIK PROVIDE APPLE OWNERS WITH THE HARDWARE TO EXPLORE THE INPUT/OUTPUT CA-PABILITIES OF THEIR COMPUTERS.

A SIMPLE COMMAND from the Apple keyboard or a Basic Program can switch an external device. Connect AC loads, such as lamps, motors, relays or solenoids directly through the MICROBOX's 4 AC OUT-LETS. Loads can range from 0 to 220VAC and draw up to 200 Watts each. Solid State Switching ISOLATES the load from your Apple for complete safety. Four LEDs provide a visual on/off status of each load.

A Complete Instruction/ Tutorial Manual is included with the MICROBOX.



#### **REAL-TIME INPUT**

The MICROSTIK is a sturdy, two axis joystick. Metal Cable Connectors assure trouble free usage over time, and enable extension cables to be added easily. Use the MICROSTIK to add real-time input to your game, graphic or experimental programs. Each MICROSTIK contains a PUSHBUTTON for added input possibilities.



MICROBOX and MICROSTIK sit comfortably on, or aside the Apple Computer. They have been designed to match the Apple in color and design.

#### ORDER TODAY AND CON-NECT YOUR APPLE TO THE OUTSIDE WORLD.

CONNECTS DIRECTLY

to the Apple Game Socket. ACCEPTS 2
Dual Axis MICROSTIKS

> ISOLATED SWITCH-ING of 4 AC loads or relays from a basic program. 4 LED status indicators.

Toggle Switch input

or 4 Paddles

The MICROBOX and MICRO-STIK can be purchased at most computer stores or can be ordered directly by mail or through our convenient 24 hour telephone service.

#### TELEPHONE: (703) 471-4291

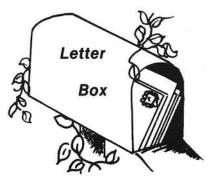
Order the MICROSET and receive the MICROBOX, 2 MICROSTIKS, the Manual and Cassette, and SAVE \$25.

MICROBOX	\$109.9
MICROSTIK (each)	34.9
Demo Cassette	9.9
MICROSET	164.9
12 ft. Ext. Cables	6.9
Relay Modules	WRITI
Solenoid Modules	WRITI

Va. residents add 4% sales tax

#### MASTER CHARGE, VISA accepted NO C.O.D.s

CJM Industries, Dept. MB 316B Victory Drive Herndon Industr Park Herndon, Va. 22070



Dear MICRO magazine,

My Dad and I have had an APPLE II for about 9 months. During this time I have learned of the special joys and sorrows that only computer people can appreciate or experience. This poem was born out of long hours at the keyboard. I hope you like it and feel that it is worth publishing.

#### Ode to My Disk

I always see verses praising the Apple But who sees the time saved by Disk II, it's ample? While Apple sits waiting to digest the data That trusty old "breadbox" spins round withoutbreakdown.

It hasn't been long since I've bought my Disk II But I know it's worth it and so do you.

I tried Panasonics and Hitachis too Resetting and loading my Apple I'd do. Frustrating it was and my hair I did pull

So soon I did tire of ERR MEM FULL.

So now my Hitachi sits dusty and wan And softly clicks Disk II, no ERR coming on.

Donna Marie Andert Connelly High School Anaheim, CA 92801

Dear Editor,

Most articles that are submitted to MICRO are claimed by their authors to execute correctly. The following program has been extensively de-bugged and is guaranteed to run *neither* on a PET *nor* on an OSI microcomputer.

10 FOR X = 1 TO 10 20 IF X = 5 THEN 40 30 NEXT X 40 REM 100 FOR Y = 1 TO 10 110 FOR X⁵1 TO 10 120 NEXT X 130 NEXT Y READY.

Can you figure out what is wrong here? If not, the answer is given in the next column

E.D. Morris, Jr Midland, MI 48640 This program was originally part of a 200 line game program with a "small bug." Through a bit of detective work, I narrowed the bug down to these eight lines. In the original game, these lines occurred in widely different sections of the program and appeared not to be related. When the program is executed, the computer will halt indicating "NEXT WITHOUT ERROR IN LINE 130".

This message is most confusing since line 100 clearly contains a "FOR Y". The program will run if lines 100 and 130 are deleted. Something appears to be wrong with the "Y" loop. If "X" is made the outer loop and "Y" is the nested loop, the program will run without error.

This is all a wild goose chase! Nothing is wrong with the "Y" loop. The first real hint of the cause is that replacing the variable "X" in lines 110 and 120 with a different variable, say "Z", solves the problem. The real culprit is line 20 where the program jumps out of a loop before finishing it. It is simple to see here in an eight line program, but not so obvious in a large program. The problem occurs when a variable from an unclosed loop is used again in a nested loop.

The moral of the story is to close loops whenever possible. For example, line 20 could have been:

20 IF X = 5 THEN Z = X : X = 10 : GOTO 30

If you can't close the loop, at least avoid using that variable in another loop.

And here is another poem from a reader, sent to us in May, 1979. We hope that he remembered to renew his subscription.

#### **End of Subscription**

There once was a town, Albuquerque, Wherein lived a genuine turkey Who, on learning his MICRO had died, Lost what little was left of his pride. Hadn't realized how close was the end. Still, he took out his pencil and penn'd "Mr. Tripp, won't you give me a chance? My check will disprove miscreance."

Nelson E. Ingersoll Albuquerque, NM 87110

We at MICRO would like to thank Donna, Earl, Nelson and all of our readers for their contributions. While all of the letters that we get are not as entertaining or as fun as these, they all certainly give us some things to think about. We welcome reader input and we encourage you to write to us with your comments, and suggestions at any time. We hope to run the Letterbox column in every issue, but it all depends on what we get from you.

The MICRO Staff

# Apple-Doc

By Roger Wagner

#### An Aid to the Development and Documentation of Applesoft Programs

This 3 program set is a must to anyone writing or using programs in Applesoft! It not only provides valuable info. on each of your programs, but allows you to change any element throughout the listing almost as easily as you would change a single line!!

With Apple-Doc you can produce a list of every variable in your program and the lines each is used on, each line called by a GOTO, GOSUB, etc., in fact, every occurance of almost anything!

You can rename variables, change constants and referenced line #'s, or do local or global replacement editing on your listing.

In fact, we guarantee that after purchase, if you don't feel APPLE-DOC is one of the most valuable programs in your library we will even refund your money! (Upon return of product.)

Unheard of? Yes! But that's how good APPLE-DOC really is!

That's not all!! Send for free info. or visit your nearest Apple dealer.

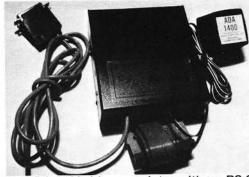
Only \$24.95 Please specify diskette or tape. (Calif. residents add 6% Sales Tax)

Available from your local computer store or:

Southwestern Data Systems P.O. Box 582-M Santee, CA 92071 (714) 562-3670

(Dealer inquiries invited)

#### **PET Printer Adapter**



CmC's ADA 1400 drives a printer with an RS-232 interface from the Commodore PET IEEE-488 bus. The ADA 1400 is addressable, works with the Commodore disk and prints upper and lower case ASCII

A PET IEEE type port is provided for daisychaining other devices.

A cassette tape is included with programs for plot routines, data formatting and screen dumps. The ADA 1400 sells for \$179.00 and includes a PET IEEE cable, RS-232 cable, power supply, case, instructions and software.

Order direct or contact your local computer store.

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.

CONNECTICUT micro COMPUTER, Inc. 150 POCONO ROAD BROOKFIELD, CONNECTICUT 06804 TEL: (203) 775-9659 TWX: 710-456-0052

# STONEWARE for APPLE II\*

# For the Serious Business or Home User: MICRO MEMO

MICRO MEMO is the first sophisticated "Desk Calendar" program to make good use of your computer's power.

- Micro Memo includes one time, weekly, monthly, semi-annual and annual reminders,
- $\bigstar$  Monthly reminders may be for fixed or "floating" dates (ex. 1st Saturday of every month).
- ★ Each reminder allows choice of one week, 2 week or 1 month advance notice—reminds you ahead of time to prepare for meetings, purchase tickets, make reservations, etc.
- Micro Memo includes "shorthand" for fast memo entry, greater capacity.
- \* Micro Memo will display or print any day's or week's reminders
- ★ Micro Memo is a "perpetual" calendar—automatically creates new months with all appropriate memos (birthdays, anniversaries, monthly meetings, etc.) as past months are dropped—system holds full year's reminders on one disk.
- \* Micro Memo "knows" most major holidays.
- \* Supports Mountain Hardware clock (optional).
- \* "Bomb Proof" menu driven command and data entry.
- \* Requires 48K, disk, RAM or ROM Applesoft.

\$3995

#### STONEWARE

Microcomputer Software P.O. Box 7218, Berkeley, CA 94707 (415) 548-3763

# And Just for Fun: TRANQUILITY BASE



TRANQUILITY BASE is a fast high resolution Lunar Lander game by Bill Budge, creator of Apple's "Penny Arcade" TRANQUILITY BASE is just like the popular arcade game, including multiple moonscapes, craft rotation, and zoom in for a close-up view as you approach the Lunar surface.

TRANQUILITY BASE requires 32K and disk

Available at your favorite computer store or direct from STONEWARE (add \$2 shipping & handling; Calif. residents add sales tax. Visa & MasterCharge accepted, no C.O.D.'s).

DEALER INQUIRIES INVITED

#Apple II is a Trademark of Apple Computer, Inc.

# The MICRO Software Catalogue: XIX

Mike Rowe P.O. Box 6502 Chelmsford, MA 01824

Name:

Dakin 5 Programming Aids II

System:

Apple II 48K

Memory: Language:

Assembler/Ap-

Hardware:

plesoft II Apple II, 2 Disk II's,

and Printer

Description: Set of seven programs: 1) Copier- copies absolutely any kind of file or program from one diskette to another. 2) Variable Cross Reference- creates a cross-reference for all variable neames used in an Applesoft BASIC program, showing all line numbers where a given variable name is used. 3) Line Cross Reference- creates a cross-reference for all referenced lines in an Applesoft BASIC program, showing where a given line is referenced by GOTO, GOSUB, THEN, or LIST statements. 4) Patcherallows the user to display any sector of a given file or program, and then to update any data within that sector. A second option enables the user to specify the particular sector he wishes to update. 5) Screen Printer - permits contents of the screen to be sent to the printer at any time the keyboard is active. The progrm remains in effect until you press RESET or reboot the system. 6) Array Editor- a simple word processor that allows you to create, modify, print and save your own text files. 7) Calculator II- a multiplication/division subroutine that handles numeric string data. Written in Assembler code, and using twenty place accuracy, it runs much faster thatn an equivalent BASIC subroutine. It is also compatible with the addition/subtraction subroutine, the Calculator, included in the first Dakin5 Programming Aids package reviewed in the December 1979 issue of The MICRO Software Catalogue XV.

Copies: Price:

Just released \$49.95

Includes:

Professionally

tion and program diskette.

Dakin5 Corporation (developer of The Controller for Apple Computer, Inc.)

bound documenta-

Local Apple Dealers

Name:

Available:

Author:

Page Format TTY

IN/OUT

Apple II System: Memory: 300.3FF (256 Bytes)

Language: Machine

Hardware:

Game Conn to TTY

Description: Program to output to and input from ASK 33 or 35 Teletype. Gives multiple kine feeds at end of each page and waits for you to tear off roll paper or insert new sheet for neat listings. Uses game connector.

Copies: Price:

Just released

\$2.00 Includes: Listing and Instruc-

> tions Ken Ellis

Author: Available:

Ken Ellis

R.D.8 Box 344

York, PA 17403

Name:

HI-RES GRAPHIC CHARTS GENERATOR

System:

APPLE II, APPLE II **PLUS** 

Memory: Language: 32K without ROM card, 16k with card APPLESOFT

BASIC

APPLE II, Disk II Hardware

(allows optional

features)

Description: This program will allow you to generate HI-RES graphic charts, either through keyboard or text file input (if using disk). 'Y' axes will be automatically scaled with values. 'X' axes will be marked for plotting points. Best of all, once graph is automatically created, you can add your own titles, comments, or symbols anywhere on the graph. Both upper and lower case characters are provided. Over 30 special symbols are included. Provisions are also included for multiple graph overlays. Disk II users can automatically have graphs made from existing data already stored.

Copies:

Price:

Just releases, 42 copies already sold. \$19.95 + \$1.25 for

postage.

Includes:

Cassette containing program, instructions on unique uses. Please specify when your order, if you have

ROM card or not. Les Stubbs

Author: Available:

Les Stubbs 23725 Oakheath Pl.

Harbor City, Ca.

90710

Name:

General Ledger Ver-

System:

sion 2.0 Apple II 48K

Memory: Language:

**Applesoft** 

Hardware:

Dual Drives, Any

Printer

Description: General Ledger Version 2.0 - This program is a complete doubleentry accounting system. User defined flexibility allowing up to 9 individualized departments in all Financial Reports. 10 levels of subtotals throughout each report gives more detailed Financial Statements. Using 5" drives, storing the entire Chart of Accounts and/or all posting approaches minicomputer times when verifying account numbers or sorting records. High-speed printer routines will process 1,000 postings into 70 accounts in less than 30 minutes. Using 8" drives, high-speed sorting routines requiring no additional disk work space and fast binary searching techniques allow data files to be limited only by your available disk space. Compatible with any printer and printer interface.

Copies: Version 1.0, 200; Version 2.0, Just released.
Price: \$180.00
Author: David A. McFarling
Available: Small Business
Computer Systems
4140 Greenwood

Name: VOCAB 1.1
System: APPLE II or APPLE II PLUS
Language: Applesoft
Memory: 32K

Hardware:

System:

Memory:

Lincoln, NE 68504

APPLE II and DISK II

Description: A vocabulary builder with over 1200 multiple choice questions allows the user to select either synonyms or antonyms. Intended as study aid for college board type exams (e.g., SAT, ACT, GRE, LSAT, etc.). Editor is included for expanding or modifying data lists. Several test formats with grading are options. Ideal for students with little computer experience.

Price: \$15.00
Includes: User documentation and diskette
Author: Steven M. Sliwa
Available: Sliwa Enterprises
257 C Clemwood Parkway Hampton, VA 23669

Name: SORT

Language: 6502 Machine
Language
Hardware 16K/32K PET, any
APPLE

Apple

PET, APPLE

32K/16K PET; any

Description: SORT is a 6502 machine language intelligent sort for commercial applications. Requires almost no user set-up when default values are used. Sorts integer, string and floating point arrays of more than one dimension with up to 20 sub-sorts-on-match (if needed).

Copies: Just released
Price: depends on end use
Author: David B. Black
Available: MATRIX SOFTWARE
INC.

1041 N. Main St. Ann Arbor, MI 48104 parison Apple II or Apple II Plus 32K with ROM Applesoft, 48K with RAM Applesoft ROM Applesoft.Can be used with RAM Applesoft by relocating above HGR2 display area not using graphics display feature. 1024 bytes of Machine Code is loaded before Main program. Cassette tape. Program supports Printer but driver

Investment Com-

gram supports
Printer but driver
subroutine not included. Apple II.

Description: We are often faced with decisions such as 'which of two investments is best?' This program provides a means of comparing them by the use of "Cash Discounting." Cash Discounting is a technique that is used to take into consideration the effects of inflation. Often we are faced with a decision of 'buying now' vs waiting a few years or paying cash vs time payments. The effects of inflation are not easy to quantify without some form of computer analysis. For each of two alternatives, entry include:

1. Inflation Rate for both

Name:

System:

Memory:

Language:

Hardware:

2. Initial Investment \$3. Number of years to salvage point and

value at that time.

4. Monthly expenses (or income)

5. Adjustments on an annual basis for the monthly expenses. This provides a means whereby you can make expenses track at a different rate than inflation.

Display is in a form of a 'cash Flow' by year, and a graphical presentation is also provided. The graphics have labels.

Just released Copies: \$16.95 Price: Includes: Cassette, loading instructions, description, and example. Author: Neil A. Robin **TECH-DIGIT** Available: 21 Canter Lane Sherwood, OR 97140 The Life Dynamic Name:

Transformation Experience
System: Apple II
Memory: 48K
Language: Applesoft and Machine Language
Hardware: Apple II Plus, Disk II

Description: Unique! This program is designed for all those people who desire to experience self-transformation, lifeawareness, making relationships work, and "getting your act together," but do NOT desire to pay est or Lifespring or any of the other "trips" of the Human Potential Movement, \$300 or so. Includes game playing as a means to a fun way of increasing awareness.

Copies: Many Price: \$15.95

Includes: (disk) w/instructions
Author: Avant-Garde Creations

Available: Avant-Garde Crea-

tions P.O.Box 30161 Dept. MC

Eugene, OR 97403

Name: I CHING

System: Apple II or Apple II Plus

Memory: 16K

Language: Integer Basic or Applesoft (please

specify)

Hardware: Cassette or disk

Description: Have your own oracle in your home. Consult the I Ching as others have through the ages. Includes a tutorial and a bibliography, as well as an interpretation of the results.

Copies: Just released
Price: \$9.95 on cassette;
\$14.95 on disk
C. Brandon

Author: Gresham, Jr.

Available: Ad Hoc Enterprises

23 Van Buren Street Dayton, OH 45402

Name: MUSIC

System: Any 6502 based system

Memory: 1.5K
Language: Assembly
Hardware: Terminal of

Hardware: Terminal or TVT and a speaker con-

nected to one output port

Description: Music is an interactive programming language for the creation of patterns of sound; "music". It is a compositional tool, not merely a music table compiler or piano roll type of program. Music's language structure is similar to "ROBOT" (see MICRO no. 10, page 15). Complex hierarchies of user defined functions - strings of musical events - which can be called like subroutines, allow the user to program highly intricate and surprising compositions.

Copies: Just released
Price: \$10.00 (KIM-1 Hyper-

tape cassette: \$3.00 extra)

Includes: User manual with

programming examples and a completely commented source and object code listing

Author: Michael Allen
Available: Michael Allen
6025 Kimbark
Chicago, IL 60637



Become a member of RAYGAMCO Computer Discount Club.



# **BIG SAVINGS ON EVERY ITEM!**



By being a RAYGAMCO Member you receive substantial discounts on every item you purchase, including all hardware, software, accessories, even books and paper! You will also receive a monthly newsletter with all the latest available for your particular computer system, and much, much more — exclusive to RAYGAMCO Members only!

All the famous brand names, including:

APPLE Alpha Micro Soroc

ATARI Alpha Pro Hazelti EXIDY/Sorcerer Cromemco Sektor

Kim/Commodore Xerox

Soroc Lear Siegler Hazeltine Shugart

Texas Instruments

SAVE 20% AND MORE!

#### Here's how to join.

Fill out the information, and mail. That's all there is to it. Nothing to buy. I want to be a RAYGAMCO Computer Discount Club Member. Please send my RAYGAMCO Membership card to:

Name		
Address		
City	State	Zip
Computer (Brand Na	ame)	
I would like informat	ion on (please specify system, p	art, accessory, book, program, etc.

WE HONOR VISA, MASTERCHARGE, BANKAMERICARD.

Store Hours: Sat 10-6, Sun 12-4, Tu-Fri 11-8

TOLL FREE, EXCEPT CA 800-854-6455

# RAYGAM, INC.

6791 WESTMINSTER AVENUE WESTMINSTER, CA 92683 TELEX 182274 (714) 891-2587

#### Ask for Instant Software at a computer store near you.

Computerland of Huntsville 3020 University Dr., Huntsville

#### Arizona

Ham Shack 4506-A N. 16th St., Phoenix Millets TV & Radio 621 East Broadway, Mesa

#### California

Byte Shop 8038 Clairmont Mesa Blvd... San Diego

Byte Shop of Mt. View 1415 West El Camino Real, Mt. View

Capital Computer Systems 3396 El Camino Ave., Sacramento Computers Made Easy 819 East Ave. Q-9, Palmdale

Computer World 6791 Westminster Ave., Westminster Computerland

16720 S. Hawthorne, Lawndale Coast Electronics

3118 No. Main St., Morro Bay Computerland

24001 via Fabricante No 904, Mission Viejo Hobbi-tronics

1378 So. Bascom Ave. San Jose Hobby World

19511 Business Ctr. Dr., Unit 6 Borthridge

Jade Computer Products 4901 W. Rosecrans, Hawthorne Opamp/Technical Books

1033 N. Sycamore Ave., Los Angeles

Radio Shack Dealer 8250 Mira Mesa Blvd., San Diego Santa Rosa Computer Center 604 7th St., Santa Rosa

Silver Spur Elect. Comm. 13552 Central Ave., Chino

The Computer Store 820 Broadway, Santa Monica

#### Colorado

3464 S. Acoma St., Englewood Computerland of North Denver 8749 Wadsworth Blvd., Arvada

The Computer Store 2300 Welton St., Denver

#### Connecticut

1439 Post Rd. E., Liberty Plaza,

#### D.C.

The Program Store 4200 Wisconsin Ave., N.W., Washington, D.C.

#### Florida

Adventure International 200 Bald Cypress Ct., Longwood **AMF Electronics** 11146 N. 30th St., Tampa

Computerland of Ft. Lauderdale 3963 N. Federal Hwy., Ft. Lauderdale Computerland of Jacksonville 2777-6 University Blvd. W Jacksonville

236 Talbot Ave., Melbourne Ukatan Computer Store Airport Rd., Destin Williams Radio & TV Inc 2062 Liberty St., Jacksonville

**Curtis Waters Enterprises** 

#### Georgia

Computerland of Atlanta 2423 Cobb Parkway, Smyrna

#### Hawaii

Computerland of Hawaii 567 N. Federal Hwy., Honolulu Radio Shack Assoc. Store 1712 S. King St., Honolulu

#### Illinois

Computerland 4507 North Sterling, Peoria Midwest Micro Computers, Inc. 708 S. Main St., Lombard

#### Indiana

Computer Center of South Bend 51591 US 31 North, South Bend

Memory Bank 4128 Brady St., Davenport

#### Louisiana

Computer Shoppe Inc. 3225 Danny Park, Suite 222, Metairie

#### Maine

Radio Shack 315 Main Mall Rd., So. Portland

#### Massachusetts

ComputerCity 175 Main St., Charlestown ComputerCity 50 Worcester Rd., Framingham Computerland of Boston 214 Worcester Rd., Wellesley Computer Packages Unlimited 244 W. Boylston St., West Boylston Lighthouse Computer Software 14 Fall River Ave. Rehobath The Computer Store 120 Cambridge St., Burlington

#### Tufts Radio & Electronics 206 Mystic Ave., Medford Michigan

Computer Center 28251 Ford Rd., Garden City Computerland of Grand Rapids

Computerland of Rochester 301 S. Livernois, Rochester Computerland of Southfield 29673 Northwestern Hwy., Southfield

Computer Mart 560 W 14 Mile Rd. Clawson

Hobby House 1035 W. Territorial Rd., Battle Creek

#### Minnesota

Computerland of Hopkins 11319 Hwy F., Hopkins Minnesota Software Inc. 5422 Fisher St., White Bear Lake

Dyer's, Inc. 200 E. Main St., West Point

#### Missouri

Computervan, Inc. 51 Florissant Oaks Shopping Center

Consolidated Software 16501 Greenwald Court, Belton

#### Montana

Intermountain Computer 529 So. 9th St., Livingston Personal Computer 121 Red Oak Dr., Carl Junction The Computer Store 1216 16th St. W. #35, Billings

#### Nebraska

Computerland of Omaha 11031 Flm St. Omaha Midwest Computer Co. Inc. 8625 I St., Omaha Midwest Computer Co. Inc. 4442 S. 84th St., Omaha Midwest Computer Co. Inc. 4403 S. 87th St., Omaha

#### **New Hampshire**

Bitsnbytes Computer Center 568 Pleasant St., Concord ComputerCity 1525 S. Willow, Manchester Portsmouth Computer Center 31 Raynes Ave., Portsmouth

#### · New Jersey

Computerland 35 Plaza Rte. #4, W. Paramus Computer Mart of NJ 501 Rte. 27, Iselin Dave's Electronics Pennsville Shopping Ctr., Pennsville

Lashen Electronics Inc. 21 Broadway, Denville Personal Computing Inc. 51 Central Sq., Linwood The Computer Emporium

#### Bldg. 103, Avenues of Commerce 2428 Rte. 38, Cherry Hill **New Mexico**

Antel Electronics Co 232 Wisconsin N.E., Albuquerque

Key Electronics

**New York** Aristo Craft 314 Fifth Ave., NYC Computer Corner 200 Hamilton Ave., White Plains Computer Era Corp. 1570 3rd Ave., New York Computerland of Nassau 79 Westbury Ave., Carle Place Computer World 519 Boston Post Rd., Port Chester Comtek Electronics, Inc. 2666 Coney Island Ave., Brooklyn Comtek Electronics, Inc. Staten Island Mall Store 220A, Staten Island Home Computer Center 671 Monroe Ave., Rochester

Schenectady Mr. Computer Imp. Plaza, Rte. 9, Wappingers Falls Softron Systems 308 Columbia Turnpike, Rensselaer The Computer Tree Inc. 409 Hooper Rd., Endwell

Upstate Computer Shop 629 French Rd., Campus Plaza **New Hartford** Ohio Altair Business Systems, Inc. 5252 North Dixie Dr., Dayton Astro Video Electronics 504 E. Main St., Lancaster Computerland 4579 Great Northern Blvd., Computerland 6429 Busch Blvd., Columbus Computerland 1288 Som Rd., Mayfield Heights Forbees Microsystems Inc. 35 N. Broad, Fairborn

call Toll-Free 1-800-258-5473.

Rhode Island Computer City 165 Angell St., Providence

Microcomputer Center 7900 Paragon Rd., Dayton

74 Robinwood, Columbus

1280 Aida Dr., Columbus

Oklahoma

Sounds Etc.

Oregon

Hyw. 33, Watonga

Pennsylvania

Audio Mart

Artco Elect

Shavertown

Vern Street Products Radio Shack Dealer 114 W. Taft St., Sapulpa

12020 S.W. Main St., Tigard

2151 Davcor St. S.E., Salem

518 Fifth Ave., New Brighton

Artco Elect. 302 Wyoming Ave., Kingston

Back Mountain Shop, Ctr.

Computer Workshoppe

2127 West 8th St., Erie

537 Easton Rd., Hersham

Personal Computer Corp.

Personal Computer Corp.

Erie Computer Co

Computer Pathways Unlimited, Inc.

3848 William Penn Hwy, Monroeville

Computerland of Harrisburg 4644 Carlisle Pike, Mechanicsburg

Mighty Byte Computer Center

24-26 West Lancaster Ave., Paol

Frazer Mall, Lancaster Ave., Frazer

Micro-Mini Computer World

Universal Amateur Radio, Inc.

Computerlab 671 S. Menden Hall Rd., Memphis H & H Electronics Inc. 509 N. Jackson St., Tullahoma

#### Texas

Computer Port 926 N. Collig, Arlington Houston Microcomputer Tech. K.A. Elect. 9090 Stemmons Frwy., Dallas Pan American Elect. Inc. 1117 Conway, Mission Utah

Quality Technology 470 E. 2nd So., Salt Lake City

#### Virginia

Home Computer Center 2927 Virginia Beach Blvd. Virginia Beach Southside Radio Comm. 135 Pickwick Ave., Colonial Heights

#### Washington

American Mercantile Co. Inc. 2418 1st Ave. S., Seattle Computerland of South King Co. 1500 S. 336 St., Suite 12 Magnolia Micro Systems 2812 Thorndyke Ave., Seattle Personal Computers S 104 Freva. Spokane Ye Old Computer Shop 1301 G. Washington, Richland Wisconsin

Byte Shop Of Milwaukee 6019 West Layton Ave., Greenfield **Petted Microsystems** 4265 W. Loomis Rd., Milwaukee

If the store nearest you does not stock Instant Software, use this order blank to purchase your software directly, or

ddress					
City		State	Zip		
			☐ AMEX ☐ Master Charge  Expiration Date		
igned			Date	-	
		Order your Instant Software	today!		
Quantity	Order No.	Program name	Unit cost	Total cost	
		Shipping and hand	ling	\$1.00	



Peterborough, New Hampshire 03458 603-924-7296

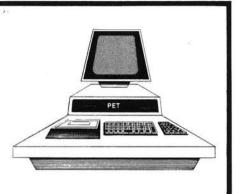
# Get a load of this . . Instant Software

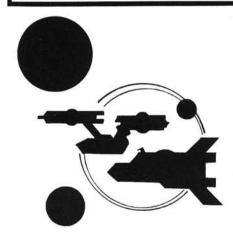
Ask for Instant Software at a computer store near you.

PET UTILITY I You're working under a serious handicap if you can't write programs in machine language. The PET Utility I package gives you the tools you need:

·Monitr - The Monitr program lets you write, edit, save, and verify any machine language and/or BASIC program. Just load and run the Monitr program and then load the program you want to edit.

• Programmer's Calculator - This program will convert numbers into the binary, octal, decimal, and hexadecimal systems and function as a floating point calculator. It will also display all four numbering systems simultaneously and allow you to handle large numbers limited only by the size of your screen. For the 8K PET. Order No. 0105P. \$7.95.





SPACE WARS You must protect your planet against a wide variety of alien attacks. In-

•Space War-One or two players can pilot their saucers and duel with laser beams or shoot stars.

.Star Ship Attack - Your mission is to protect the food station satellites from destruction by the enemy star ship. You must capture, destroy, or drive off the attacking ship.

•Battlefield - Guess the location of the four enemy divisions and destroy them before your forces are wiped out.

Engaging in battle requires the Apple 8K and Integer BASIC. Order No. 0096A \$7.95.

ACCOUNTING ASSISTANT This package will help any businessman solve many of those day-to-day financial problems. Includ-

will give you a complete breakdown of any

•Depreciation Schedule - You can get a depreciation schedule using any one of the following methods: straight line, sum of years-digits, declining balance, units of pro-

This package is available for both the PET and Apple. It requires the Apple 16K and Applesoft II BASIC. Order No. 0088A \$7.95 or the PET 8K. Order no. 0048P \$7.95.

the five different versions of this game. You must match the sequence and location of signals displayed by your Apple. You'll need an Apple with 24K and Integer BASIC. Order No. 0025A \$7.95.

HAM PACKAGE I This versatile package lets you solve many of the problems commonly encountered in electronics design. With your

Solve problems involving Ohm's Law, voltage dividers, and RC time constants.

· Dipole and Yagi Antennas - Design antennas easily, without tedious calculations. This is the perfect package for any ham or technician. Order No. 0054P \$7.95.

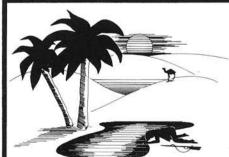


•Loan Amortization Schedule - This program loan or investment.

duction, or machine hours.

MIMIC Test your memory and reflexes with

8K PET, you have a choice of: Basic Electronics with Voltage Divider —



SAHARA WARRIORS Enjoy all the gritty realism of desert warfare:

·Commando - You must send your commandoes to trap a German general and cut him off from his troops.

•French Foreign Legion - The battalions of the French Foreign Legion are in a race with the Arabs. Which side will get its battalions into the oasis in the shortest time? You and a friend can find out.

Both programs in this package require an Apple 8K and Integer BASIC. Order No. 0080A

MORTGAGE WITH PREPAYMENT OPTION/ FINANCIER These two programs will more than pay for themselves if you mortgage a home or make investments:

• Mortgage with Prepayment Option - Calculate mortgage payment schedules and save money with prepayments.

• Financier - Calculate which investment will pay you the most, figure annual depreciation. and compute the cost of borrowing, easily and quickly.

All you need to become a financial wizard with a 16K Apple and Integer BASIC. Order No. 0094A \$7.95.

CHIMERA If you think the legendary Chimera was hard to handle, wait until you try this package. Included are:

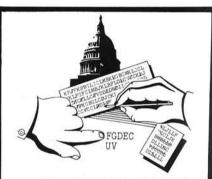
•Dropoffs - You must make your opponent's men "dropoff" the board by moving and firing your own men. For one or two players.

 Dots – Place your lines carefully as you try to build and capture the squares. For one

·Batter-up - You and another player take turns at bat as your PET becomes both the pitcher and the umpire. For two players.

•Reflex - Round and round the little white ball rolls. Only fast reflexes can guide it into the center of the maze.

You'll almost be able to feel the Chimera's fiery breath as you play the games on your 8K PET. Order No. 0110P. \$7.95.



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

•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.

Ask your local dealer for the latest Instant Software catalog or write: Instant Software Catalog Dept., Peterborough, N.H. 03458

Copyright 1980 by Instant Software Inc. Peterborough, New Hampshire 03458

All Rights Reserved



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

#### HUDSON DIGITAL ELECTRONICS INC.

# **ANNOUNCING THE HDE OMNIDISK 65/8**



Now, you can "plug in" the latest in a successful series of flexible disk systems developed by HDE for the KIM, SYM and AIM microcomputers. The OMNIDISK 65/8 is a complete system, using 8 inch soft sectored diskettes with a formatted (IBM Standard) capacity of 256K. Of course, a disk formatting function is included as are system supporting utilities for file renaming, disk packing, copy (dual systems) and others.

TED, a full featured, line oriented editor is standard in KIM and SYM based versions to get you up and running on your project in a hurry. The AIM version uses the on-board editor. With the OMNIDISK 65/8 you can con-

centrate on your problem, the disk supports you all the way.

OMNIDISK 65/8 is available in an attractive walnut wood cabinet, or unpackaged for OEM applications in dual and single drive configurations. The HDE disk controller is a state-of-theart 41/2" by 61/2" card electronically compatible with the 44-pin KIM-4 bus structure. The controller and disk-driver are designed to operate with the popular Shugart 801-R and compatible devices.

The OEM single drive is \$1195, the dual, \$1895 and the dual in the walnut cabinet, \$2200. Price is another reason to step up to the proven quality of an HDE system.

# HDE PRODUCTS – BUILT TO BE USED WITH CONFIDENCE AVAILABLE DIRECT OR FROM THESE FINE DEALERS:

# 6502 Bibliography: Part XIX

Dr. William R. Dial 438 Roslyn Avenue Akron, OH 44320

#### 581. Compute, Iss. 1 (Fall 1979)

Moser, Carl W, "Universal 6502 Memory Test," pgs. 32-33.

A memory test program with ways to adapt it to PET, APPLE II, SYM, KIM, TIM, OSI 65D, Western Data Systems,
ATARI, AIM, Super Kim, etc...

Thornburg, Katie A. and David D., "Flying with PET PILOT," pgs.40-45.

Kids and microcomputers at Peninsula School.

Tulloch, Michael, "CORVUS 11A Disc Drive," pg. 61.
How about a 9.6 megabyte memory on line? Compatible with the Apple DOS. Plugs into one slot of the Apple II.

Victor, John, "Atari Computers: The Ultimate Teaching Machines?," pgs. 62-64.

Discussion of the advantages of the Atari in educational

Lindsay, Len, "The Evolution of a Magazine," pgs. 65-66.
Discusses the history of the PET Gazette and its successor.

Butterfield, Jim, "PET in Transition," pgs. 68-70.
Discusses new modifications of the PET and how programs must be modified to accommodate the new systems.

Isaacs, Larry, "Retrofitting ROMS," pgs. 76-77.

How to replace the old PET Roms with the new units.

Malmberg, David, "Screen Print Routine," pgs. 78-79.
General utility to print the screen using the new PET printers.

Anon., "Cassette Format Revisited," pgs. 80-81.
Discussion of the PET Cassette format.

Butler, Brett, "Trace for the PET," pgs. 84-85.

TRACE allows you to see Basic executing, displaying each actual line as it is executed.

Hunkins, Arthur, "8-Bit Digital to Analog Converter," pgs.

A review of the MicroTechnology Unlimited DAC board for Hal Chamberlin's 4-part music program.

Stuart, Chuck, "Using Direct Access Files with the Dual Drive Disk," pgs. 93-96.

A tutorial on the Commodore 0240 Dual Drive Disk.

#### 582. Creative Computing 5, No 10 (October 1979)

North, Steve, "Mountain Hardware SUPERTALKER," pgs. 42-44.

A review of this new accessory for the Apple II.

Yob, Gregory, "Personal Electronic Transactions," pgs 180-183.

Changes in the New PET; more on the structure of Basic variables; Simple variables, PET Basic Pointers, Discussion of cryptograms.

#### 583. The Target (Sept/Oct 1979)

Roland, Don, "AIM 65 MOVIT," pgs. 2-3. A move program for the AIM 65.

Butterfield, Jim. "Mortgage," pg. 4. Mortgage program for AIM Basic.

Clem, Don, "Memory Display," pg. 5.

A program to show what the 6502 is seeing at the output port of the AIM micro.

Riley, Ron, "AIM BASIC," pgs. 6-8. All about AIM Basic.

Anon., "Statistical Analysis," pg. 10. Regression programs for the AIM.

#### 584. The Paper 2, Iss. 7 (September 1979)

Busdiecker, Roy, "Think Negative!," pgs. 3-7.

Negative numbers and subtraction are covered in this continuing tutorial on binary numbers, with examples for the PET and 6502.

Busdiecker, Roy, "Decimal to Binary," pg. 12. PET Basic program for converting Decimal to Binary.

Busdiecker, Roy, "Warning: Prevent Tape Decay," pgs. 21-22.
Use your PET tapes or at least rewind them occasionally to avoid print through on long undisturbed storage.

Oakes, Peter L. A., "PLOT 2-MP," pgs. 23-25. A plotting program for the PET.

#### 585. Call Apple 2, No. 7 (September 1979)

Wagner, Roger, "Speeding in Applesoft," pg.2. How to speed up Applesoft Subroutines.

Winston, Alan B., "The Multi-Lingual Apple," pgs. 4-5. All about Pascal for the Apple.

Hertzfeld, Andy, "Assembler Mini Reviews," pgs. 7-10.
A Consumer's guide to Apple II Assemblers: Apple Mini-Assembler, TED/ASM, Micro Products (Moser/Bishop) Assembler, Randy's Weekend Assembler, Aresco

Assembler, LISA, Microproducts 6-Character Label Assembler, E.A.T. (Apple II Text Processing System), 6502 Macro Assembler (Moser), UCSD Pascal Assembler, S-C Assembler, etc. etc....

Golding, Val J., "Peeking at Numbers," pg. 10.

Demonstration of a couple of ways to print numbers in Integer Basic larger than 32767 on the Apple.

Verlaque, Richard, "Decimal Division/Integer Basic," pg. 16. Arithmetic program for Integer Basic on the Apple.

Thing, Mike, "Applemash," pgs. 18-19.

All about Modem Operation and the newly established "Apple-Crate," the Call-Apple message system ABBS.

Hoyte, Jim, "Subroutine to Allow Prohibited Character in String Inputs," pg. 24.

Routine lets you use commas, etc. in string inputs on the Apple.

Williams, Rick, "Flash Cards," pg. 28. A Tutorial program for the Apple.

Capella, Mark, "Hide," pg. 29.

A program to hide program names from appearing in the catalog, but still permitting them to be recovered.

Golding, Val J., "Poke Chr\$ to screen," pg. 29.

Routine to put characters on screen at specified locations.

Dunmire, Darrell and Golding, Val J., "So Who Needs Applesoft?" pg. 31.

How to use string functions such as VAL, STR\$, and CHR\$ in Integer Basic on the Apple.

Neulen, Bob and Golding, Val J., "Change Catalog to CC," pg. 33

A routine to change catalog to CC for the 3.2 DOS.

Corsetti, Vincent S., "File Restore," pg. 35.

How to restore a program on a disk after you realize you just wiped out something you didn't mean to, on the Apple.

#### 586. Personal Computing 3, No. 11 (November, 1979)

Anon., "Jim and Kay Weir Re-invent the Wheel," pg. 14.
Discusses an application of an Apple II in the tire retreading business.

#### 587. Computer Cassettes Review, No 7 (Fall 1979)

Purser, Robert Elliott, "Software for Apple, PET and TRS-80." Software Directory, cassette reviews, game reviews, and a list of "The Classics—the best of the past."

#### 588. Rainbow 1, No. 8 (September 1979)

Ladbrooke, Brian D., "Running PIMS on the APPLE II," pg. 1. How to adapt this TRS-80 or PET program on Personal Information Management System to the Apple II.

Rost, Randi J., "HiRes Color," pgs. 4-7. A tutorial.

Heller, H. Lewis and Gothrie, Nelson, "Dollar Conversion," pgs. 7-11.

Three programs for the Apple to convert Applesoft outputs to rounded off dollar and cents values.

Buchen, Steven E., "Slow Down," pgs. 16-21.

Two programs to slow down the list function of the Apple

MacDougal, John, "Parallel Interface for the Apple," pgs. 22-24.

Hardware article on a parallel interface providing 2 output ports and 2 input ports.

#### 589. Interface Age 4, No. 12 (December 1979)

Gilbert, Betsy, "The Computerized Artist," pgs 72-73.

A description and evaluation of the Apple Graphics Tablet.

#### 590. Kilobaud Microcomputing No. 35 (November 1979)

Lancaster, Don, "Lower case for your Apple II, Part 1," pgs. 30-36.

Expand the usefulness of your Apple with this inexpensive addition.

Schmeltz, Leslie R., "The Apple Goes to Market," pgs. 70-76. Gather and analyze data from the stock market.

Derfler, Frank J., "Boy, Did I Make a Killing!" pgs. 112-114. Real Estate profit guide program for the PET.

Bajcz, Bill, "A 'Pentronics' System," pgs. 158-160. How to interface a PET to a Centronics 101 Printer.

#### 591. Southeastern Software Newsletter No. 13 (Oct. 1979)

McClelland, George, "Pascal," pgs. 1-3. Discussion of Pascal for the Apple.

Carpenter, Chuck, "Indexing Fundamentals, Part II," pgs. 4-6.

A tutorial on indirect addressing including a routine to read and print a memory range.

Anon., "Four HI-RES Programs," pgs 7-8.
Several hires routines to frame pages of a program.

#### 592. Ohio Scientific's Small Systems Journal (1979)

Anon., "OSI Programs."

A collection of about 60 programs for OSI computers.

#### 593. The Cider Press 2, No. 5 (September 1979)

Anon., "Disk of the Month," pg. 3. Seventeen more programs.

Anon., "Serial Handshake Modification," pg. 7.

The high speed serial interface card of Apple Computer Company can be made to run faster than 300 baud by simple modification.

Hockenhull, James L., "Better Sounding Apples," pg. 8. How to improve the sound of the apple by toggling the cassette output jack.

Nareff, Max J., "Crossfooting," pg. 8.
How to use crossfooting in tabulation of numbers.

#### 594. MICRO 17 (October 1979)

Connolly, Rick, "Nicer Writer," pgs. 5-6. Eliminate wraparound in your Apple.

Reynolds, William, "Disassembling the DOS 3.2," pgs. 7-10. Use the Apple DOS 3.2 more effectively with this information on its organization.

Scanlon, C. H., "Hooking PET to Ma Bell," pgs. 11-13.
Use your system as a terminal with an inexpensive modem.

Mimlitch, Thomas R., "Spelunker," pgs. 15-24. An adventure type game for the Apple.

DeJong, Marvin L., "6522 Timing and Counting Techniques," pgs. 27-39.

Application of the 6522 versatile interface adapter.

Chan Hark, "Card Shuffling Program for KIM-1," pgs. 41-42. Teach the Kim to shuffle the cards.

Hoyt, Bruce, "How Do You Connect Peripherals to Your Superboard II?" pgs. 43-46.

Some concise information on the configuration and use of the I/O Ports of the OSI Superboard II.

Rowe, Mike (staff), "The MICRO Software Catalog: XIII," pgs. 49-51.

Nine programs are reviewed, mostly for Apples.

Morris, E. D., "Hypocycloids," pgs. 52-53. A fast graphics program for the OSI.

#### **MUSICAL COMPUTER I AND II**

#### Learn How to Read Music!

Music lessons taught you in your home or at a studio cost from \$7 each half-hour and up! Now, available to you Apple II owners, is an opportunity for you to have your private music teacher in the comfort of your own

Written by a M.A. educator with over 20 years of music experience, this two-program cassette provides an alternative to music education!

These programs, utilizing high resolution graphics and a 32KApple II, will make music learning fun and enjoyable for the entire family!

Apple II is a TM of Apple Computers, Inc.

COMPUTER **APPLICATIONS TOMORROW** 

P.O. Box 7000-363 Palos Verdes, CA 90274

Address:..... City, State, zip.....

Send Check or Money Order for \$34.95 plus \$1.00 postage and handling. (CA residents add 6% sales tax)

#### OSI

#### OSI

#### SOFTWARE FOR OHIO SCIENTIFIC

Over 50 programs for C1, C2, C4 & Superboard, on tape and disk. All come with listings and compete documentation.

GAMES - 4K - Tape		UTILIT
CHESS FOR OSI -		CIP C
specify system	\$19.95	315.
STARFIGHTER	5.95	gives
Real time space war.		screen
SEAWOLFE	5.95	RENUM
Floating mines, three		SUPER
target ships, etc.		Has R
LUNAR LANDER	5.95	maker
With full graphics		
TEN TANK BLITZ	9.95	BUSIN
A sophisticated real time		SMALL
tank game.		Does p
8K GAMES		breake
BACKGAMMON	9.95	pages
BLACKJACK	6.95	STOCK
Plays all Vegas rules		
Add \$1.00 each for Color	/Sound	Keeps

CURSOR CONTROL

real backspace, one key clear, and midline editing MBERER 5.95 RUTILITY 12.95 Renumberer, Variable table and Search

\$9.95

L BUSINESS ANALYSIS 15.95 profit and loss, quick ratio, ven analysis and more. 13 of documentation. K PORTFOLIO

track of your investments

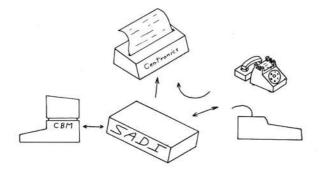
Our \$1.00 catalog has free game and utility listings, programming hints and a lot of PEEKs and POKEs and other stuff that OSI forgot to mention - and a lot more programs for sale.

DISKS 5" COLOR/SOUND \$29.95 DISK 1. STARFIGHTER, ROBO-TANK, SEA WOLFE, BOMBER, TEN TANK BLITZ DISK 2 BREAKTHROUGH, LUNAR LANDER, ALIEN INVADER, KILL ERROBOTS, SLASHBALL

#### **AARDVARK**

1690 Bolton, Walled Lake, Michigan 48088 • (313) 624-6316

# 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

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

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.

#### STOCK MARKET ANALYSIS PROGRAM DJI WEEKLY AVERAGE 1897-DATE

ANA1\* (ANALYSIS 1) is a set of BASIC Programs which enables the user to perform analyses on the Dow Jones Industrial weekly average data. From 6 months to 5 years of user selected DJI data can be plotted on the entire screen in one of 5 colors using Apples' High Resolution capabilities. The DJI data can be transformed into different colored graphic representations called transforms. They are: user specified moving averages; a least squares linear fit (best straight line); filters for time, magnitude, or percentage changes; and user created relationships between the DJI data, a transform, or a constant using +,-,x,/ operators. Colored lines can be drawn between graphic points. Graphic data values or their dates of occurrence can be displayed in text on the screen. Any graph or text can be outputted to a users printer. The Grid Scale is automatically set to the range of the graphs or can be user changed. As many colored graphs as wanted can be plotted on the screen and cleared at any time. The user can code routines to operate on the DJI/transform data or create his own disk file data base. ANA1 commands can be used with his routines or data base. An Update program allows the user to easily update the DJI file with current DJI weekly data

data.

The ANA1 two letter user commands are: CA = Calculate, no graph. CG = Clear Graphs, leave Grids. CK = Checking out program, known data. CO = Color of next graph (red, green, violet, white, blue). CS = Clear Screen. DL = Draw Line between points. Fl = Filter data for time, magnitude, or percent change. FU = Data, transform, or constant Function with +, -x, Operator. GD = Graphic mode, display all Graph Data on screen. GR = Graph data to screen. GS = Set Grid Scale. HE = Help, summary of any commands usage. LD = Load Data from disk file from inputted date to memory. LG = Leave Graphs, automatic Grid rescaling. LO = Look, select a range of the LD data and GR; All commands can now be used on this range. LS = Least squares linear fit of the data. MA = Moving Average of the data. NS = No Scale, next graph on screen does not use Grid Scale. NT = No Trace. PR = User implimented Printer routine. TD = Text mode, display Text Data on screen. TI = Time number to date or vice versa. TR = Trace. TS = Text Stop for number of lines outputted to screen when in TD. U1/U2 = User 1/2 implimented routines. VD = Values of Data outputted in text. VG = Values of Grid; low/high/delta. VT = Values of Transform outputted in text.

APPLE® II, 48 K, APPLESOFT ROM CARD, DISK II DOS 3.2 ANA1 DISK & MANUAL . . . \$49.95 (CA residents add 6% sales tax) GALAXY DEPT. A02 P.O. BOX 22072 SAN DIEGO, CA 92122

- \* Software Review in Call-A.P.P.L.E. (2/80): "An example of an excellent piece of software exploiting most of Apple II's major features." Overall Rating = 92.1
- \* Software Review in Apple Orchard (3/80): "A remarkably flexible approach to the analysis and plotting of any time series data." Overall Rating = 85.7

#### DISCOUNT DATA PRODUCTS

#### BASF 51/4" DISKETTES: \$34.50 PER BOX OF 10

HIGHEST QUALITY DISKETTES AT A BARGAIN PRICE! LABELS AND WRITE-PROTECT TABS INCLUDED.

#### VINYL DISKETTE HOLDERS FOR NOTEBOOKS

THE IDEAL WAY TO STORE DISKETTES. EACH VINYL PAGE HOLDS TWO DISKETTES AND INCLUDES A POCKET FOR EACH DISKETTE'S LABEL. SAFELY KEEP UP TO 40 DISKETTES IN A SINGLE 1" 3-RING NOTEBOOK!

\$4.95/SET OF 10

MARKETING YOUR OWN SOFTWARE? DDP OFFERS DEALER & SOFTWARE HOUSE DISCOUNTS ON NOT ONLY THE ABOVE ITEMS, BUT ALSO THE FOLLOWING PRODUCTS:

9" x 12" ZIP-LOCK BAGS FOR PACKAGING & DISPLAY OF SOFTWARE.

CORRAGATED MAILERS TO SHIP TO USERS OR DEALERS!

SEND FOR FREE INFORMATION AT: DISCOUNT DATA PRODUCTS

> P.O. BOX 19674-M SAN DIEGO, 92119

(ADD \$1.00 SHIPPING/HANDLING CHARGE TO ALL ORDERS.)

# 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

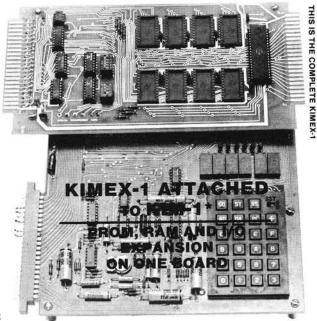
#### APPLICATIONS

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.....



PA RESIDENTS INCLUDE 6% STATE SALES TAX

P.O.BOX 207 ● BETHLEHEM, PA 18016



\$139.95
LIMITED TIME 1 K RAM FREE!!!

\* KIM IS A REGISTERED TRADEMARK OF MOS TECHNOLOGY, INC.

# Software for Apple II

for your Entertainment · Business · Education

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*<del>\*</del>

# Star Attractions:

+WRITE-ONI Professional-Word Processing lets you edit, move, delete, find, change and repeat any body of text, merge and save on disk. Does right-justified margins, centering, page numbering. You can enter name & address onto form letters when printing. Edit and merge any text disk file—even files not created by WRITE-ON—and spool text to disk for letter printing or editing. Chain up to 100 files in a single printer run. Needs Applesoft and 32K.

On Disk with operating manual \$99.50

FILEMASTER I Here's a powerful data file manager giving you two programs—FORMAT & RETRIEVAL. It handles everything from phone lists to legal abstracts. You can design your own data structure with up to 500 characters per record and up to 15 searchable fields in any combination. Needs 32K. On Disk \$49.95

FILEMASTER II Has all the same features of FILEMASTER I plus allows for totaling, advanced math routines, more powerful print formatting,

+ Space (Edu-Ware) Six programs form a unique epic game series.

Multi-faceted simulation of life in interstellar society. You and opponents
must make life & death decisions. Keeps track of your progress from
one game to next. Needs 48K and Applesoft ROM. Disk . . . . . . . . \$29.95

Pot D'Gold I or our All New Pot O' Gold II. A collection of 49 programs for 16K Apple. Everything from Logic to action games. Only a buck a game. Specify I or II.

Price each: Tape .......\$49 Disk ........\$54

32K Disk Inventory: Use stock numbers, description, vendor, record of purchase and sales date, amount on hand, cost & sell price, total value. Holds up to 300 items.

16K Memory Verify Diagnostic routine to check range of memory. Indicates faulty addresses, data in memory cell, and faulty data. Tape . . . . \$5





16K Appleadion Music synthesis composes original Irish jigs. Enter y	our
own music and save on tape or disk. Includes 3 Bach fugues.	
Tape	\$10

#### OTHER SPECIAL ITEMS FOR YOUR APPLE II

VERSAWRITER This digitizer drawing board, complete with a powerful software package on disk, lets you create any picture in color with high-resolution graphics. It's ideal for mass graphics. You can trace, edit, save and recall what you draw. It's a simple-to-use system for students, artists, engineers and graphic programmers. Has an 8½'x11'' working area. New applications added include: •Text Writer adds text to your pictures. You control size, color and direction of text; •Electronic Drawing lets you create schematics and includes commonly used symbols for transistors, OPAMPS, and FETS; •Distance/Area lets you compute distances on maps or area of any frame. Applesoft ROM and 32K required. \$249.95 Add \$5 (U.S.) or \$10 (Foreign) for shipping.

Don't see what you've been looking for, here? Then write for our FREE SOFTWARE CATALOG. We're saving one just for you!

To order, add \$2 (U.S.): \$5 (foreign) for shipping. California residents add 6% sales tax. Sorry, we cannot ship to P.O. Boxes. VISA/MASTER-CHARGE and BANKAMERICARD Welcomed!



Golden Plaza Shopping Center, Dept. 1A 9719 Reseda Blvd., Northridge, CA 91324 Telephone: (213) 349-5560

# What's NEW SOFTAPE

#### JOURNEY

You are about to embark on a very hazardous but profitable JOURNEY. The Apple is your eyes, ears, arms and legs. You

YOU ARE STANDING BEFORE A SMALL BRICK HITH 25 FEET OF BOSELL HAS AN OAK WINCH CLIMB DOWN ROPE MHEREOURS YOU SYNEST TO CLIMB?

can "GET" an object that is laying on the ground and you can travel North, East, South, West, Up and Down. You need to acquire tools as you JOURNEY forth and score precious points to become a GRANDMASTER JOURNEYER. JOURNEY REQUIRES 48K and loads on any

JAB-879 \$19.95

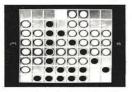
#### BASEBALL FEVER

Catch the fever with this ball game that never gets rained out. You are the pitcher, batter and manager for your team. As manager you control line-ups, pitching staff and base stealers. The optional sound effects enhance the colorful animation. When the game is over you can save the game statistics to tape or disk and you are on your way to a winning season. Requires 32K or 48K with disk. INTEGER BASIC

BFM-879 \$12.95

#### OTHELLO

Play a true competitor of this ancient game of territorial strategy. By flanking a line of the opponents men you "flip" them over



to your own color. Be cautious though, for OTHELLO will never say die until the last move. OTHELLO is available for both INTEGER BASIC and APPLESOFT, and loads in 16K.

OHS-279 \$14.95

#### **CONEY ISLAND**

Enjoy the excitement of an amusement park at home. CONEY ISLAND has 22 varieties of paddle games that are fast. Written in FORTH II for speed, and using the beautiful color graphics of the Apple, one or two can play the most exciting paddle games yet written. CONEY ISLAND can be loaded on any Apple. 16K

CIW-879 \$12.95

#### FORTH II

FORTH II is an extremely well documented version of the Forth language that has been in use since the late 1960's.

It is many times faster than basic and is easy to use. Many of its features are as follows:

#### **NEW FEATURES**

- \*RUNS ON ANY APPLE II COMPUTER
- \*(24K minimum)
- \*SUPPORTS DOS 3.2
- \*CONTROL C BREAK AND CONTINUE \*COMPATIBLE WITH AUTOSTART ROM
- \*"SAVE IT" FILE FOR CUSTOMIZING SYSTEM

#### STANDARD FEATURES

- \*INHERENTLY STRUCTURED LANGUAGE \*DISK BASED EDITOR AND COMPILER
- \*COMPLETE INSTRUCTIONAL REFERENCE MANUAL
- \*EXCELLENT EXECUTION SPEED AND MEMORY EFFICIENCY
- \*SUPPLIED ON MASTER DISKETTE
- \*VERBS FOR GRAPHICS, GAME I/O, SOUND, DISK AND TAPE I/O

FOG-279 \$49.95

#### PADDLE PLUS

If you have the same problem as Arnold Zieback with constantly changing paddles and PENS, then you too need PADDLE PLUS. This extender plugs into your game I/O port and is conveniently secured for easy access.

PPA-180 \$14.95

WHERE TO GET IT: Look for the SOFTAPE Software display in your local computer store. Apple dealers throughout the United States, Canada, South America, Europe and Australia carry the SOFTAPE Software line of quality products.

If your local dealer is sold out of SOFTAPE Software you can order it direct from us by check or Visa/Master Charge. If you have any questions please call us at:

VISA 1-213-985-5763



Or mail your order to the address below. We'll add your name to our mailing list for free literature and announcements of new products.

# SOFTAPE

10432 Burbank Blvd. • North Hollywood, CA 91601

#### **BRIGHT PEN**

What is the difference between a light pen and the BRIGHT PEN. Intelligent software and extensive documentation. The software will help you to calibrate your system for optimum operation. The documentation details the BRIGHT PEN DRIVERS and how they are appended to your INTEGER BASIC programs. BRIGHT PEN includes documentation booklet, two cassettes and, of course, the BRIGHT PEN.

BPE-279 \$34.95

#### **DUMP-RESTORE**

With DUMP-RESTORE you will be able to ba up your disk files to cassette and restore the disk. This allows you to relocate disk space for maximum efficiency and speed. The programs are saved and restored individually or the entire disk can be saved and restored. DUMP-RESTORE loads with INTEGER BASIC and requires 32K.

DRG-879 \$14.95

#### RESET GUARD

Tired of hitting reset by mistake? If so RESET GUARD will solve the problem. RESET GUARD is a hardware package that plugs directly into your Apple. It protects your programs because it will only Reset if hit twice in one second. Guard your Apple and your sanity with RESET GUARD.

RGA-180 \$34.95

