double columns. It is a little tricky to set up because the columns will not align evenly *unless* care is taken when placing carriage returns. It was not bothersome, however, and, after some experimenting, the text printed very nicely, each column having margin requirements pre-selected by the user.

On page 43 the automatic page numbering function is presented as CTRL+@. This combination does not work. It took a few minutes to locate the correct sequence for page numbering; it is SHIFT +@.

Its few disadvantages considered, Textwizard has a number of features which do set it apart from a simple text editor and turn it into a functional word processor.

One reviewer argued that the best single feature of the program is the Insert Text function. Because of the ease with which this works, there is little need for a lot of text moving. During this mode of operation, the operator is able to make insertions anywhere in the text without concern for erasure of previously written text. In addition, the screen border changes color during this operation, providing a constant reminder of the mode of operation. Wrap-around is maintained during the insert phase and this is a definite asset during text editing. Indeed, the ability to wrap-around text, thus keeping the text on the screen readable, is one of the really fine, and well-executed, features of Textwizard.

The use of changed screen colors and borders is a feature of Textwizard, which truly takes advantage of the versatility of the Atari, and helps to bring to this product an overall feeling of polish, while giving the user a very clear indication of the current mode of operation.

Print commands and formatting ability with this program are superb, if the Atari 825 or Centronics 737 printer is used in conjunction with the program. The ability to do multiple column printing is a great asset and it is here that Textwizard demonstrates a clear superiority over other word processors for the Atari. This feature alone would make the program worthwhile for anyone who publishes a newsletter.

Printing copy on cut (vs. continuous) paper is facilitated by the page eject and wait commands. The former performs a form feed, seeking the top of the next page. The latter causes printing to be suspended until a new sheet of paper is loaded.

The chaining feature is a powerful means to overcome the limitations imposed on the size of text files by the amount of available memory. With a 32K system, no text file may be greater than the equivalent of about 6.5 single-spaced pages. Sooner or later, most users will confront this ceiling, and will despair unless they see one implication of CHAIN: manuscript components (ranging from single characters to the largest amount of text permitted by memory) may be strung together with a command string in the first text file, resulting in the sequential printing of the whole.

The Final Overview

- **Panelist #1:**

"Overall, Textwizard is a very clean, useful word processor, delivering all that Datasoft says it will. It is easy to use and requires very little effort on the user's part to get excellent performance. It is also fast. The editing and searching functions are extremely swift and accurate. The chain command works well and facilitates printing and editing large blocks of text efficiently. The only two enhancements it could use (but doesn't need) are graphic display of the formatted page, and perfect spelling."

- **Panelist #2:**

"In short, Textwizard is a generally well-conceived word processing system for the Atari. Sometime between conception and delivery to the user, however, various gnomes intruded and left indelible marks on the product. Textwizard is well-suited for preparing term papers, inter-office memos, and informal personal correspondence. It may even be appropriate for the Great American Novelist. Professional technical writers and business executives will be happier and more productive with the much more powerful – and costly – word processing products that are targeted to their more complex needs."

- **Panelist #3:**

"With over fifty commands available to aid in editing, formatting, storing and printing text, Textwizard certainly provides the user with serious word processing capabilities. The program is well thought out; the formatting commands are simple and easy to use. Although a touch typist will probably have some small difficulty learning to use the extra keys with finesse, this is certainly not a drawback of the program. While certain portions of the program are weak...lack of menu and scrolling, and a very time consuming search and replace function, these are more than offset by the speed and ease of use which other areas of the program deliver to the user. All things considered, Textwizard, at a list price of $99.95, is a good buy and one which could be recommended to all Atari 800 owners."

Textwizard. Datasoft Inc., 19519 Business Center Drive, Northridge, CA 91324. $99.95. 32K and one or more disk drives and compatible with Atari 825, Centronics 737, and Epson MX-80. ©

Put Graphics Modes 1 And 2 At The Bottom Of Your Screen

R. Alan Belke
DeKalb, IL

Most of you who are regular readers of **COMPUTE!** are familiar with the mixing of the graphics modes ("Mixing Atari Graphics Modes," **COMPUTE!** #6). The only problem is that you can't use a mode past its regular range. That is, if you wanted to use Mode 1 past line 20 or Mode 2 past line 10, you couldn't. So you were stuck putting text you wanted at the top of the screen or in the text window. Until now, that is!

What's The Display List?

First we'll look at the Display List to see what it is and what it does. Figure 1 shows the Display List for Mode 3. You can verify this by running Program 1. Locations 560,561 contain the starting address of the list.

Figure 1.
112,112,112,72,112,158,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,66,96,159,2,2,2,65,78,158

The purpose of the list is to tell the computer how to display the information stored in the screen and/or text memories. Let's see how it does this. The first three bytes (112) set up the margin at the top of the screen. How they do this, I don't know. Anyone out there know? Next comes what I call an address byte, (72). In this case, a Mode 3 address byte. (Figure 2 shows what the address bytes are for each of the modes.) This byte pulls double duty. First, it sets the first line to Mode 3. Then it tells the computer that the next two bytes contain the address of the screen memory.

Figure 2.

MODE	0	1	2	3	4	5	6	7	8	
ADDRESS BYTE	66	70	71	72	73	74	75	76	77	79

The next 19 bytes (8) set one line each to Mode 3. I call these Mode 3 bytes. You get the value for these bytes by subtracting 64 from the address byte (72-64 = 8). From this, we can deduce that any byte with bit 6 on is an address byte. Also, notice that 19 Mode 3 bytes with the Mode 3 address byte give you 20 rows of Mode 3, which fills the screen up to the text window.

For whatever mode you are in, you will have 1 address byte and the number of rows, minus 1, regular bytes. For example, Mode 7 will have a Mode 7 address byte (72) and 79 regular Mode 7 bytes. Giving you 80 rows. To find out how many rows each mode has, check the "Table of Modes and Screen Formats." It's on the inside back cover of your *Basic Reference Manual.*

The Last Three Rows Of The Text Window

Now here's the important part. The next byte (66) is a Mode 0 address byte. But, instead of the next two lines containing the address of the Screen memory, they contain the address of the Text Editor memory. This is the start of the text window. Modes 1 through 8 use the Screen memory. Mode 0 uses the Text Editor memory. As you may have already guessed, the next 3 bytes (2) are Mode 0 bytes, giving us the last three rows of the text window. If we were in a full screen format, these last six bytes would not be here.

Now we are to the end of the list. This next byte (65) is also an address byte. But it has a special purpose. It tells the computer that it has reached the end of the list and that the next two bytes contain the starting address of the list. (The same as locations 560,561.)

Before we go on, let me say that the bytes that contain the addresses may vary, depending on the Mode you're in and on the amount of memory you have. All the other bytes will be the same.

So how do we get Modes 1 and 2 on the bottom of the screen? It's simple! Basically, all we do is change the Mode 0 bytes to Mode 1 or 2 bytes. Presto! The computer now displays the Text Editor memory in Modes 1 or 2.

Let's look at Program 2 to see how this is done:

Line 10: sets the margins to 40 characters per line and selects mode 3 with text window. Then it finds the address of the Display List.

Line 20: searches the list for the start of the text window.

for Atari*

A Financial Wizard

The Ultimate Personal Finance Management System

This new, comprehensive system enables you to determine how much you can afford to spend and what can be saved or invested. It graphically illustrates the answers to: "Where does the money go?" It also does a great deal more . . . nothing else even comes close!

Budget Projections

You are given 21 categories for which you enter a monthly plan and your anticipated income. The categories can be modified by name or amount at any time by you. The program prompts you every step of the way.

Check Entry

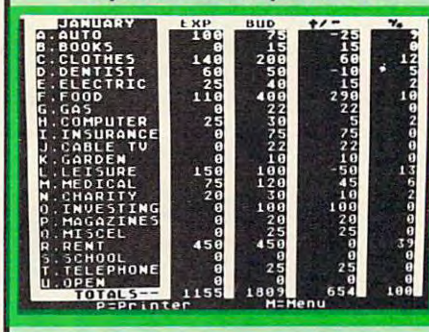
Enter data . . . month, check #, amount, y or n for tax deductibility, payee. Enter bank deposits and interest. Record up to ninety-six checks per month in amounts up to $99,999.00.

Check Search Menu

Your checks can be sorted and displayed in four different ways: Name, Category, Tax Deductibility, Number. This can be done on a monthly or annual basis. The checks, deposits and interest data are automatically filed for access by the Bank Statement Reconciliation Program.

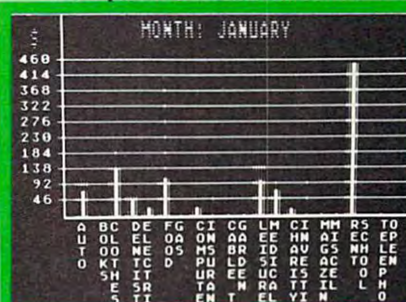
Check Search

Expense Comparisons



Detailed comparisons are displayed on a monthly or categorical basis which show ± and percentage relationships. You can then identify and analyze obvious and not-so-obvious relationships of the entered factors.

Graphic Presentations



Colored bar charts are generated rapidly via machine language comparing the amounts projected with actual expenditures. Chart displays are produced for both months and categories.

Also included in the system are:

- Printout options—all popular 40/80 col. printers.
- Multi-color graphics and sound enhancements.
- 4 record keeping categories for electric usage, gas consumption or whatever you wish to track.
- Check reconciliation-calculating totals and displaying # of checks and deposits outstanding.
- An audio/visual reward if your account balances.
- 8 utility programs.
- User-friendly operation and full documentation.

The system is designed for Atari* computers having a minimum of 24K and operating from a disk drive.

* trade mark of Atari, Inc. OK Residents Add 4 % Tax

The price of the system is only $49.95 plus $2.00 for handling/postage. If your local dealer does not have the Financial Wizard . . . Telephone orders are accepted on Mastercharge or Visa credit cards. Mail order must be accompanied by check or money-order or credit card #. Dealer inquiries invited. (405) 751-2783 From Computari.

Exclusively thru ON LINE COMPUTER CENTERS of OKC™ **10944-A North May Oklahoma City, OK 73120 (405) 751-2781**

☐ CHECK ENCLOSED ☐ VISA ☐ MASTERCHARGE

NAME (Print) ____________________

Address ____________________

City ____________ State ________ Zip ________

Card # ____________ Exp. ________ Sig. ____________

Line 30: changes the Mode 0 bytes to Mode 1 bytes.

Line 40-50: the periods denote spaces.

There are a few things of which to be aware. Even though you are using Modes 1 and 2, you're using the Text Editor memory; so the computer thinks in 40 column, not 20 column lines, which means two lines now equal one old line. Here is an example. Suppose we use an empty PRINT statement, planning to leave a blank line. Sorry, it won't work. We would have two blank lines. What we do is put 20 spaces in front of what we want printed on the second line. Also remember that we are using the Text Editor, so PRINT #6 will not work. Try some different things yourself.

What About Mode Two?

Well, that's almost as simple. Mode 2 lines are twice as wide as Mode 1 and 0; so there are only two combinations using Mode 2 possible: two rows of Mode 2 or one row Mode 2 with two rows of Mode 1. We can only use the same amount of room as was originally there. Program 3 uses the latter option from above:

Lines 10-20: same as Program 2.

Line 30: basically the same as in Program 2; only this time we make the second line Mode 2. And, since we use one less byte, we have to move the end of the list one location forward.

By now you should be able to change the text window into any combination of Modes 1 and 2 you want. If you have a program that would work better with the text at the bottom of the screen or the text window as Modes 1 or 2, get to work, experiment! Remember, you're the boss.

Program 1.

```
10 GRAPHICS 3:A=PEEK(560)+PEEK(561)*256
20 D=PEEK(A):? D;",";:IF D<>65 THEN A=A+
1:GOTO 20
30 ? PEEK(A+1);",";PEEK(A+2)
40 GOTO 40
```

Program 2.

```
10 POKE 82,0:GRAPHICS 3:A=PEEK(560)+PEEK
(561)*256
20 IF PEEK(A)<>66 THEN A=A+1:GOTO 20
30 POKE A,70:POKE A+3,6:POKE A+4,6:POKE
A+5,6
40 ? ".ATARI.AND.COMPUTE!.....AN.UNBEATA
BLE.."
50 ? "........TEAM..........FOUR.LINES.M
DE..1"
60 COLOR 2:SETCOLOR 1,10,6:PLOT 17,1:DRA
WTO 17,10:DRAWTO 9,18
70 PLOT 19,1:DRAWTO 19,18:PLOT 20,1:DRAW
TO 20,18
80 PLOT 22,1:DRAWTO 22,10:DRAWTO 30,18
90 GOTO 90
```

Program 3.

```
10 POKE 82,0:GRAPHICS 3:A=PEEK(560)+PEEK
(561)*256
20 IF PEEK(A)<>66 THEN A=A+1:GOTO 20
30 POKE A,70:POKE A+3,7:POKE A+4,6:POKE
A+5,65:POKE A+6,PEEK(A+7):POKE A+7,PEEK(
A+8)
40 ? ".ATARI.AND.COMPUTE!...1.LINE.OF.MO
DE.2."
50 ? "..2.LINES.OF.MODE.1"
60 COLOR 2:SETCOLOR 1,10,6:PLOT 17,1:DRA
WOO 17,10:DRAWTO 9,18
70 PLOT 19,1:DRAWTO 19,18:PLOT 20,1:DRAW
TO 20,18
80 PLOT 22,1:DRAWTO 22,10:DRAWTO 30,18
90 GOTO 90
```

©

Fantasy for your ATARI

Ali Baba and the forty thieves

By Stuart Smith

Guide your alter ego, Ali Baba, through the thieves mountain den in an attempt to rescue the beautiful princess. Treasure, magic, and great danger await you! One or more human players can guide up to seventeen friendly characters through the many rooms, halls, and caves. Some characters wander around randomly, making each adventure a little different.

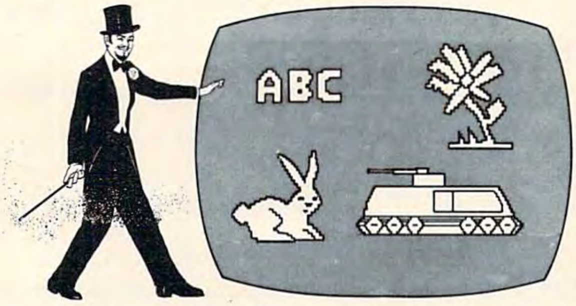
ALI BABA AND THE FORTY THIEVES is written in high resolution color graphics and includes music and sound effects. Adventures can be saved to disk and resumed at a later time. Requires 32K.

On Diskette Only — $32.95

Graphics for your ATARI

Character Magic

By Chris Hull



It's easy to create your own character sets and save them to diskette or cassette with CHARACTER MAGIC. But this is not just another character editor. CHARACTER MAGIC helps you use all the character types that the Atari is capable of, including descending characters (8 x 10 dots) and two types of five-color character graphics not supported by Atari's Operating System. Documentation includes examples of display lists that let you use these "secret" graphics modes. Requires 32K.

Cassette or Diskette - $29.95

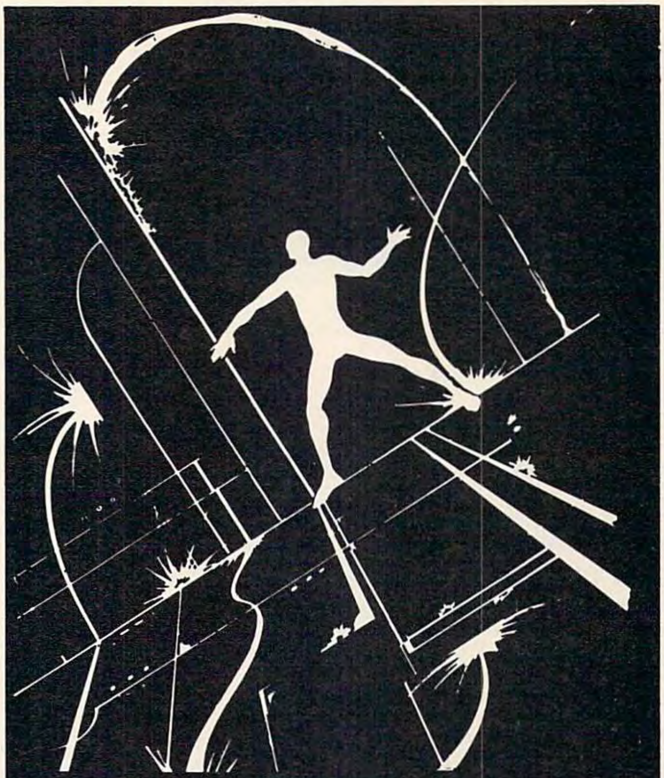
FOR OUR COMPLETE LINE OF ATARI SOFTWARE PLEASE WRITE FOR OUR CATALOG

ASK FOR QUALITY SOFTWARE products at your favorite computer store. If necessary you may order directly from us. MasterCard and Visa cardholders may place orders by calling us at (213) 344-6599. Or mail your check or bankcard number to the address above. California residents add 6% sales tax. *Shipping Charges:* Within North America orders must include $1.50 for shipping and handling. Outside North America the charge for airmail shipping and handling is $5.00. Pay in U.S. currency.

QS QUALITY SOFTWARE

6660 Reseda Blvd., Suite 105, Reseda, Ca. 91335 (213) 344-6599

*Indicates trademarks of Atari



MICROWORLD, an adventure within your computer, is available on the ATARI and the TRS-80. You are transformed into an electroid and must explore the circuits of your computer. Over 80 locations and many original problems exist within the maze of transformers and transistors. We dare you to explore the maze of bit cells! Each version of Microworld explores the workings of its respective computer. Atari or TRS-80 Microworld comes with a booklet defining terms and describing the function of the mystifying inner workings of home computers. Come face to face with a staticon! Explore the Microworld!

SATISFACTION GUARANTEED!

If for any reason you are not satisfied with our products, return your order within 14 days for a prompt and cheerful refund.

ORDERING INFORMATION

Orders are processed within five working days. Shipping and handling charge of $1.00 will be added to all orders within the U.S. and Canada. Overseas orders please add $3.00 for air post.

Atari Microworld Atari 400 and 800
32K Cassette $19.95
Atari 400 and 800
32K Diskette $22.95
TRS-80 Microrwold TRS-80 Model I and Model III
Level II 16K Cassette $19.95
TRS-80 Model I and Model III
Level II 32K Diskette $22.95

- MICROWORLD $ ______
- SHIPPING 1.00
- TOTAL $ ______

■ TRS-80 ■ ATARI
■ CASSETTE ■ DISKETTE

NAME ______
STREET ______
CITY ______ STATE ______ ZIP ______

■ CHECK ■ VISA ■ MASTERCARD

______ EXP. DATE ______

MED SYSTEMS SOFTWARE
P.O. Box 2674, Chapel Hill, NC 27514
1-800-334-5470

Atari PILOT At The Helm

Patricia Tubbs
Sunnyvale, CA 94086

As instructor for Gifted/Talented students in the Sunnyvale Elementary School District and educational computer consultant, I have had the opportunity of field testing Atari's PILOT language for the past year.

As a programmer in BASIC at an intermediate ability level, I've found Atari's PILOT extremely easy to use. Not only is text manipulation easily managed, but also both sound and high-resolution graphics are within the reach of beginners.

PILOT was originally developed by Dr. John Starkweather of the University of California Medical Center, San Francisco. It is a computer language which is word-oriented rather than number-oriented. People without any prior knowledge of computer programming find it easy and understandable to use in a very short time. The knowledge developed while using PILOT is a good foundation for moving on to other computer languages.

Curriculum-based Programming

ATARI PILOT makes preparing current curriculum-specific programs fairly easy. It has three modes of operation; they are: immediate mode, auto-number input mode, and run mode. When using the immediate mode your commands are executed immediately upon typing them and pushing the RETURN key. The auto-number input mode accepts PILOT statements, checks them for syntax errors and, if correct, assigns a number in sequence to each line and stores the statement in the program storage area. The run mode executes any program in the computer's memory.

With this language the programmer has the ability to control the appearance of words on the screen. In BASIC any print statement may appear to be spaced accurately and not divide words in inappropriate ways until that print statement is run. At that time the computer automatically divides any word at the end of a 40 space line. This is especially difficult if you wish to have a string variable (such as the student's name) inserted within that line. However, in PILOT, the computer will not break any words in a T: (type) statement, but will simply move them ahead to the next line. This feature is probably my favorite point for using PILOT when writing curriculum-based programs.

Another of the built-in features that is extremely helpful is the ability to renumber the program lines. The lines within the program may be renumbered by any increment starting at any number. This is especially helpful if you have created a particularly useful graphic design or musical piece and wish to use this module in some other program at another date. By renumbering the module to correspond to the new program, this module can become a part of the new program without the need of retyping it into the computer's memory.

The main text of your program is made with simple to use commands.

T: tells the computer to Type this on the screen.

A: the computer Accepts the user's input

M: matches the user's input with the programmer's expected answers.

C: Compute uses only integer arithmetic within the range of -32768 to 32767.

J:*LABEL allows the program to Jump to a module of the same name.

U:*LABEL allows Use of a module and then returns to the next statement following the U: statement.

***LABEL**
E: a module is created between these two commands.

GR: this command allows use of any of the various GRaphics capabilities.

SO: allows use of SOund.

PA: PAuses for a specific length of time.

Atari PILOT has been extended beyond PILOT's usual text and computational abilities to include Atari's capability for high-resolution graphics and sound. This graphics ability is called "turtle graphics," which comes to PILOT from the LOGO programming language, developed by Dr. Seymour Papert at MIT.

The programmer is able to control an imaginary robot called a "turtle" on the video screen. This turtle may be commanded to turn any number of degrees and to move forward any appropriate number of spaces. In doing so, it leaves a trace on the screen.

The full range of Atari colors is available for the turtle to use. However, you may use only three colors and the erase (or background color) at a time. After a figure has been drawn with the turtle, you may paint it by giving it a FILL command, at which time your figure is filled with color. (See

ALTERNATE REALITY SOFTWARE

Presents for the Atari

THE I CHING

THE ANCIENT CHINESE BOOK OF DIVINATION

- The complete text of the world's oldest book on disk
- 40K program
- 73 disk files (155,000 bytes)
- Occupying 1211 disk sectors
- High Resolution Graphics
- Animation
- Music and Sound Effects
- Instructional text material
- Calculates and Displays Hexagrams
- Displays Judgement, Image, Moving Lines for primary & secondary Hexagrams
- $44.95

order from:

Alternate Reality Software
2111 W. Arapahoe Drive
Littleton, Colorado 80120

Dealer inquiries invited
Atari is a T.M. of Atari, Inc.

ATARI® 400 AND 800™ COMPUTER OWNERS

AN OFFER YOU CAN'T REFUSE!

PDI produces some of the most innovative software for the Atari computer now available. In order to convince you, we have developed a sample cassette to show off our programming skills. The sample cassette has an animation-based educational program on side A, and a mixed graphics and voice demonstration on side B. Cost? Just $3.95. If you don't like the programs, simply erase and use the cassette as a blank! (The cassette is a superior quality, screw type C-20 with premium AGFA or BASF tape.)

If you accept our offer, we will credit $3.95 against any future order with our company. Just return the cassette with your order.

For the sample cassette send $3.95 to Program Design, Inc., 11 Idar Court, Greenwich, CT 06830. (Foreign orders include an extra $2.00 for postage.)

ATARI® is the registered trademark of ATARI, Inc.

SYNCRO, INC.

SOFTWARE DIVISION
31332 VIA COLINAS SUITE 107
WESTLAKE VILLAGE, CALIF. 91361
213-889-9508

SOFTWARE FOR ATARI T.M. 400/800

ALIEN HELL © 1982

ALL GRAPHIC ADVENTURE. CAN YOU FREE YOURSELF FROM ALIEN HELL? IN THE SPIRIT OF THE "MAZE OF DEATH".

JOYSTICK REQUIRED

16K Tape $14.95 — 24K Disk $19.95

MAR TESORO © 1981

A NEW GRAPHIC ADVENTURE, SIT BACK WITH YOUR JOYSTICK AND DIVE THE DEEP BLUE SEA. RENT RIG AND CREW TO PLUNDER THE STRANGE TREASURES THAT AWAIT YOU.

JOYSTICK REQUIRED

24K Tape $16.95 — 32K Disk $21.95

MAZE OF DEATH © 1981

ARE YOU A SUPERAGENT? A FAST ACTION, ALL GRAPHIC GAME WITH PLENTY OF SPECIAL EFFECTS IN A MAZE LIKE UNDERGROUND COMPLEX. FIVE LEVELS OF PLAY IN REAL TIME.

JOYSTICK REQUIRED

16K Tape $12.95 — 24K Disk $17.95

VISA AND MASTERCARD ACCEPTED.
CALIF. RES. PLEASE ADD 6% SALES TAX
PLEASE INCLUDE $1.50 SHIPPING
DEALER INQUIRIES WELCOME

Sunrise Software®

BUDGETMASTER ($34.95)

- is a professional budgeting tool geared to develop and maintain even the most complex budget. At the same time, the easy to read manual and the friendly prompts allow even the beginning budgeter without computer experience to develop a budget to meet their needs.
- includes the option for hardcopy printout of any of the 39 different displays as well as an option to save your data to cassette or diskette.
- requires 32K RAM

CHECKBALANCER ($16.95)

- although the perfect companion program to the Budgetmaster, stands alone as a great tool to balance your checking account(s), with a monthly breakdown of your spending pattern.
- allows you to store your spending analysis to tape or diskette, or use **Checkbalancer** in conjunction with the **Budgemaster**, and automatically rewrite the expense data of the latter program process.
- requires 16K RAM

BUY BOTH PROGRAMS TOGETHER FOR $39.95 AND SAVE $11.95!!!

Send check or money order plus $1.50 for shipping and handling and save C.O.D. fee (Sorry, no credit cards)

Ohio Residents Add Sales Tax

(DEALER INQUIRIES WELCOME)

Sunrise Software® • P.O. Box 25621 • Garfield Hts., Ohio 44125
(216) 581-9359

*Atari is a Trademark of Atari, Inc.

Program 1.)

When writing SOund components in your program you have the facility to command four voices (up to a four-note chord) between C below middle C to F# and C above middle C. This gives you a chromatic scale of 31 notes with C below middle C = 1 and F# above C above middle C = 31.

By including the PAuse command, one may hold each note or chord for a given length of time. The length of the pause is determined by the number of 1/60th of a second *ticks* selected (e.g. one tick equals 1/60th of a second, and 60 ticks equal one second). (See program 2 and Figure 1.)

The Atari joysticks and paddles can also be used within your program. You may use up to four joysticks in as many as 11 positions, with the joystick trigger having a two position value. Four pairs of paddle controllers with a range from 0 to 227 rotary positions may be used.

Another useful feature is the synchronization of an audio tape to the computer. One could use the audio portion of a tape to give instructions for a given program or perhaps give a spelling word orally. I see this as a particularly convenient aspect when writing programs for beginning readers or children with learning disabilities. While the audio portion is running, the computer can display information on the monitor simultaneously.

PILOT is an easy to use programming language, one I am sure that most educators will find very useful for curriculum development for their specific classroom needs.

Beatrix Potter On The Computer

Program 3 was written to use in the study of the literature written by Beatrix Potter. This program was used as a biographical introduction and follow-up lesson. With each question, several possible answers were displayed. If students typed in the incorrect answer, they were given some further information and returned to the original question. In this particular program, even wrong answers produced further learning experiences.

This shows a color picture on the TV monitor when a correct answer is typed: a bunny and a duck. To draw these pictures using the turtle, I utilized the instructions which I found in a crafts book for sewing them in cross-stitch embroidery. These instructions come printed on a grid which is similar to the imaginary grid on the computer's video monitor. By placing the turtle at the middle of the embroidery grid, you can have it move the appropriate number of squares, turn, and move again where needed.

With the PILOT cartridge, Atari has provided an exceptionally beautiful teaching guide for children. It is an easy to follow as well as aesthetically attractive manual which all children will enjoy using. Also included is a general manual, as well as a documented demonstration tape.

My fifth-grade students who field-tested Atari PILOT had no difficulty learning quickly to draw pictures, manipulate text, and add sound effects to their programs. I see this language as the first language that beginning programmers should learn. It is motivating and a good basis for learning other computer programming languages.

Figure 1.
PILOT Sound

Program 1.

```
200 *BLOCK
210 ,   R:THIS IS A SAMPLE PROGRAM OF
        BLOCKS STACKED ON ONE ANOTHER
220 ,      GR:GOTO-0,-30;CLEAR
300 ,      GR:PEN RED
310 ,      U:*SQUARE
350 ,      GR:GOTO -0,-14
360 ,      GR:PEN BLUE
370 ,      U:*SQUARE
380 ,      GR:GOTO-0,2
390 ,      GR:PEN YELLOW
400 ,      U:*SQUARE
405 ,      U:*SCALE
410 ,      E:
490 *SQUARE
500 ,      GR:4(DRAW16;TURN90)
550 ,      GR:FILL16
560 ,      E:
```

COMPUTE!'s Listing Conventions

Many programs which are listed in **COMPUTE!** use cursor control keys, color keys, and so forth. We have established a listing convention which we believe eases the task of typing programs in accurately.

Atari Conventions

For the Atari, all the editing and cursor-control characters are spelled out and surrounded by brackets: [CLEAR] for "clear screen." Other characters, such as CTRL-T (the "ball" character) will be listed as the "normal" character, but within brackets: [T]. A series of identical control characters will be indicated by a number within the brackets: [3 DOWN] means type the cursor-down key three times; [12 R] means type CTRL-R twelve times.

Two control characters, [=] and [–] should be shifted. Any reverse field text will be enclosed within vertical lines. (Press the Atari logo key [Atari logo] for each vertical line you see.)

Program 2

```
990 R: THIS IS THE ATARI MUSICAL SCALE
       WITH EACH NOTE HELD FOR 15/60TH
         SECOND
1000 *SCALE
1010 ,     C:#S=#S+1
1020 ,     SO:#S
1030 ,     PA:15
1040 ,     J(#S<31):*SCALE
1050 E:
```

Program 3.

```
10 T:

110 C:#H=1
120 *BORDER
130 C:#H=#H+1
140 T:

150 J(#H<09):*BORDER
160 T:                BEATRIX POTTER

170 T:

180 T:    ONCE upon a time there were fo
ur    little Rabbits, and their names
  \
190 T:  were _____
  \
200 T:                    Flopsy,
                Mopsy,
  \
210 T:          Cotton-tail,
          and Peter.
\
220 T:

230 T:

240 T:

250 PA:100
260 U:*MUSIC
270 R:THIS PROGRAM WAS WRITTEN TO BE USE
D AS AN AUTOBIOGRAPHICAL INTRODUCTION TO
 BEATRIX POTTER AND HER BOOKS.
290 R:  PATRICIA TUBBS, SUNNYVALE, CAIFO
RNIA
300 R: LINE NUMBER 970 SHOULD BE CHANGED
 YEARLY TO ALLOW FOR THE CORRECT    ANSW
ER.
310 GR:QUIT
320 T:What is your name?
330 A:$NAME
340 *FIRST
350 T:Who wrote THE TALE OF PETER RABBIT
?\
360 T: Beatrix Potter
   Mark Twain                          E
zra Keats
370 A:
380 GR:QUIT
390 M:BEATRIX, MARK,EZRA
400 JM:*BEATRIX *MARK *EZRA
410 *MARK
420 T:
430 T:Mark Twain wrote HUCKLEBERRY FINN
and other stories.  Please try again.\
440 J:*FIRST
450 *EZRA
460 T:
470 T:Ezra Keats wrote WHISTLE FOR WILLI
E and other stories.  Please try again.\

480 J:*FIRST
490 *BEATRIX
500 T:
510 T:Yes, Beatrix Potter wrote THE TALE
 OF PETER RABBIT and 22 more books for c
hildren like you and your friends.\
520 A:
530 U:*BUNNY
540 *SECOND
550 T:Why did Miss Potter write THE TALE
 OF PETER RABBIT?\
560 T:
570 T: just for the fun of it
   for the money                       a
s a get well letter
580 A:
590 GR:QUIT
600 M:JUST,FOR,WELL
610 JM:*JUST *FOR *WELL
620 *JUST
630 T:Yes, Miss Potter did enjoy writing
 stories, but that was not her only reas
on.  Try again.\
640 T:
650 J:*SECOND
660 *FOR
670 T:Miss Potter did not need the money
 because she was the daughter of a wealt
hy lawyer.  Please try again.\
680 T:
690 J:*SECOND
700 *WELL
710 T:Yes, she wrote this story as a get
```

```
 well letter for a young friend, 5 year
old Noel.  \
720 T:
730 A:
740 U:*BUNNY
750 *THIRD
760 T:Miss Potter lived in what country?
\
770 T:
780 T:      England
       Canada
    United States
790 A:
800 GR:QUIT
810 M:ENGLAND, CANADA, UNITED STATES
820 JM:*ENG    *CAN   *UNI
830 *CAN
840 T:
850 T:  Canada is an English speaking co
untry but not the home of Miss Potter.\
860 J:*THIRD
870 *UNI
880 T:
890 T: Both Ezra Jack Keats and Mark Twa
in are from the United States but Miss P
otter was not.\
900 J:*THIRD
910 *ENG
920 T:
930 T: Yes, she lived in England and THE
 TALE OF PETER RABBIT first appeared in
print in 1901.  \
940 *NUMBER
950 T:Can you tell your teacher and me h
ow long ago that was?  \
960 A:
970 M:80
980 TN: Try subtracting that one more ti
me.  Remember you put this year's number
 on the top with 1901 underneath.\
990 TY: My, that was a long time ago, wa
sn't it, $NAME ? \
1000 JN:*NUMBER
1010 A:
1020 U:*DUCK
1030 *FOURTH
1040 T:Why were the books published in t
he small size?      \
1050 T:
1060 T:  To fit small hands (or)
       The paper would cost less
1070 A:
1080 GR:QUIT
1090 M:HANDS,PAPER
1100 JM:*HANDS *PAPER
1110 *PAPER
1120 T: You're right, the paper would co
st less, but she had another reason for
wanting the books nice and small.\
1130 J:*FOURTH
1140 *HANDS
1150 T:Miss Potter told her printer that
 the books were to be small enough for l
ittle hands to hold, \
1160 T:and printed on 'stout' paper.  Th
e size she suggested was 5 inches by 3
3/4 inches. \
1170 A:
1180 U:*DUCK
1190 *FIFTH
1200 T:Was Peter Rabbit translated into
any other languages?
1210 A:
1220 GR:QUIT
1230 M:YES
1240 TY:Yes,Peter Rabbit has been printe
d in twelve languages including Afrikaan
s and Japanese as well as in\
1245 TY: Braille.
1260 TN:TRY AGAIN!
1270 JN:*FIFTH
1280 A:
1290 U:*BUNNY
1300 *SIXTH
1310 T:From where did the ideas for the
characters in Beatrix Potter's books com
e? \
1320 T:
1330 T:    other people
      pets and animals she watched
  children
1340 A:
1350 GR:QUIT
1360 M:OTHER,PETS,CHILD
1370 JM:*OTHER *PETS *CHILD
1380 *OTHER
1390 T:No not other people, try one more
 time, please, $NAME
1400 J:*SIXTH
1402 *CHILD
1404 T: Other children did often give he
r ideas; however, her main ideas came ot
her ways.
1406 J:*SIXTH
1410 *PETS
1420 T:Yes, as a child Beatrix owned man
y pets.  She took home wild, ill or hurt
 animals and nursed them back to\
1430 T: health.  Beatrix was the only da
ughter of wealthy parents who did not wa
nt her to go to public or private\
1440 T: school.  She had a tutor at home
```

```
, but no other children to play with, so
 she made friends with animals \
1450 T:  She spent much of her time sket
ching their pictures, which she used yea
rs later in her stories. \
1460 A:
1470 U:%DUCK
1480 U:%MUSIC
1490 U:%BUNNY
1495 T:Now it's time for you to read som
e of Beatrix Potter's books.  I hope you
 enjoy them, $NAME \
1500 E:
1510 %CHILD
1520 T:Sometimes children did give her i
deas.  She wrote many story letter to yo
ung friends including new sotries\
1530 T:expecially written for them.  But
, this was not wher most of her ideas ca
me.  Try again. \
1550 %BUNNY
1560 GR:DRAW8
1570 GR:TURN270;DRAW2
1580 GR:TURN90;DRAW3
1590 GR:TURN270;DRAW2
1600 GR:TURN90;DRAW2
1610 GR:TURN270;DRAW2
1620 GR:TURN90;DRAW3
1630 GR:TURN270;DRAW1
1640 GR:TURN90;DRAW3
1650 GR:TURN90;DRAW2
1660 GR:TURN270;DRAW3
1670 GR:TURN90;DRAW2
1680 GR:TURN 270;DRAW5
1690 GR:TURN270;DRAW2
1700 GR:TURN270;DRAW3
1710 GR:TURN90;DRAW2
1720 GR:TURN270;DRAW3
1730 GR:TURN90;DRAW2
1740 GR:TURN90;DRAW3
1750 GR:TURN270;DRAW2
1760 GR:TURN90;DRAW3
1770 GR:TURN270;DRAW2
1780 GR:TURN270;DRAW6
1790 GR:TURN90;DRAW2
1800 GR:TURN270;DRAW2
1810 GR:TURN45;DRAW3
1820 GR:TURN315;DRAW3
1830 GR:TURN270;DRAW3
1840 GR:TURN90;DRAW2
1850 GR:TURN270;DRAW2
1860 GR:TURN90;DRAW2
1870 GR:TURN90;DRAW2
1880 GR:TURN270;DRAW1
1890 GR:TURN90;DRAW2
1900 GR:TURN270;DRAW1
1910 GR:TURN270;DRAW2
1920 GR:GO2;DRAW2
1930 GR:TURN90;DRAW2
1940 GR:TURN90;DRAW3
1950 GR:TURN270;DRAW1
1960 GR:TURN90;DRAW1
1970 GR:TURN270;DRAW2
1980 GR:TURN270;DRAW1
1990 GR:TURN90;DRAW2
2000 GR:TURN270;DRAW1
2010 GR:TURN90;DRAW1
2020 GR:TURN270;DRAW3
2030 GR:TURN90;DRAW1
2040 GR:TURN270;DRAW9
2050 GR:TURN325;DRAW4
2060 GR:TURN270;DRAW7
2070 GR:PEN UP;DRAW-4
2080 GR:TURN135
2090 C:@710=12%16+3
2100 GR:PEN BLUE;FILL 50
2120 GR:PEN RED ;DRAW2
2130 T:Peter Rabbit really likes you,$NA
ME. \
2140 PA:400
2150 GR:QUIT
2160 E:
2170 %DUCK
2180 GR:PEN ERASE
2190 C:@710=15%16+4
2200 GR:PEN BLUE;TURN 55;DRAW8
2210 GR:TURN-55;DRAW6
2220 GR:TURN45;DRAW8
2230 GR:TURN45;DRAW2
2240 GR:TURN-90;DRAW2
2250 GR:TURN-90;DRAW4
2260 GR:TURN-45;DRAW10
2270 GR:TURN45;DRAW6
2280 GR:TURN-90;DRAW2
2290 GR:TURN90;DRAW5
2300 GR:TURN-90;DRAW2
2310 GR:TURN90;DRAW4
2320 GR:TURN90;DRAW2
2330 GR:TURN-90;DRAW2
2340 GR:TURN-90;DRAW2
2350 GR:TURN-90;DRAW2
2360 GR:TURN90;DRAW4
2370 GR:TURN-45;DRAW6
2380 GR:TURN-45;DRAW4
2390 GR:TURN-90;DRAW2
2400 GR:TURN90;DRAW6
2410 GR:GOTO15,16;PEN RED;DRAW4
2420 GR:TURN90;DRAW2
2430 GR:TURN90;DRAW4
2440 GR:TURN90;DRAW2;TURN180;FILL2
2450 GR:PEN BLUE
2460 GR:GOTO-10,1;TURN180;FILL5
```

```
2470 GR:GOTO-13,2;FILL 5
2480 GR:GOTO-10,-1;FILL5
2490 GR:GOTO-9,6;FILL4
2500 GR:GOTO-3,8;FILL2
2510 GR:GOTO4,6;FILL2;GOTO6,8;FILL2
2520 GR:GOTO4,10;FILL2
2530 GR:GOTO6,12;FILL2
2540 GR:GOTO8,14;FILL2
2550 GR:GOTO10,16;FILL2
2560 GR:GOTO-15,7;FILL2
2570 GR:PEN RED;GOTO-17,-2;4(TURN90:DRAW
2);TURN180:FILL2
2580 GR:GOTO-10,-4;4(TURN90;DRAW2);TURN9
0;FILL2
2590 GR:GOTO-15,-6;4(TURN90;DRAW2);TURN9
0;FILL2
2600 GR:GOTO-4,-2:4(TURN90:DRAW2);TURN18
0;FILL2
2610 GR:GOTO1,-6;4(TURN90:DRAW2);TURN90;
FILL2
2620 GR:GOTO2,-4;4(TURN90:DRAW2);TURN90;
FILL2
2630 GR:GOTO12,12;PEN YELLOW;DRAW6
2640 GR:TURN-90;DRAW4;TURN-45;DRAW4
2650 GR:TURN-115;DRAW6
2660 GR:GOTO0,-8;FILL4
2670 T:   Jemima Puddle-Duck knows you h
ave been working hard, $NAME     \
2680 PA:400
2690 GR:QUIT
2700 E:
2710 *MUSIC
2730 C:#M=#M+1
2740 SO:20
2760 PA:60
2770 PA:60
2780 SO:13
2790 PA:60
2800 SO:17
2810 PA:60
2820 SO:6
2830 PA:60
2840 SO:15 5
2850 PA:60
2860 SO:13 3
2870 PA:60
2880 SO:17 1
2890 PA:120
2900 JK(#M<2):*MUSIC
2910 SO:
2920 E:
```

©

Moire Magic

Leo Cerruti
Syosset, NY

This program produces such beautiful three-dimensional effects in any of four graphics modes that you will, in all likelihood, find yourself staring at the screen and shouting "Hey look at that one..." for hours on end.

This program is user controlled – you can request the density and type of patterns you like. Making use of the "attract mode" produces constant variations in color and intensity and you will never see the same pattern twice. There are two pattern options to choose from: center oriented and corner oriented. That is, lines will emit from the center of the screen or from the corners, as you wish.

Useful Subroutines

This program makes use of several subroutines which should prove useful in many other types of programs, both graphic and nongraphic. For instance; the subroutine from lines 1060 to 1075 causes the word START to flash within the sentence. Line 1070 demonstrates use of the START switch, line 1080 sets attract mode.

When the program is run, it will prompt you to press START to begin. After this it will ask you which graphics mode you wish, modes 3,5,7 or 8. Remember that you need at least 16K to operate in GR.8. The next prompt asks how much maximum spacing you want between the lines drawn on the screen. I suggest numbers between 25 and 40 for GR.8 and 15 to 25 for GR.7. The computer will randomly space the lines up to your maximum. It will never give a spacing of two or less, to prevent filling up the screen with lines or blanks.

You are then asked which pattern you want. Center patterns give a cartwheel moire effect while corner patterns give more of a three-dimensional effect similar to depth lines drawn on a flat plane to give the illusion of distance. Either choice has its own special effects which demonstrate the Atari's superior graphics and color capabilities. You are then asked if you want changing colors. If you type "Y," the "attract mode" is set. Then, happy viewing.

By the way, if you select attract mode do not touch any keys because this will set the "attract mode" counter back to default colors. After each complete cycle in both the center and corner patterns the drawing will pause, select a new line spacing, and then continue. The longer you let the drawing continue, the more complex it will become. You can press CTRL 1 to pause the drawing at anytime, and to continue, press CTRL 1 again. Pressing the BREAK key will end the drawing.

Program Modules

LINE #	DESCRIPTION
0-4	Displays title and jumps to line 1060 which instructs user to push START to begin.
6-9	Requests which graphics mode to use and jumps to lines 1000 to 1030 to set screen margins for the appropriate mode.
10	Requests maximum number of spaces to use between each line.
12-14	Requests corner or center effect moire patterns. Program jumps to line 16 for center pattern or line 600 for corner patterns.
1050	Will set attract mode if you wish.
1060-1090	Will display instruction to push START button and wait for you to do so. Type the word START and a space before and after an inverse video in line 1065. Use normal video for the word START and the spaces before and after in line 1075. This will flash the word START within the sentence.

```
0 ? "}":POSITION 9,8:? "*** MOIRE PATTE
RNS ***"
1 POSITION 9,10:? "*** BY LEO CERRUTI
***"
2 ? :?
3 DIM C$(1)
4 GOSUB 1060
6 POKE 752,0:? "}":? "WHICH GRAPHICS MOD
```

```
E DO YOU WISH:"
7 ? ,"(1) MODE 3":? ,"(2) MODE 5":? ,"(3
) MODE 7":? ,"(4) MODE 8"
8 TRAP 8:INPUT GM:IF GM<1 OR GM>4 THEN 6
9 ON GM GOSUB 1000,1010,1020,1030
10 ? :? "HOW MUCH SPACING MAXIMUM ":INPU
T SPACE
12 ? :? "WHICH MOIRE PATTERN:":? ,"(1) F
ROM CENTER":? ,"(2) FROM CORNERS":INPUT
P
14 ON P GOTO 16,600
15 REM *** CENTER PATTERN ***
16 GOSUB 1050
20 GRAPHICS GR+16:COLOR 1
25 GOSUB 500
30 FOR A=0 TO Y STEP S
40 GOSUB 900:DRAWTO X,A
50 NEXT A
60 FOR B=X TO 0 STEP -S
70 GOSUB 900:DRAWTO B,Y
80 NEXT B
90 FOR C=Y TO 0 STEP -S
100 GOSUB 900:DRAWTO 0,C
110 NEXT C
120 FOR D=0 TO X STEP S
130 GOSUB 900:DRAWTO D,0
140 NEXT D
145 GOSUB 500
147 GOSUB 910
150 FOR E=0 TO Y STEP S
160 GOSUB 900:DRAWTO X,E:COLOR 0
170 NEXT E
180 FOR F=X TO 0 STEP -S
190 GOSUB 900:DRAWTO F,Y
200 NEXT F
210 FOR G=Y TO 0 STEP -S
220 GOSUB 900:DRAWTO 0,G
230 NEXT G
240 FOR H=0 TO X STEP S
250 GOSUB 900:DRAWTO H,0
260 NEXT H
265 GOSUB 910
270 COLOR 1:GOTO 25
299 REM
300 REM ***   RANDOM SPACING   ***
310 REM *** MAXIMUM DETERMINED ***
320 REM ***       BY USER      ***
321 REM
500 S=INT(RND(1)*SPACE)+2:RETURN
549 REM
550 REM *** CORNER PATTERNS ***
551 REM
600 GOSUB 1050
620 GRAPHICS GR+16:COLOR 1
625 GOSUB 500
630 FOR A=0 TO Y STEP S
640 PLOT 0,0:DRAWTO X,A
650 NEXT A
660 FOR B=X TO 0 STEP -S
670 PLOT 0,0:DRAWTO B,Y
680 NEXT B
685 GOSUB 500
690 FOR C=0 TO X STEP S
700 PLOT 0,Y:DRAWTO C,0:COLOR 0
710 NEXT C
720 FOR D=0 TO Y STEP S
730 PLOT O,Y:DRAWTO X,D
740 NEXT D
745 GOSUB 500
750 FOR E=Y TO 0 STEP -S
760 PLOT X,Y:DRAWTO 0,E:COLOR 1
770 NEXT E
780 FOR F=0 TO X STEP S
790 PLOT X,Y:DRAWTO F,0
800 NEXT F
805 GOSUB 500
810 FOR G=X TO 0 STEP -S
820 PLOT X,0:DRAWTO G,Y:COLOR 0
830 NEXT G
840 FOR H=Y TO 0 STEP -S
850 PLOT X,0:DRAWTO 0,H
860 NEXT H
870 COLOR 1:GOSUB 910:GOTO 625
900 PLOT C1,C2:RETURN
904 REM
905 REM *** PAUSE BETWEEN CYCLES ***
906 REM
910 FOR TIME=1 TO 400:NEXT TIME:RETURN
949 REM
950 REM *** GRAPHICS LIMITS ***
951 REM
1000 GR=3:X=39:Y=23:C1=19:C2=11:RETURN
1010 GR=5:X=79:Y=47:C1=39:C2=23:RETURN
1020 GR=7:X=159:Y=95:C1=79:C2=47:RETURN
1030 GR=8:X=319:Y=191:C1=159:C2=95:RETUR
N
1040 REM *** SET ATTRACT MODE ***
1041 REM ***    AND START     ***
1042 REM
1050 ? :? "DO YOU WANT CHANGING COLORS (
Y OR N)":INPUT C$
1055 ? :? "HIT [CTRL  1] TO FREEZE OR R
ELEASE  PATTERN  AT ANY TIME"
1060 C=PEEK(84)
1063 POKE 752,1
1065 FOR P=1 TO 50:NEXT P:POKE 84,C:? "
      HIT [START] TO BEGIN"
1070 IF PEEK(53279)=6 THEN 1080
1075 FOR P=1 TO 50:NEXT P:POKE 84,C:? "
      HIT  START  TO BEGIN":GOTO 1065
1080 IF C$="Y" THEN POKE 77,128
1090 RETURN
```

©

Put A Rainbow In Your Atari

Fred and Doug Tedsen
Sonoma, CA

You've probably seen programs that display 128 colors on the Atari. They are usually interesting to look at, but what do you do with them after you have run them two or three times? Well here is a program that displays a moving rainbow of all 128 colors, and the techniques could easily be used for dramatic title screens in your own programs.

The program begins by drawing the word COLOR in large block letters on the GRAPHICS 7 screen. This is performed by the subroutine at lines 1000 to 1200. While the letters are being drawn, the program is doing a graphics fill. The letter outline is drawn with color register 1 (controlled by SETCOLOR 1) and the inside area of the letters is filled with color register 0 (COLOR 1, SETCOLOR 0). Lines 2000 through 2500 contain the data points for drawing the letters.

The program now goes through a color changing sequence (lines 110-190). This section was included to demonstrate how colors can be controlled inside and outside of the fill areas and also to heighten anticipation for the part that follows. First we randomly change the colors inside the letters, leaving the background black. Next the background colors is changed while the letters remain black. And, finally, we change both the letter and background colors independently. Notice that the letter outlines remain white throughout.

Now The Fun Part

Now we come to the fun part. At the beginning of the program, the subroutine at lines 3000 to 3040 was run to load the machine language color rainbow generator into the strings CUP$ and CDOWN$. These are now used to produce the rainbow pattern. The pattern is first set moving up the screen within the letters, with a black background. The pattern is then put on the background with solid color letters. These steps are then repeated with movement down the screen, just to show that we can go both ways.

So how does it work? Briefly, the POTO register is read and the value obtained placed in one of the playfield color registers. Since the pots are continually counting down to zero, this value changes every scan line. A write to WSYNC makes the change occur at the end of a scan line, resulting in solid lines across the screen. Movement is accomplished by adding or subtracting the value of the 1/60th of a second frame counter to the POTO value before writing it to the color register. The write is directly to the registers in CTIA because the OS shadow registers are not copied until vertical blank and therefore would do nothing. The assembler source listing is included for reference. Notice that there is a direct correspondence between the source listing and the BASIC data statements at lines 3100 to 3280.

Modifications

There are several things which you can do with this routine to change the display:

1. Parameter two in the USR statement is the time in seconds that the routine is to be run. Thus X=USR(ADR(CUP$),4) will display the pattern moving up the screen for about four seconds.

2. You can affect any of the five playfield color registers. To do this you can change either the DATA statement at line 3180 or the machine language string. The values to use are 22, 23, 24, 25, and 26 for color registers 0, 1, 2, 3, and the background, respectively. For example, changing line 3180 to DATA 141,26,208 will affect the background. Line 240 demonstrates how the string may be changed to give the same result.

3. You can obtain a stationary rainbow pattern by changing line 3170 to DATA 234,234 (NOP's). Don't try to use CDOWN$ if you do this, however. A better way would be to change elements 12 and 13 of the string.

4. For those of you with Assembler, there is a myraid of patterns which can be generated by using AND and ORA before writing to the color register. You can create patterns with large bars of color, with small bars of different shades of the same color, or with some combinations – a rainbow of pastels.

5. While the machine language routine is running, your program can't do anything else. Though we haven't tried it yet, it should be possible to incorporate the logic in a display list interrupt routine. This would allow you to do things such as having the pattern roll down the screen with a curtain effect.

There are a couple of things to watch out for when running the program. A time value of zero will run the rainbow routine for about four minutes. If you accidently do this, you will have to press SYSTEM RESET to get out. Also, unplug your paddles from game port 1 to get the proper rainbow effect. You might want to plug a paddle in later to see the effect. It's kind of interesting.

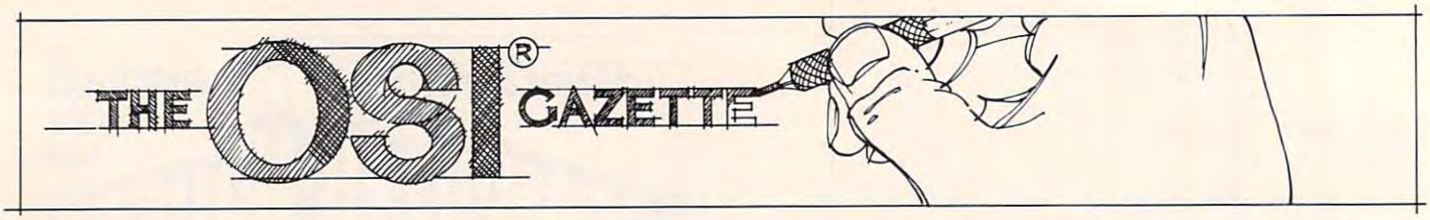
```
1 REM COLOR RAINBOW
2 REM FRED AND DOUG TEDSEN, OCT 1981
10 DIM D$(3),CUP$(32),CDOWN$(32)
20 NS=4:NT=15
30 GOSUB 3000
40 GOSUB 1000
50 FOR I=1 TO 1000:NEXT I
100 N=0
110 SETCOLOR 0,INT(RND(0)*16),2*INT(RND(
0)*8)
120 FOR I=1 TO 150:NEXT I:N=N+1:IF N<NT
THEN GOTO 110
130 SETCOLOR 0,0,0:N=0
140 C=INT(RND(0)*16):I=INT(RND(0)*8)*2:S
ETCOLOR 2,C,I:SETCOLOR 4,C,I
150 FOR I=1 TO 150:NEXT I:N=N+1:IF N<NT
THEN GOTO 140
160 N=0
170 C=INT(RND(0)*16):I=INT(RND(0)*8)*2:S
ETCOLOR 2,C,I:SETCOLOR 4,C,I:SETCOLOR 0,
INT(RND(0)*16),2*INT(RND(0)*8)
180 FOR I=1 TO 120:NEXT I:N=N+1:IF N<NT
THEN GOTO 170
190 SETCOLOR 2,0,0:SETCOLOR 4,0,0
210 SETCOLOR 0,12,6
220 CUP$(15,15)=CHR$(22)
230 X=USR(ADR(CUP$),NS)
240 CUP$(15,15)=CHR$(26)
250 X=USR(ADR(CUP$),NS)
260 SETCOLOR 0,3,4
270 CDOWN$(15,15)=CHR$(22)
280 X=USR(ADR(CDOWN$),NS)
290 CDOWN$(15,15)=CHR$(26)
300 X=USR(ADR(CDOWN$),NS)
310 SETCOLOR 0,7,2
400 FOR I=1 TO 1000:NEXT I:GOTO 40
1000 GRAPHICS 7+16
1010 SETCOLOR 0,0,0:SETCOLOR 1,0,14:SETC
OLOR 2,0,0:SETCOLOR 4,0,0
1020 COLOR 2:FCOLOR=1
1030 RESTORE 2010
1100 READ D$:IF ASC(D$)<64 THEN GOTO 118
0
1110 IF D$="P" THEN READ ROW,COLUMN:GOSU
B 1200:PLOT COLUMN,ROW:GOTO 1100
1120 IF D$="O" THEN READ RORIGIN,CORIGIN
:GOTO 1100
1130 IF D$="END" THEN RETURN
1140 IF D$<>"F" THEN GOTO 1100
1150 READ ROW,COLUMN:GOSUB 1200:POSITION
 COLUMN,ROW:POKE 765,FCOLOR
1160 XIO 18,#6,0,0,"S:":PLOT COLUMN,ROW:
GOTO 1100
1180 ROW=VAL(D$):READ COLUMN:GOSUB 1200:
DRAWTO COLUMN,ROW:GOTO 1100
1200 ROW=ROW+RORIGIN:COLUMN=COLUMN+CORIG
```

```
IN:RETURN
2000 REM "C"
2010 DATA 0,10,2
2020 DATA P,1,9,1,19,3,23,5,25,9,27,15,2
7,15,18,F,12,18,F,10,16,10,12,12,10,29,1
0,31,12
2030 DATA P,26,18,26,27,32,27,36,25,38,2
3,40,19,40,9,F,38,5,F,36,3,F,32,1,F,9,1
2040 DATA F,5,3,F,3,5,F,1,9,P,31,12,F,31
,16,F,29,18,F,26,18
2100 REM "O"
2110 DATA 0,18,32
2120 DATA P,1,9,1,19,3,23,5,25,9,27,32,2
7,36,25,38,23,40,19,40,9
2130 DATA P,10,12,F,10,16,F,12,18,F,29,1
8,F,31,16,31,12,29,10,12,10,10,12
2140 DATA P,40,9,F,38,5,F,36,3,F,32,1,F,
9,1,F,5,3,F,3,5,F,1,9
2200 REM "L"
2210 DATA 0,26,62
2220 DATA P,1,1,1,10,32,10,32,27,40,27,4
0,1,F,1,1
2300 REM "O"
2310 DATA 0,34,92
2320 DATA P,1,9,1,19,3,23,5,25,9,27,32,2
7,36,25,38,23,40,19,40,9
2330 DATA P,10,12,F,10,16,F,12,18,F,29,1
8,F,31,16,31,12,29,10,12,10,10,12
2340 DATA P,40,9,F,38,5,F,36,3,F,32,1,F,
9,1,F,5,3,F,3,5,F,1,9
2400 REM "R"
2410 DATA 0,42,122
2420 DATA P,1,1,1,19,3,23,5,25,9,27,15,2
7,19,25,21,23,22,20,40,27
2430 DATA 40,18,F,21,10,40,10,40,1,P,9,1
0,F,9,16,F,11,18,F,13,18,F,15,16,15,10,9
,10,P,40,1,F,1,1
2500 DATA END
3000 RESTORE 3100
3005 FOR I=1 TO 32
3010 READ C:CUP$(I)=CHR$(C)
3020 NEXT I
3030 CDOWN$=CUP$:CDOWN$(12,12)=CHR$(229)
3040 RETURN
3100 DATA 104
3110 DATA 104
3120 DATA 104
3130 DATA 72
3140 DATA 162,57
3150 DATA 160,0
3160 DATA 173,0,210
3170 DATA 101,20
```

```
3180 DATA 141,22,208
3190 DATA 141,10,212
3200 DATA 136
3210 DATA 208,242
3220 DATA 202
3230 DATA 208,237
3240 DATA 104
3250 DATA 56
3260 DATA 233,1
3270 DATA 208,228
3280 DATA 96
```

```
0100;  ;     RAINBOW COLOR GENERATOR
0110   ;     Fred and Doug Tedsen
0120 ;
0130   RTCLOK3  =     $14
0140   COLPF0   #     $D016
0150   POT0     =     $D200
0160   WSYNC    =     $D40A
0170   ;
0180            PLA           Throw out no. arguments
0190            PLA           Throw out high order byte
0200            PLA           Get no. seconds to run
0210   LOOPA    PHA           Push on stack
0220            LDX   #57     57x256 is about 1 second
0230   LOOPB    LDY   #0
0240   LOOPC    LDA   POT0    Read Pot 0,
0250            ADC   RTCLOK3 add value od 1/60 timer,
0260            STA   COLPF0  and put result in color register 0.
0270            STA   WSYNC   Wait for end of scan line.
0280            DEY
0290            BNE   LOOPC
0300            DEX
0310            BNE   LOOPB
0320            PLA           Get second counter from stack.
0330            SEC           Subtract 1 from counter
0340            SBC   #1      and branch until zero.
0350            BNE   LOOPA
0360            RTS
```

©



Part II:

A Small Operating System: OS65D, The Disk Routines

T. R. Berger
Coon Rapids, MN

Editor's Note: Part I appeared last month. Here, the author presents a map of the disk routines. — RTM

Let's turn to track zero. Exactly one ms. after the index hole a two byte address is recorded on the disk in high byte-low order. This address is read by the ROM on boot. It is the start address for loading track zero into memory. Next comes the number of pages in track zero. Finally, that many pages of data are written on the track. There are no track start or stop markings. After track zero is loaded, the computer always jumps to $2200. Hopefully, track zero has been loaded in that vicinity. It would appear that OSI did not think the track zero format over very carefully.

Subroutine Descriptions

Most of the disk routines are self-explanatory. Because these routines are far more involved than those in the kernel, many more flow charts are needed. Let's run through the memory map in order, commenting on special properties of certain subroutines.

The timing routines at $2678, $267A, and $26A2 are independent of the system clock. The wait time in the routines at $2700, $289F, and $28A4 should be divided by *T* if the system clock is T MHZ.

OS65D does not use binary track numbers, but BASIC does. Thus BASIC uses $26A6, but OS65D enters this routine at $26BC with the BCD track number in the accumulator. With a binary track number in the accumulator, this routine may be entered at $26A9. It will move the disk head over the correct track after some error checking.

The sequence beginning at $2728 may be viewed as the standard startup to read or write a track or sector. It puts the head on the disk, finds the index hole, then initializes the disk data ACIA.

The EXAMINE command uses $2739 to load the entire contents of a track into memory without regard to error checking, track formatting, or sectoring. This type of command is only possible with the asynchronous data format used by OSI. If you crash a track, this command can prove invaluable in retrieving what may remain. I view this routine as a utility. It should reside on the disk and not in memory, unless needed. The initialize routine at $2768 used on a full disk falls in the same category. Such programs as these should be transient, i.e. only called when needed.

The major "Save a Sector" routine begins at $27D7. It uses the data in $265E-$2661. Most of OS65D's disk data is stored in page zero. Because Zpage is swapped out when BASIC comes in, the most important data is repeated in $265C-$2662. BASIC passes its values to these latter locations. LOAD and SAVE routines must then move this data to Zpage. Since OS65D can put information directly into Zpage, it puts the save vector into $FE, $FF directly, entering the Save routine at $27E1. Except when SAVE or CALL are used, all saving is done in Sector one for 12 ($OC) pages on 8" floppies and for eight pages on minifloppies. After a write, the sector is reread and compared with memory. If the comparison fails, the sector is reread again. This may occur up to four times. If comparison still fails, another attempt is made to write the sector. If comparison fails after four rereads again, the operation is aborted with Error #2. To my recollection, I've never seen Error #2 occur. It might happen on an old worn disk, on a midnight special, or with a very dirty head.

The major "Read a Sector" routine is $295D. It uses data in $265-E-$2662. Again OS65D may enter this routine at $2967 if the load vector at $FE, $FF has been set. This program tries to read a sector seven times. The only error check (other than sector seek errors which abort immediately) is a parity check for each byte. If, after seven tries, a read still fails, then the head is moved down then up one track. This whole process may be repeated up to four times before Error #1 is reported. This error also seems to be very rare.

Both read and save routines use the sector seeking routine at $28C4 which, in turn, calls $2998.

OSI TRS-80 COLOR-80 OSI

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

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

THE AARDVARK JOURNAL

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

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

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

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

ADVENTURES!!!

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

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

ESCAPE FROM MARS (by Rodger Olsen)

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

PYRAMID (by Rodger Olsen)

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

TREK ADVENTURE (by Bob Retelle)

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

DEATH SHIP (by Rodger Olsen)

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

VAMPIRE CASTLE (by Mike Bassman)

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

OSI NEW-NEW-NEW OSI
TINY COMPILER

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

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

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

TINY COMPILER — $19.95 on tape or disk OSI

SUPERDISK II

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

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

It also has a full screen editor (in machine code on C2P/C4)) that makes corrections a snap. We'll also toss in renumbering and program search programs — and sell the whole thing for —

SUPERDISK II $29.95 (5¼") OSI

BARE BOARDS FOR OSI C1P

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

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

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

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

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

WORD PROCESSING THE EASY WAY— WITH MAXI-PROS

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

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

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

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

Best of all, it is in BASIC (0S65D 51/4" or 8" disk) so that it can be easily adapted to any printer or printing job and so that it can be sold for a measly price.

MAXI-PROS — $39.95. Specify 5¼ or 8" disk.

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

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

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

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

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

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

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

16K COLOR-80 OR TRS-80 ONLY. $14.95

Please specify system on all orders

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

OSI

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

COLOR-80

Further, they both use a dual purpose routine at $2905. If the accumulator is zero on entry, this routine reads to memory. If it is nonzero, then the routine compares with memory. The actual read and compare loops within this routine are separate. With 8" floppies and a 1 MHZ clock, the 6502 is not fast enough to get from one disk byte to the next if the read and compare loops are combined into one. As it stands, the compare loop just barely returns in time for the next comparison. With a 2 MHZ clock there is plenty of time.

I view the sector directory routines at $29F3 and $2A41 as utilities. They do not need to be resident in memory.

Machine language routines may access the disk directly. For example, to write a sector, locations $265E-$2662 should be assigned correct values. The following segment of code will write a sector to the disk.

```
10 JSR $26A6  ;Move head to track
20 JSR $2754  ;Engage head, find start of track
30 JSR $27D7  ;Write sector
40 JSR $2761  ;Disengage head
50 RTS
```

If the write address is already in $FE, $FF then $27D7 may be entered at $27E1. In this case, lines 20-40 may be replaced by JSR $2CA7, a kernel routine.

To read a sector, again assign correct values to $265E-$2662 then perform the following.

```
10 JSR $26A6  ;Move head to track
20 JSR $2754  ;Engage head, find start of track
30 JSR $295D  ;Read sector
40 JSR $2761  ;Disengage head
50 RTS
```

If the read address is already in $FE, $FF then $295D may be entered at $2967. In this case, lines 20-40 may be replaced by the kernel routine:

```
JSR $2B1A
```

When we discuss the I/O section of OS65D we will see additional ways to read from and write to the disk.

References:

1. Jefferson Harman, "IBM Compatible Disk Drives", Byte October 1979, p. 100

2. Ira Rampil, "A Floppy Disk Tutorial", Byte December 1977, p. 24

3. Les Solomon, "BASICS of Computer Disk Systems", Popular Electronics November 1980, p. 53

MAP – OS65D DISK HANDLER

DISK-MEMORY DATA

265C	DRIVE NUMBER
265D	CURRENT BCD TRACK NUMBER
265E	SECTOR NUMBER
265F	PAGE LENGTH OF SECTOR
2660	LOW BYTE LOAD/SAVE VECTOR
2661	HIGH BYTE LOAD/SAVE VECTOR
2662	BINARY TRACK NUMBER

DISK-Z PAGE

E5	LAST TRACK OF FILE BEING HANDLED
F6	NUMBER OF RETRIES ON WRITE
F7	NUMBER OF HEAD MOVE RETRIES ON READ
F8	NUMBER OR READ RETRIES BEFORE HEAD MOVE
F9	SECTOR COUNT
FA	TARGET TRACK NUMBER ON SEEK
FB	SECTOR NUMBER READ ON DISK
FC	STACK POINTER (IN $29F3)
FD	SECTOR PAGE COUNT (IN $27D7)
FE	SYSTEM POINTER. USED AS
FF	LOAD AND SAVE VECTOR BY DISK

Subroutines – OS65D Disk Handler

2663	Home the Disk. Move the disk head to track 0.
2678	Wait 12 ms.
267A	Wait X ms.
2683	Step up one track toward track 76.
268A	Step down one track toward track 0.
26A2	Wait 8 ms.
26A6	Fetch binary track number from 2662 then:
26A9	Convert track number to BCD then:
26BC	Check for track 0-76 BCD, check for drive ready, move disk head to track, adjust head current, and if an error occurs, abort and send an error message via 2A4B.
2700	Wait 20Y + 7 microseconds (1 MHZ clock).
2708	Adjust head current.
271D	Find trailing edge of index hole.
2728	Engage head then:
272B	Find index hole then:
272E	Initialize disk ACIA.
2739	Engage head, read from index hole full around to index hole, then quit.
2754	Head down.
2761	Head up.
2768	Initialize full disk.
277D	Initialize one track.
27C2	Send a byte to the disk.
27CD	Fetch a byte from the disk.
27D7	Fetch sector save vectors then:
27E1	Save a sector.
289F	Wait 800($FA) microseconds.
28A4	Wait 100Y microseconds.
28B0	Fetch a byte from the disk. Abort with an error message if over the index hole.
28C4	Find the end of the sector preceding the one in 265E.
2905	Read a sector to or compare a sector with memory.
295D	Fetch disk read vector then:
2967	Read and reread a sector to memory, quit if successful or the full number of retries are exhausted.
2998	Find the end of the present sector.
29C6	Select the drive in 265C then:
29DA	Check if the drive is ready.
29EB	8 drive select data bytes.
29F3	Output a sector directory.
2A41	Output subroutine for 29F3.

A Correction For Progressive Computing Chess 1.9

Dave Leskin
Calgary, Canada

Progressive Computing, based in Windsor Ontario, is an excellent source of OSI software with prompt and courteous service; however, there is a major error in their tape version of "Chess 1.9". This error is found in the opening tables. If you try the following sequence of moves you can determine if your copy of "Chess 1.9" has this error too. Note that the last move by the computer is illegal. Microchess notation in brackets.

	Computer (White)	Human (Black)
1	P-K4 (13-33)	P-K4 (63-43)
2	N-KB3 (01-22)	N-QB3 (76-55)
3	B-QN5 (02-46)	N-KB3 (71-52)
4	B-KN5 (05-41) ???	

As you see the "B-KN5" jumps right over the Queen Pawn at 14. To solve this problem I changed the program so that the Queen Knight was moved from 06 to 25 instead. This results in a "Four Knights Game" which is a common opening used by many players. Just follow the steps listed below to effect the change.

The Changes

1. Load "Chess 1.9"

2. Press "D" to enter monitor once the board appears.

3. Press ".0B34"

4. Press "/" to enter the data mode

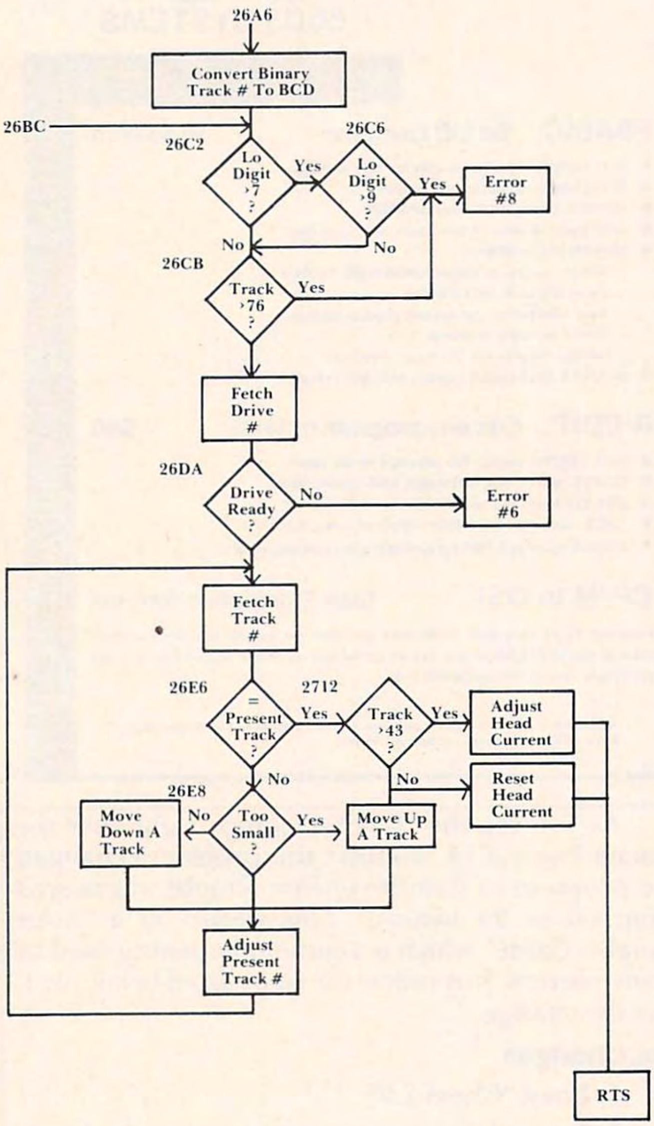
5. Press "25" which refers to the square that the piece will be moved to

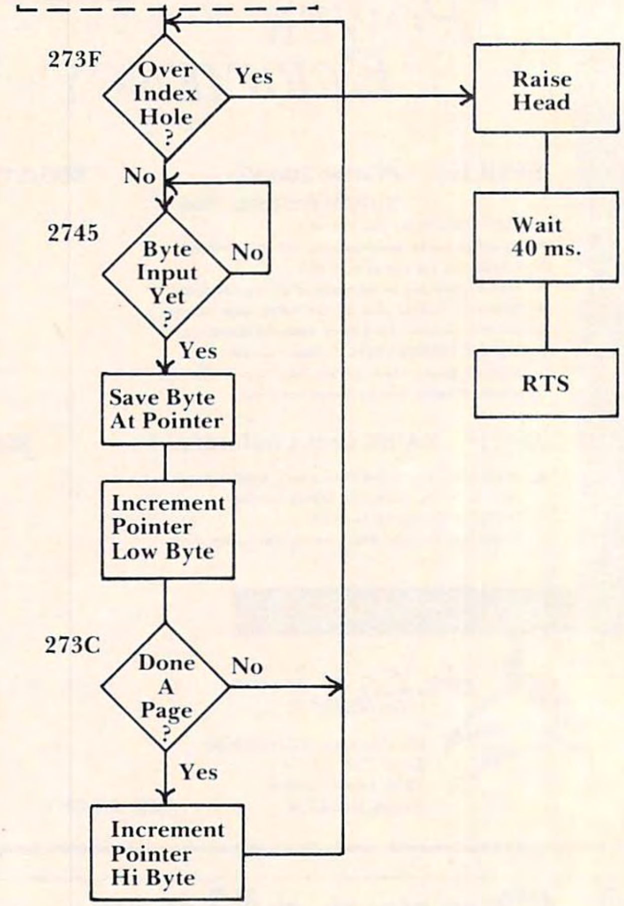
6. Press the "RETURN" key

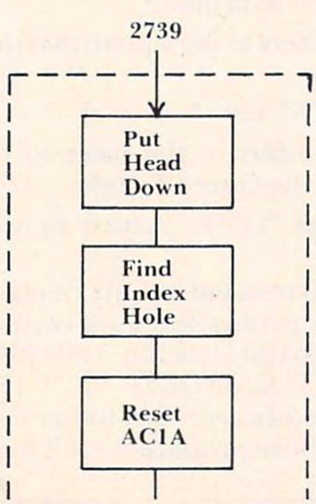
7. Press "07" which refers to the piece to be moved — in this case the Queen Knight

8. Press ".03AC" and "G" to return to the program

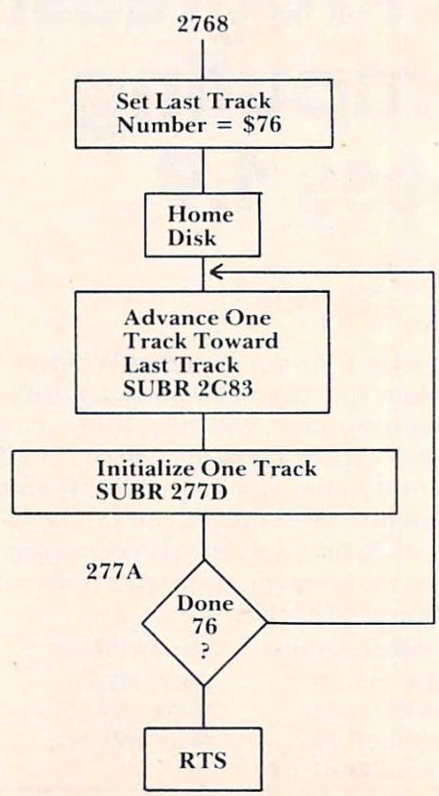
Now the program will respond N-QB3 in place of B-KN5 each time this opening sequence occurs. If you have the capability to store machine code programs (Aardvark's "AUTOLOADER" or "C1E" ROM) then record the modified version (otherwise you'll have to follow the above procedure each time you power up). ©

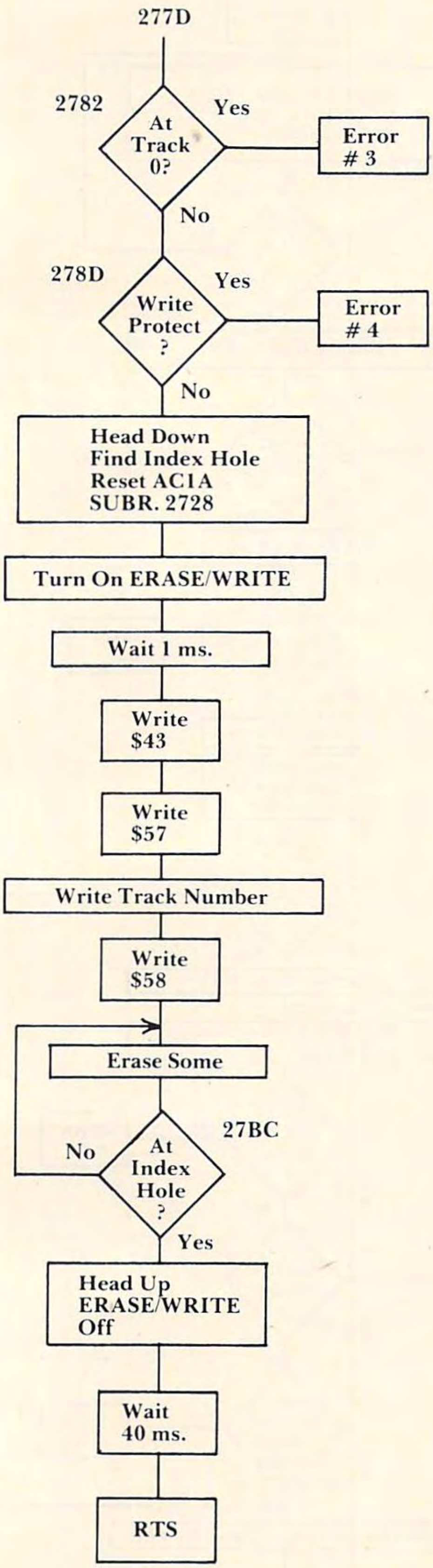

26A6
Convert Binary Track # To BCD
26BC
26C2
Lo Digit ›7 ?
Yes
26C6
Lo Digit ›9 ?
Yes
Error #8
No
No
26CB
Track ›76 ?
Yes
Fetch Drive #
26DA
Drive Ready ?
No
Error #6
Fetch Track #
26E6
= Present Track ?
2712
Yes
Track ›43 ?
Yes
Adjust Head Current
No
Reset Head Current
No
26E8
Too Small ?
No
Move Down A Track
Yes
Move Up A Track
Adjust Present Track #
RTS

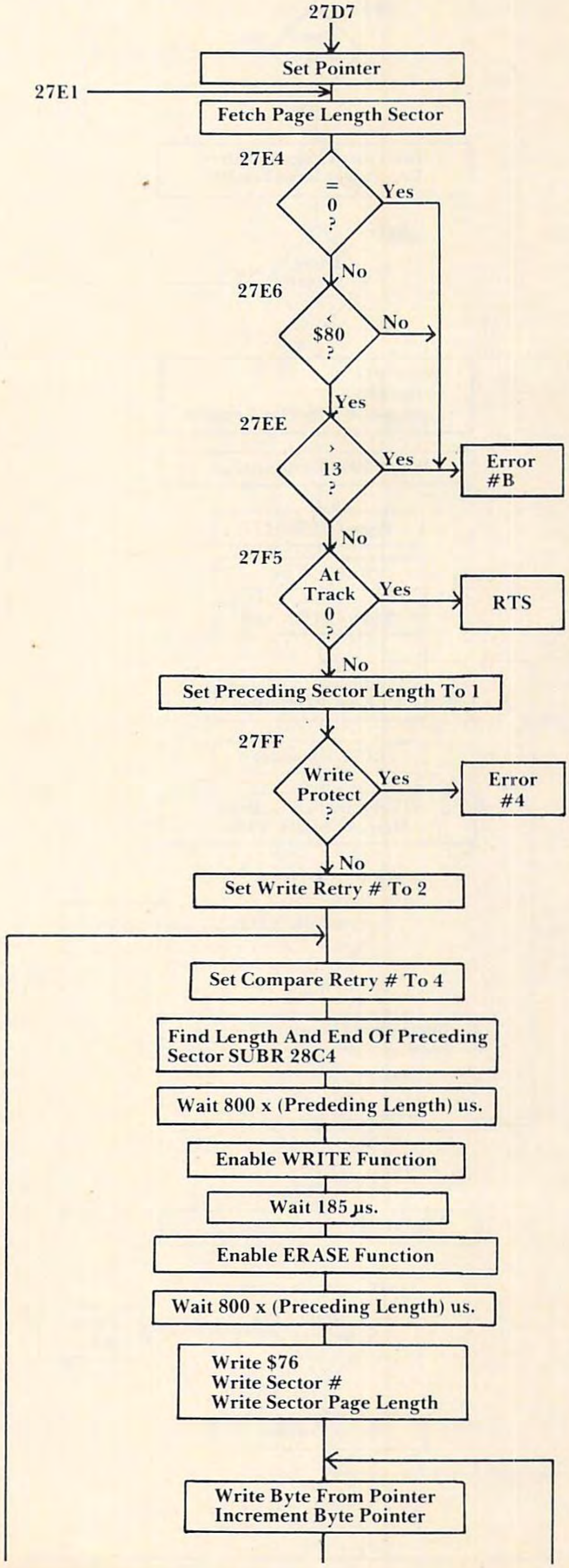

273F
Over Index Hole ?
Yes
Raise Head
Wait 40 ms.
RTS
No
2745
Byte Input Yet ?
No
Yes
Save Byte At Pointer
Increment Pointer Low Byte
273C
Done A Page ?
No
Yes
Increment Pointer Hi Byte

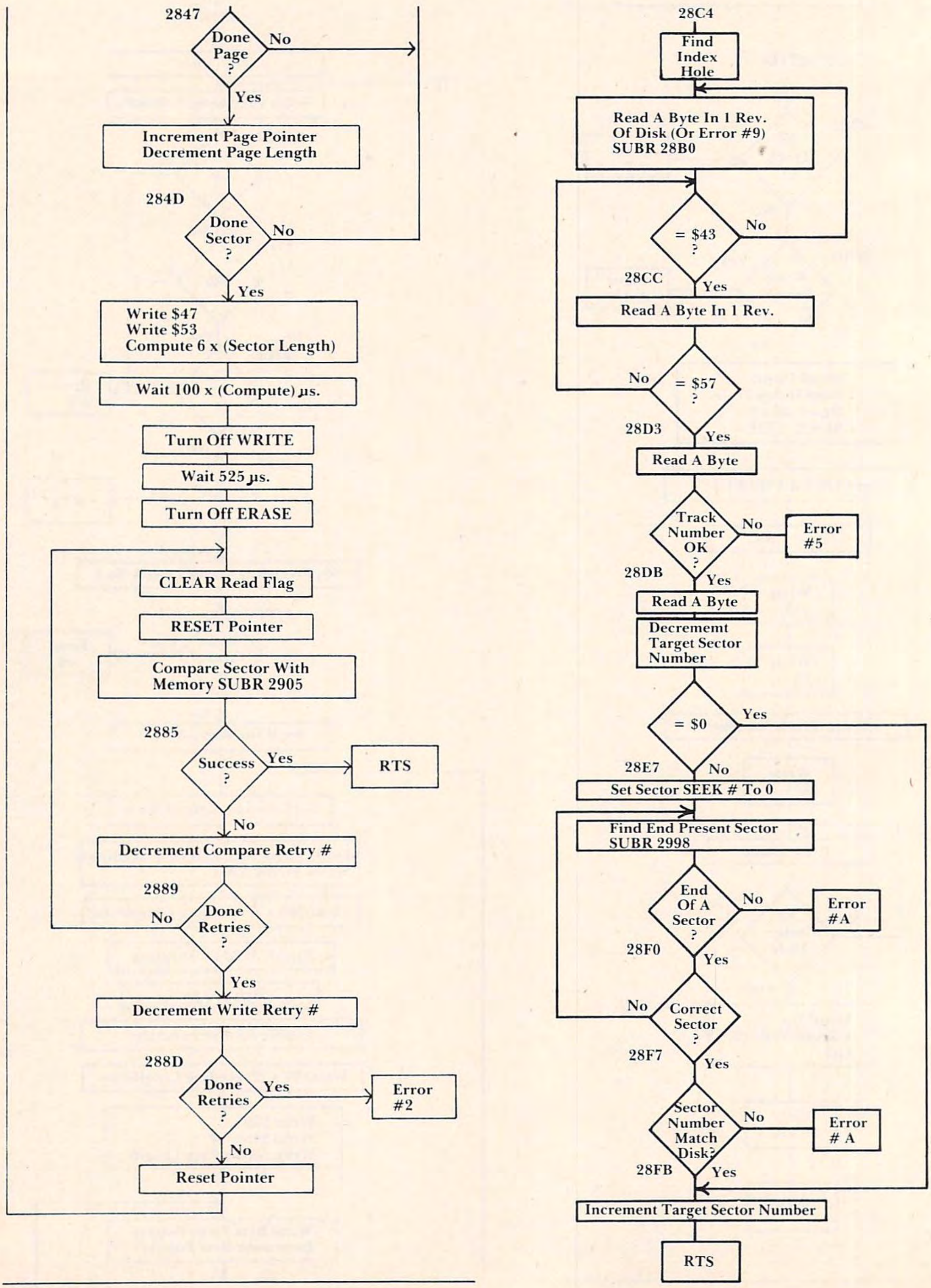

2739
Put Head Down
Find Index Hole
Reset AC1A

SUBR 2728

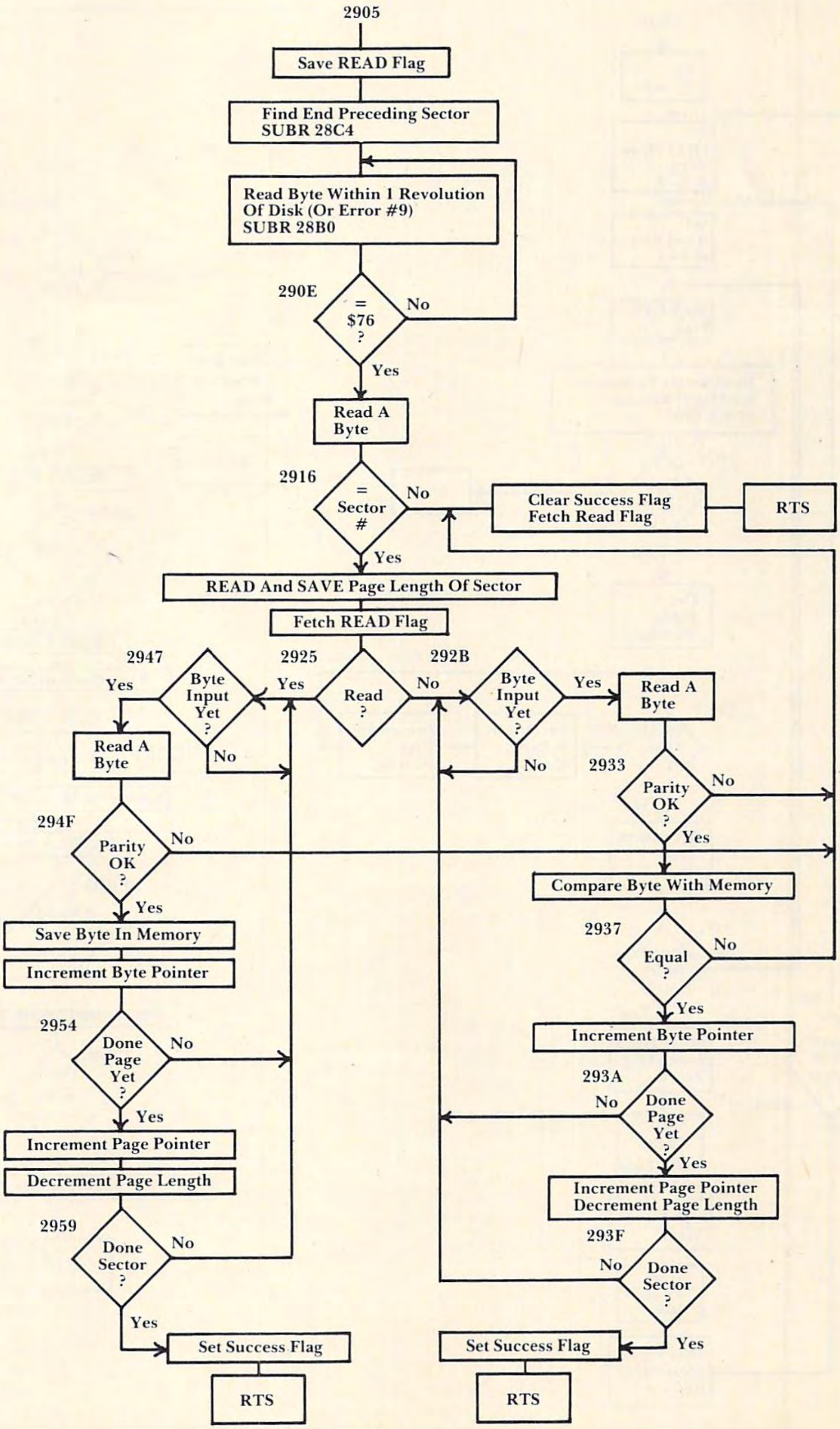

2768
Set Last Track Number = $76
Home Disk
Advance One Track Toward Last Track SUBR 2C83
Initialize One Track SUBR 277D
277A
Done 76 ?
RTS

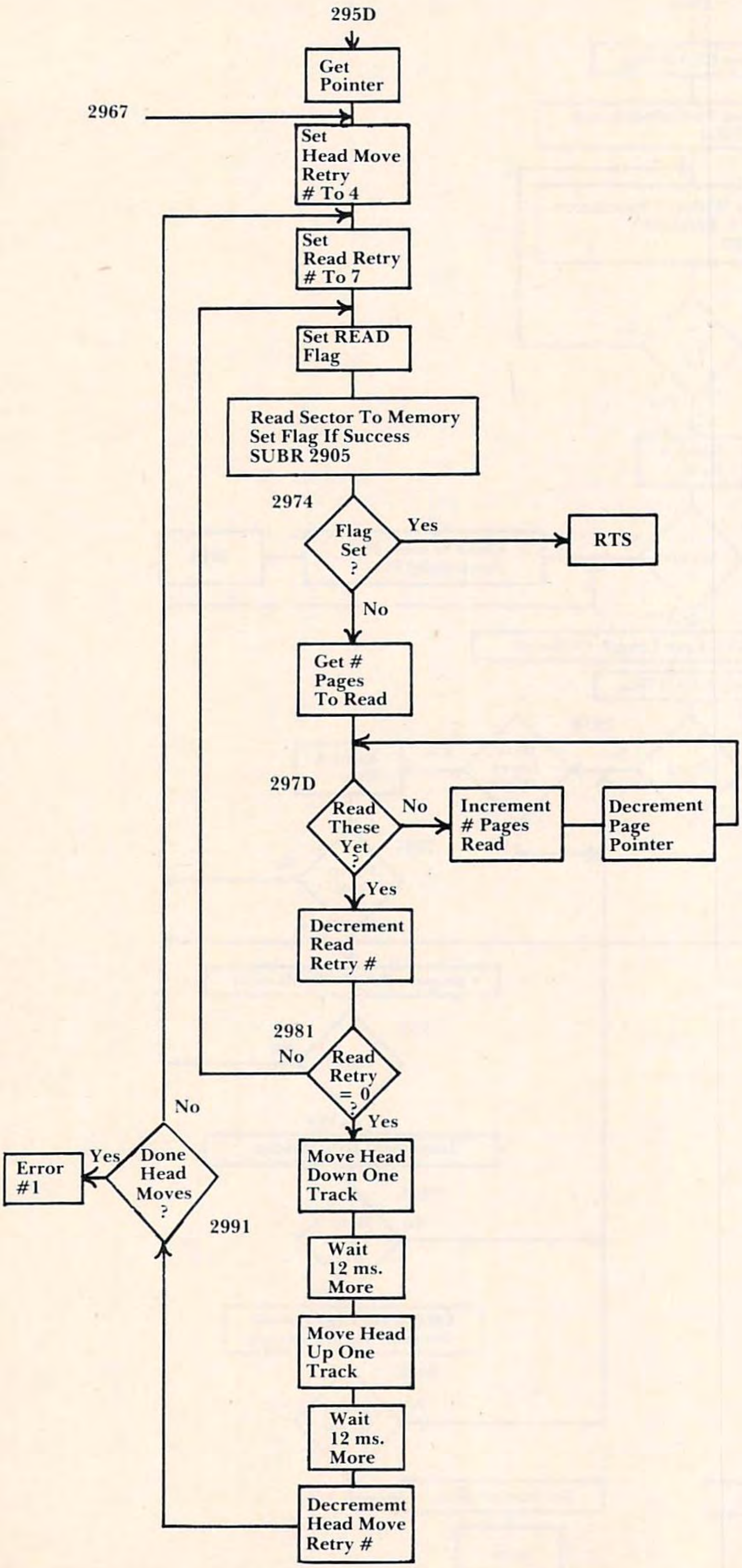

277D
2782
At Track 0?
Yes
Error # 3
No
278D
Write Protect ?
Yes
Error # 4
No
Head Down
Find Index Hole
Reset ACIA
SUBR. 2728
Turn On ERASE/WRITE
Wait 1 ms.
Write $43
Write $57
Write Track Number
Write $58
Erase Some
27BC
No
At Index Hole ?
Yes
Head Up
ERASE/WRITE Off
Wait 40 ms.
RTS

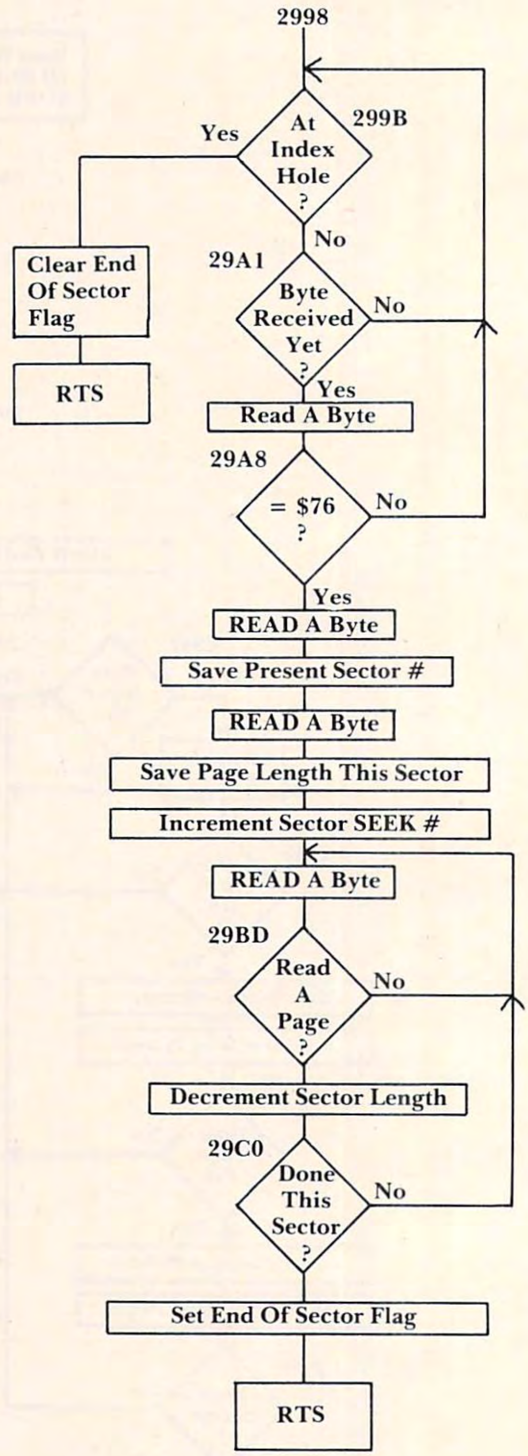

27D7
Set Pointer
27E1
Fetch Page Length Sector
27E4
= 0 ?
Yes
No
27E6
< $80 ?
No
Yes
27EE
> 13 ?
Yes
Error #B
No
27F5
At Track 0 ?
Yes
RTS
No
Set Preceding Sector Length To 1
27FF
Write Protect ?
Yes
Error #4
No
Set Write Retry # To 2
Set Compare Retry # To 4
Find Length And End Of Preceding Sector SUBR 28C4
Wait 800 x (Prededing Length) us.
Enable WRITE Function
Wait 185 µs.
Enable ERASE Function
Wait 800 x (Preceding Length) us.
Write $76
Write Sector #
Write Sector Page Length
Write Byte From Pointer
Increment Byte Pointer

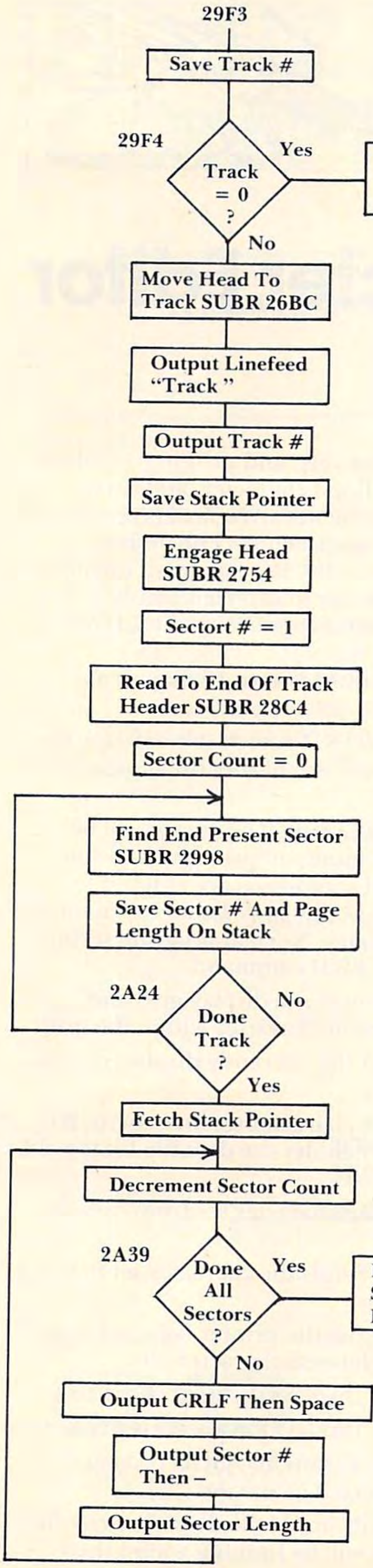


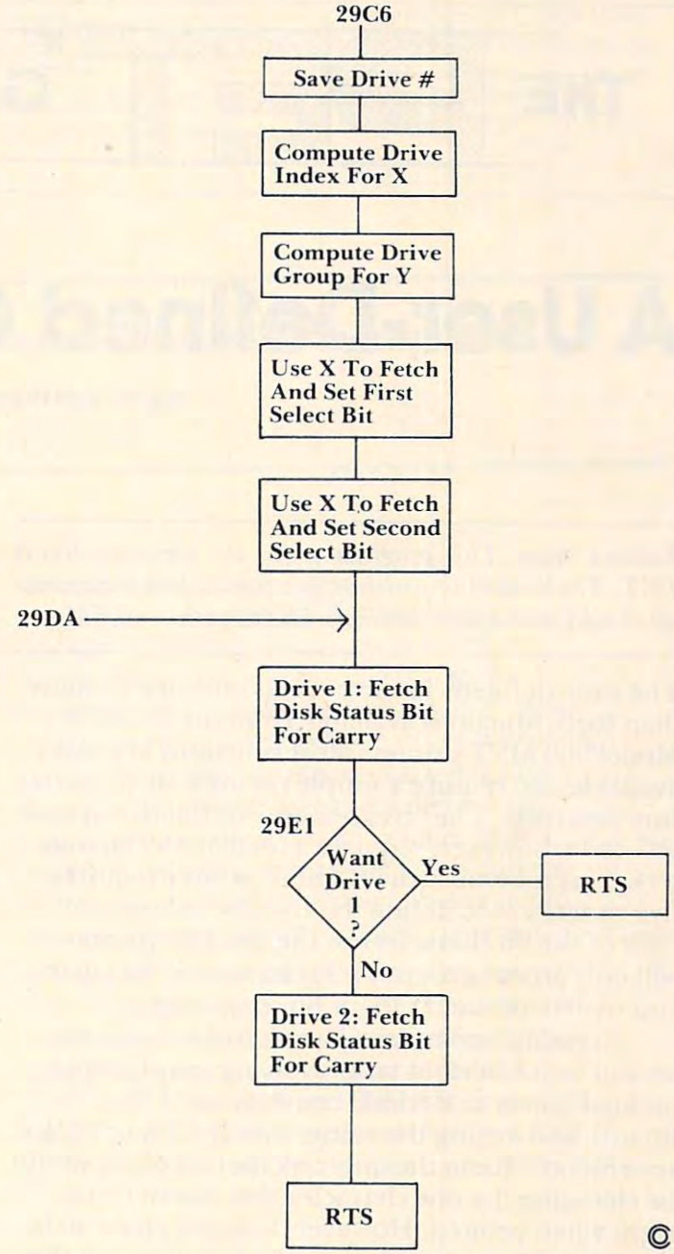
2847
Done Page ?
No
Yes
Increment Page Pointer
Decrement Page Length
284D
Done Sector ?
No
Yes
Write $47
Write $53
Compute 6 x (Sector Length)
Wait 100 x (Compute) µs.
Turn Off WRITE
Wait 525 µs.
Turn Off ERASE
CLEAR Read Flag
RESET Pointer
Compare Sector With Memory SUBR 2905
2885
Success ?
Yes
RTS
No
Decrement Compare Retry #
2889
Done Retries ?
No
Yes
Decrement Write Retry #
288D
Done Retries ?
Yes
Error #2
No
Reset Pointer
28C4
Find Index Hole
Read A Byte In 1 Rev. Of Disk (Or Error #9) SUBR 28B0
= $43 ?
No
28CC
Yes
Read A Byte In 1 Rev.
= $57 ?
No
28D3
Yes
Read A Byte
Track Number OK ?
No
Error #5
28DB
Yes
Read A Byte
Decrememt Target Sector Number
= $0
Yes
28E7
No
Set Sector SEEK # To 0
Find End Present Sector SUBR 2998
End Of A Sector ?
No
Error #A
28F0
Yes
Correct Sector ?
No
28F7
Yes
Sector Number Match Disk?
No
Error # A
28FB
Yes
Increment Target Sector Number
RTS


2905
Save READ Flag
Find End Preceding Sector
SUBR 28C4
Read Byte Within 1 Revolution
Of Disk (Or Error #9)
SUBR 28B0
290E
= $76 ?
No
Yes
Read A Byte
2916
= Sector # ?
No
Clear Success Flag
Fetch Read Flag
RTS
Yes
READ And SAVE Page Length Of Sector
Fetch READ Flag
2947
2925
292B
Byte Input Yet ?
Yes
Read ?
No
Yes
No
Byte Input Yet ?
Yes
Read A Byte
No
No
Read A Byte
2933
Parity OK ?
No
Yes
294F
Parity OK ?
No
Yes
Compare Byte With Memory
Save Byte In Memory
Increment Byte Pointer
2937
Equal ?
No
Yes
Increment Byte Pointer
2954
Done Page Yet ?
No
Yes
293A
No
Done Page Yet ?
Yes
Increment Page Pointer
Decrement Page Length
Increment Page Pointer
Decrement Page Length
2959
Done Sector ?
No
Yes
293F
No
Done Sector ?
Yes
Set Success Flag
Set Success Flag
RTS
RTS


295D
Get Pointer
2967
Set Head Move Retry # To 4
Set Read Retry # To 7
Set READ Flag
Read Sector To Memory Set Flag If Success SUBR 2905
2974
Flag Set ?
Yes
RTS
No
Get # Pages To Read
297D
Read These Yet ?
No
Increment # Pages Read
Decrement Page Pointer
Yes
Decrement Read Retry #
2981
No
Read Retry = 0 ?
Yes
Move Head Down One Track
Wait 12 ms. More
Move Head Up One Track
Wait 12 ms. More
Decrememt Head Move Retry #
2991
Done Head Moves ?
No
Yes
Error #1


2998
299B
At Index Hole ?
Yes
Clear End Of Sector Flag
RTS
No
29A1
Byte Received Yet ?
No
Yes
Read A Byte
29A8
= $76 ?
No
Yes
READ A Byte
Save Present Sector #
READ A Byte
Save Page Length This Sector
Increment Sector SEEK #
READ A Byte
29BD
Read A Page ?
No
Decrement Sector Length
29C0
Done This Sector ?
No
Set End Of Sector Flag
RTS


29F3
Save Track #
29F4
Track = 0 ?
Yes
Clear Carry
RTS
No
Move Head To Track SUBR 26BC
Output Linefeed "Track"
Output Track #
Save Stack Pointer
Engage Head SUBR 2754
Sector # = 1
Read To End Of Track Header SUBR 28C4
Sector Count = 0
Find End Present Sector SUBR 2998
Save Sector # And Page Length On Stack
2A24
Done Track ?
No
Yes
Fetch Stack Pointer
Decrement Sector Count
2A39
Done All Sectors ?
Yes
Reset Stack Pointer
RTS
No
Output CRLF Then Space
Output Sector # Then –
Output Sector Length


29C6
Save Drive #
Compute Drive Index For X
Compute Drive Group For Y
Use X To Fetch And Set First Select Bit
Use X To Fetch And Set Second Select Bit
29DA
Drive 1: Fetch Disk Status Bit For Carry
29E1
Want Drive 1 ?
Yes
RTS
No
Drive 2: Fetch Disk Status Bit For Carry
RTS

©

character grid while in this mode. Table 2 contains the subcommands of the EDIT mode. A raster (*) is the only exit from this mode.

GENERAL INPUT. The CURSOR input routine. It is called by COMMAND DECODE and INPUT CODE.

TEST. Prints the entire character set on the printer.

TABLE. Prints a table in the format of Table 3.

END. Closes all devices and clears the screen. Can be called at any time without disturbing the new character set.

It is possible to create a reasonable facsimile of the PET's shifted keyboard graphics with the 5 by 7 BASE 2 print matrix. (The PET screen matrix is 8 by 8.) DATA statements for this pseudo-PET character set are included in Program 2. Printing all characters displayable on the PET screen would require 3 user defined character sets on the BASE 2. An easier way of implementing "full" pseudo-PET graphics is via the graphics function of the BASE 2.

Standard and user-defined characters can be displayed on the same line, but doing so requires two passes of the print head; i.e., normal print followed by carriage return without linefeed and user-defined print. This is not difficult to arrange during formatted printing from a program. Listing a program is more complicated. One approach is to list to the screen, sort and count characters, and then use one of the many screen printing routines previously published in **COMPUTE!** and other magazines.

While on the subject of listings, it should be mentioned that the normal list sequence of:

OPEN5,4:CMD5:LIST

will not work for long listings, probably due to some bug in the timing when CMD is invoked. The following sequence, using the terminal buffer feature of the BASE 2 (run in either immediate or program mode) will work:

OPEN5,4
PRINT#5,CHR$(27);CHR$(82);CHR$(20);CHR$(80);
PRINT#5,CHR$(27);CHR$(54);
CMD5:LIST (or LIST XXX- in program mode)
PRINT#5,CHR$(27);CHR$(83):CLOSE5

Whether your application is mathematics, foreign languages, APL, or whatever, design your own character set with ease using CHARACTER EDITOR.

Table 1.

COMMANDS (all followed by RETURN)

I(RETURN)##	**Input code and display character**
+	**Display next character**
–	**Display previous character**
E	**Edit. (See Table 2)**
L	**Load character set from tape**
P	**Download Character Set To Printer**
S	**Save Character Set to Tape**
T	**Toggle printer**
TABLE	**Print table in format of Table 3**
TEST	**Test print entire character set**
END	**Terminate program**

Table 2.

EDIT MODE SUBCOMMANDS

>	**Move cursor to next grid point**
<	**Move cursor to previous grid point**
↑	**Move cursor up one row**
←	**Move cursor down one row**
SPACE key	**Erase matrix point and move to next grid point**
shifted &	**Insert matrix point at grid point**
C	**Clear character grid**
*****	**Store displayed character in memory and return to COMMAND DECODE mode**

Table 3.

```
CHR$( 32)  " "   0    0    0    0    0
CHR$( 33)  "▌"   127  127  127  0    0
CHR$( 34)  "▄"   120  120  120  120  120
CHR$( 35)  "‾"   1    1    1    1    1
CHR$( 36)  "_"   64   64   64   64   64
CHR$( 37)  "| "  127  0    0    0    0
CHR$( 38)  "▒"   85   42   85   42   85
CHR$( 39)  " |"  0    0    0    0    127
CHR$( 40)  "▒"   80   40   80   40   80
CHR$( 41)  "◤"   127  63   15   3    1
CHR$( 42)  "▐"   0    0    0    127  127
```

Program 1.

```
100 REM PET/BASE TWO PROGRAMMABLE ¬
    ¬CHARACTER EDITOR
105 REM
110 REM BY P. J. ROVERO
115 REM    NOCC COMNAVMAR BOX 2
120 REM    FPO S. F.       96630
125 REM
130 REM THIS PROGRAM ENABLES THE USER ¬
    ¬TO EASILY BUILD, STORE, AND
135 REM EDIT CHARACTERS IN THE FORMAT ¬
    ¬USED BY THE BASE TWO MODEL
140 REM 800 MST PRINTER.
145 REM
150 REM THE VARIABLE BA SHOULD BE ¬
    ¬CHANGED TO SUIT THE SYSTEM.
155 REM BA= START ADDRESS OF 482 BYTES ¬
    ¬OF BASIC-PROTECTED
160 REM MEMORY REQUIRED FOR USER ¬
    ¬CHARACTER TABLE.
165 REM
170 REM COMMAND SUMMARY:
175 REM
180 REM S  SAVE CHARACTER SET TO TAPE#1
185 REM L  LOAD CHARACTER SET FM TAPE#1
190 REM P  DOWNLOAD CHARACTER SET TO ¬
    ¬PRINTER MEMORY
195 REM T  TOGGLE PRINTER BETWEEN ¬
```

```
     ¬CHARACTER SETS
200 REM TEST  PRINT USER DEFINED ¬
     ¬CHARACTER SET
205 REM TABLE  PRINT A TABLE IN FORMAT ¬
     ¬CHR$(X)  "CHARACTER"  DATA STREAM
210 REM +  DISPLAY NEXT CHARACTER
215 REM -  DISPLAY PREVIOUS CHARACTER
220 REM I  INPUT CHARACTER CODE AND ¬
     ¬DISPLAY CHARACTER
225 REM E  EDIT CHARACTER DISPLAYED ON ¬
     ¬SCREEN.  SUBCOMMANDS IN THIS
230 REM    MODE ARE >,<,^,^ FOR CURSOR  ¬
     ¬CONTROL AND * TO ENTER
235 REM    DISPLAYED CHARACTER INTO ¬
     ¬MEMORY.
240 REM END  CLOSE ALL DEVICES AND END
245 REM******
250 REM       INITIALIZATION ROUTINE
255 REM******
260 POKE134,00:POKE135,60:REM PROTECT ¬
     ¬TOP 1K OF MEMORY
265 PRINT"ĥINITIALIZING VARIABLES"
270 TG=77:CU=32849:BA=15361
275 DIM SL%(7),SH%(7),CC%(5),LC%(7)
280 CR$=CHR$(13)
285 FORI=0TO4:CC%(I)=1+I*3:NEXT
290 FORI=0TO6:LC%(I)=1+I*3:NEXT
295 FORI=0TO6:READ SL%(I):NEXT
300 DATA80,200,64,184,48,168,032
305 FORI=0TO6:READ SH%(I):NEXT
310 DATA128,128,129,129,130,130,131
315 CO%=32:POKEBA,27:POKE(BA+1),75
320 REM******
325 REM       COMMAND DECODE ROUTINE
330 REM******
335 PRINT"ĥ"
340 GOSUB415:REM INITIAL DISPLAY
345 GOSUB1165:REM  INPUT
350 IFIN$="S"THENGOTO650
355 IFIN$="L"THENGOTO765
360 IFIN$="P"THENGOSUB835
365 IFIN$="T"THENGOSUB890
370 IFIN$="TEST"THENGOSUB1260
375 IFIN$="+"THENGOSUB930
380 IFIN$="-"THENGOSUB965
385 IFIN$="I"THENGOSUB1000
390 IFIN$="E"THENGOSUB1060
395 IFIN$="TABLE"THENGOSUB1315
400 IFIN$="END"THENGOTO1305
405 GOTO345
410 REM******
415 REM       INITIAL DISPLAY
420 REM******
425 A$="O##":B$="O#P":C$="L$$":D$="L$:"
430 E$="%  ":F$="%  '"
435 A1$=A$+A$+A$+A$+B$
440 A2$=E$+E$+E$+E$+F$
445 A3$=C$+C$+C$+C$+D$
450 PRINT"ȟ"
455 FORI=0TO6
460 PRINTA1$:PRINTA2$:PRINTA2$
465 NEXT
470 PRINTA3$
475 GOTO345
480 REM******
485 REM       DISPLAY SR
490 REM******
495 BT=BA+2+((CO%-32)*5)
500 FORM=BTTO(BT+4)
505 C2=CU+((M-BT)*3)
510 CH%=PEEK(M)
515 FORN=0TO6
520 C3=C2+(N*120)
525 POKEC3,42
530 IFCH%AND(2^N)THENPOKEC3,102:GOTO540
535 POKEC3,32
540 NEXTN
545 NEXTM
550 POKE224,184:POKE225,129:POKE226,19:
     ¬POKE245,11:POKE5,19
555 PRINT"          <<<<<<<<<<";"CHR$(";
     ¬CO%;")";
560 RETURN
565 REM******
570 REM       RASTER SR
575 REM******
580 FORJ=0TO4
585 CH%=0:SPOT=CU+(J*3)
590 FORK=0TO6
595 IFPEEK(SPOT)=102ORPEEK(SPOT)=230THEN
     ¬CH%=CH%+(2^K)
600 SPOT=SPOT+120
605 NEXTK
610 BT=BA+2+((CO%-32)*5)+J
615 POKEBT,CH%
620 NEXTJ
625 GOSUB485
630 RETURN
635 REM******
640 REM       SAVE SR
645 REM******
650 PRINT"ĥ"
655 INPUT"CHARACTER FILENAME";A$
660 POKE243,122:POKE244,2
665 OPEN1,1,2,A$
670 M=0
675 FORN=(BA)TO(BA+481)
680 CI%=PEEK(N)
685 PRINT#1,CI%;
690 PRINTCI%;
695 M=M+1:IFM=191THENGOSUB725
700 PRINT#1,CHR$(13);
705 M=M+1:IFM=191THENGOSUB725
710 NEXT
715 PRINT"ĥ":CLOSE1:GOTO415
720 REM******
725 REM       FORCE INTER-RECORD GAP
730 REM******
735 POKE59411,53
740 T1=TI
745 IF(TI-T1)<20GOTO745
750 POKE59411,61:M=0
755 RETURN
760 REM******
765 REM       LOAD SR
770 REM******
775 PRINT"ĥ"
780 INPUT"CHARACTER FILENAME";A$
785 OPEN1,1,0,A$
790 FORN=0TO481
795 INPUT#1,CI%
```

SUPERGRAPHICS

128,000 ACCESSIBLE POINTS

Features

* 128,000 accessible points arranged in 200 x 640 grid
* 16K static ram supplies memory mapping of pixels on separate board
* supported through extended Basic commands supplied in ROM
* uses no memory from the existing system
* mix standard CBM text and low-res graphics with hi-res displays
* easy to design and display user defined shapes
* normal or inverted display mode for partial or total screen
* switch graphic display on or off - continue output in passive mode
* save graphic images on a disk file
* recall images from disk with option to mix with existing display
* supports hardcopy to Epson MX-82 or Centronix 739 printers
* easy to follow installation instruction included

CMD

CANADIAN MICRO DISTRIBUTORS LTD.

365 Main St., Milton, Ontario L9T 1P7
416 878-7277

DISTRIBUTORS OF:
MUPET • DOUBLE-MUPET • SPOOLER
THE MANAGER • I/O PRODUCTS

```
800 IFST=-128GOTO820
805 POKE(BA+N),CI%
810 PRINTCI%;
815 NEXT
820 CLOSE1
825 PRINT"ĥ":GOTO415
830 REM******
835 REM      DOWNLOAD CHARSET
840 REM******
845 OPEN5,4
850 FORM=BATO(BA+481)
855 CH%=PEEK(M)
860 PRINT#5,CHR$(CH%);
865 NEXTM
870 PRINT#5:PRINT#5,CHR$(27);CHR$(76)
875 CLOSE5
880 RETURN
885 REM******
890 REM      TOGGLE SR
895 REM******
900 IFTG=77THENTG=76:GOTO910
905 IFTG=76THENTG=77
910 OPEN5,4:PRINT#5,CHR$(27);CHR$(TG)
915 CLOSE5
920 RETURN
925 REM******
930 REM      INCREMENT SR
935 REM******
940 CO%=CO%+1
945 IFCO%>127THENCO%=32
950 GOSUB485
955 RETURN
960 REM******
965 REM      DECREMENT SR
970 REM******
975 CO%=CO%-1
980 IFCO%<32THENCO%=127
985 GOSUB485
990 RETURN
995 REM******
1000 REM      INPUT CO% SR
1005 REM******
1010 GOSUB1165
1015 IN=VAL(IN$)
1020 IFIN<32THENIN=32
1025 IFIN>127THENIN=127
1030 CO%=INT(IN)
1035 GOSUB485
1040 RETURN
1045 REM******
1050 REM      EDIT SR
1055 REM******
1060 POKE224,080:POKE225,128:POKE226,1:
     ¬POKE245,1
1065 N=0:A$=""
1070 POKE548,0
1075 GETA$:IFA$=""GOTO1075
1080 IFA$=">"THENN=N+1:GOTO1120
1085 IFA$="<"THENN=N-1:GOTO1120
1090 IFA$="^"THENN=N-5:GOTO1120
1095 IFA$="^"THENN=N+5:GOTO1120
1100 IFA$="&"THENPRINT"&<";:N=N+1:
     ¬GOTO1120
1105 IFA$=" "THENPRINT" <";:N=N+1:
     ¬GOTO1120
1110 IFA$="*"THENPOKE548,1:GOSUB570:
     ¬GOTO1155
1115 IFA$="C"THENPOKE548,1:GOSUB1205:
     ¬GOSUB485:GOTO1060
1120 POKE514,0:WAIT514,6
1125 IFN<0ORN>34THENN=0
1130 NC=N:NL=INT((N/5))
1135 IFNC>4THENNC=NC-5:GOTO1135
1140 POKE224,(SL%(NL)):POKE225,(SH%(NL))
     ¬:POKE226,(CC%(NC)):POKE245,
     ¬(LC%(NL))
1145 SPOT=256*SH%(NL)+SL%(NL)+CC%(NC)
1150 GOTO1070
1155 RETURN
1160 REM******
1165 REM      GENERAL INPUT SR
1170 REM******
1175 POKE224,24:POKE225,129:POKE226,19:
     ¬POKE245,7:POKE005,19
1180 PRINT"          <<<<<<<<<<<CMD?";
1185 IN$=" ":ZT=TI:ZC=2:ZD$=CHR$(20)
1190 GETZ$:IFZ$<>""THENGOTO1220
1195 IFZT<TITHENPRINTMID$(" &",ZC,
     ¬1);"<";:ZC=3-ZC:ZT=TI+10
1200 GOTO1190
1205 BT=BA+2+(5*(CO%-32))
1210 FORI=BTTO(BT+4):POKEI,0:NEXTI
1215 RETURN
1220 Z=ASC(Z$):ZL=LEN(IN$):IF(ZAND127)<3
     ¬2THENPRINT"<";:GOTO1235
1225 IFZL>254THENGOTO1190
1230 IN$=IN$+Z$:PRINTZ$;ZD$;Z$;
1235 IFZ=13THENIN$=MID$(IN$,2):PRINTCR$;
     ¬:RETURN
1240 IFZ=20ANDZL>1THENIN$=LEFT$(IN$,
     ¬ZL-1):PRINT"<";:GOTO1190
1245 IFZ=141THENZ$=CHR$(-20*(ZL-1)):
     ¬FORZ=2TOZL:PRINTZ$;:NEXTZ:GOTO1165
1250 GOTO1190
1255 REM******
1260 REM      TEST SR
1265 REM******
1270 OPEN5,4:PRINT#5,CHR$(27);CHR$(50)
1275 FORN=32TO127:PRINT#5,CHR$(N);:NEXT
1280 PRINT#5:CLOSE5
1285 RETURN
1290 REM******
1295 REM      END
1300 REM******
1305 PRINT"ĥ":CLOSE1:CLOSE5:END
1310 REM******
1315 REM      TABLE SR
1320 REM******
1325 OPEN5,4
1327 PRINT#5,CHR$(27);CHR$(106);
1330 FORI=0TO95
1335 PRINT#5,CHR$(27);CHR$(77);
1340 PRINT#5,"CHR$(";I+32;")    ";CHR$(34
     ¬);CHR$(13);
1345 PRINT#5,CHR$(27);CHR$(76);"        ¬
     ¬       ";CHR$(I+32);CHR$(13);
1350 PRINT#5,CHR$(27);CHR$(77);"        ¬
     ¬       ";CHR$(34);"    ";
1355 FORK=0TO4
1360 J=BA+2+(I*5)+K:PRINT#5,PEEK(J);"  ¬
     ¬";
1365 NEXTK
```

```
1370 PRINT#5
1375 NEXTI
1380 CLOSE5:RETURN
READY.
```

Program 2. PSEUDO-PET CHARACTER SET

```
115 DATA27,75,0,0,0,0,0,127,127,127,0,0,120,120,120,120,120,1,1,1,1,1
120 DATA64,64,64,64,64,127,0,0,0,0,85,42,85,42,85,0,0,0,0,127,80,40,80,40,80
125 DATA127,63,15,3,1,0,0,0,127,127,0,0,127,8,8,0,0,120,120,120,0,0,15,8,8
130 DATA8,8,120,0,0,96,96,96,96,96,0,0,120,8,8,8,8,15,8,8,8,8,120,8,8
135 DATA8,8,127,0,0,127,127,0,0,0,127,127,127,0,0,0,0,127,127,127
140 DATA3,3,3,3,3,7,7,7,7,7,112,112,112,112,112,64,64,64,64,127
145 DATA120,120,120,0,0,0,0,15,15,15,8,8,15,0,0,15,15,15,0,0,15,15,127,120,120
150 DATA8,8,8,8,8,28,14,127,14,28,0,127,127,0,0,24,24,24,24,24,12,12,12,12,12
155 DATA2,2,2,2,2,48,48,48,48,48,0,127,127,0,0,0,127,127,0,8,8,112,0,0,0,0,7
160 DATA8,8,8,8,7,0,0,127,64,64,64,64,3,4,8,16,96,96,16,8,4,3,127,1,1,1,1
165 DATA1,1,1,1,127,62,127,127,127,62,32,32,32,32,32,12,30,60,30,12
170 DATA0,127,0,0,0,0,0,112,8,8,99,20,8,20,99,62,65,65,65,62,28,10,127,10,28
175 DATA0,0,0,127,0,12,30,63,30,12,8,8,127,8,8,85,42,85,0,0,0,0,127,127,0
180 DATA4,126,2,126,3,1,7,15,63,127
185 DATA0,0,0,0,0,127,127,127,0,0,120,120,120,120,120,1,1,1,1,1
190 DATA64,64,64,64,64,127,0,0,0,0,85,42,85,42,85,0,0,0,0,127,80,40,80,40,80
195 DATA127,63,15,3,1,0,0,0,127,127,0,0,127,8,8,0,0,120,120,120,0,0,15,8,8
200 DATA8,8,120,0,0,96,96,96,96,96,0,0,120,8,8,8,8,15,8,8,8,8,120,8,8
205 DATA8,8,127,0,0,127,127,0,0,0,127,127,127,0,0,0,0,127,127,127
210 DATA3,3,3,3,3,7,7,7,7,7,112,112,112,112,112,64,64,64,64,127
215 DATA120,120,120,0,0,0,0,15,15,15,8,8,15,0,0,15,15,15,0,0,15,15,127,120,120
```

©

Marquee

Mark Bernstein
Department of Chemistry
Harvard University

Editor's Note: Although Mr. Bernstein's annotated source code starts at address $7000 (28672 decimal), we have included a BASIC loader (Program 1) which places the routine at $0360 (864 decimal) for those who have less RAM memory or prefer the convenience of storing machine language routines in the second cassette buffer. The screen size is set for 40 and the speed is 5. For an 80 column screen, POKE 864,80 and to change the speed, POKE 866,X. To test the routine, at this location, you would type SYS 1008. — RTM

The video display is a programmer's canvas. In this small space the programmer must communicate, inform, and perhaps entertain and enthrall. But, like all artists, programmers must work within the confines of their frame and the limits of their medium; all too often, the TV screen seems cramped and small.

All programmers must adjust to and accomodate the limitations of their computer's display. When using machine language, though, programmers must often work with awkward and clumsy tools. BASIC, PASCAL, FORTH and the like provide simple amenities like carriage returns, automatic spacing and tabs, while machine language leaves programmers to do all the work themselves. High level languages let programmers think in terms of character strings and display lines; assembly language programmers must think of individual symbols and screen locations.

In simulation and game programming, screen design can become a contest between graphics and text. An abundance of information, some vital, some merely interesting, competes for space within the screen's limited frame. Intricate graphics and display modes can compress lots of information into a small space — a picture is worth a thousand words — but usually demand intricate and time-consuming programming. Often the special programming is simply not worth the effort, and so the display has to be pruned. Information that won't fit on the screen remains forever hidden inside the computer.

Scrolling Text

The programmer's art ought not to be limited by the confines of the machine, only by skill and imagination. One useful solution to this conflict between the information and display space is the *marquee*, a small area of the screen across which text scrolls from right to left. The whole message doesn't have to be displayed at one time, so less space needs to be reserved for text and more area can be used for graphics. Long and short messages can be displayed with equal ease. And users, trained by long years of watching scoreboards, advertising displays and theatre marquees, find scrolling displays easy to understand and to use.

Using Interrupts

The computer takes only a few milliseconds to write a conventional message on the screen. Normally, writing occupies the computer's complete attention, and everything else must wait until the whole message has been displayed. But, since computers can write very quickly and people read comparatively slowly, most of the computer's time remains free for data processing.

Marquee displays, on the other hand, are intimately tied to human reading speed. The computer needs very little time to update the marquee, and could add a new letter a thousand times a second. If the computer wrote at full speed, the message would whiz across the screen, an illegible blur. To be useful, the marquee must move slowly.

Long marquee displays require many seconds, even minutes. This delay would be unacceptable if the computer were continuously occupied while displaying the message. The computer should not have to wait for the slow human reader. Instead, useful work can be accomplished in the long intervals between marquee updates.

We use a programmable timer to *interrupt* the computer periodically. A few times each second this interrupt instructs the computer to advance the marquee one step. The computer spends the rest of the time running its program normally and returns, after each marquee update, to the task that was interrupted.

An important benefit of this *interrupt-driven* strategy is *transparency*. Conventional, all-at-once output is simple and modular. The user's program calls an output routine, the output routine writes the specified letters on the screen, and then control returns to the user's program. To make marquees easy for the programmer, they should seem (to the programmer) to work just like normal output routines.

Interrupts make marquee displays as easy to use as normal output routines. Conventional routines do the writing immediately; the marquee controller arranges for the periodic interrupts which, without further intervention from the program, will draw the message on the marquee.

The Marquee Generator

Figure 1 shows the structure of a user program which invokes the marquee generator. The user program can activate the marquee by calling STOP. When the marquee is active, periodic interrupts divert the computer's attention from the user's program (left column) to the interrupt service routines (right column).

Figure 1.

The user doesn't have to control the marquee directly. Instead, periodic interrupts invoke IRQSRV, which decides whether the marquee should be updated. Updates are handled by invoking SCROLL.

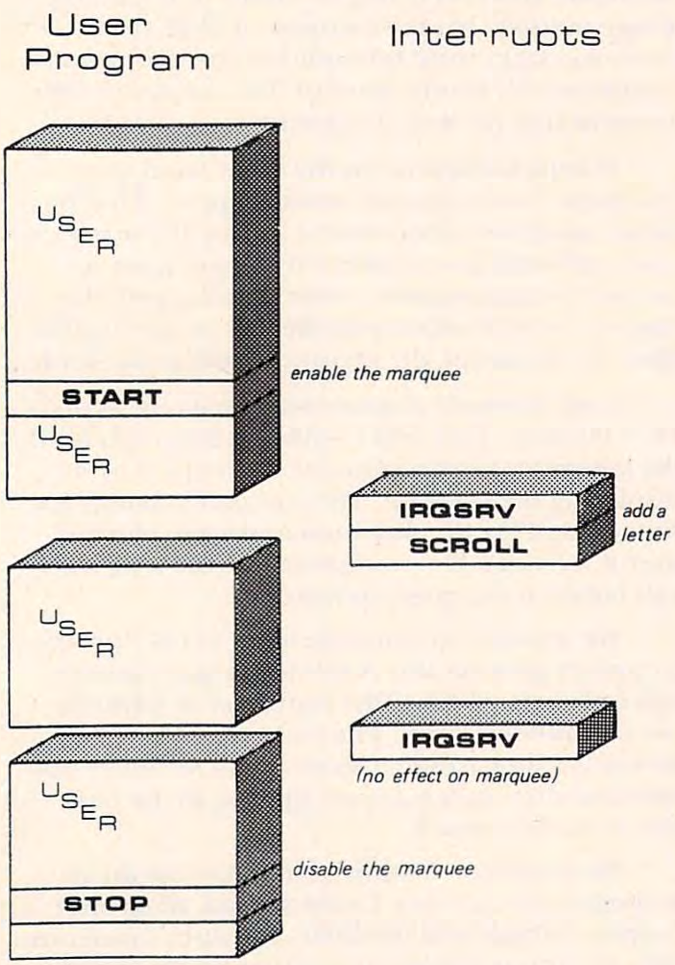


Figure 2 represents the logic of the marquee interrupt system in more detail. A programmable timer creates periodic interrupts (represented as marbles) at regular intervals. When the marquee is inactive, control falls directly into the computer's "normal interrupt handler" which ultimately returns control to the user program.

When the marquee is active, interrupt processing is diverted through IRQSRV, which decides whether or not to update the marquee. If no update is necessary, control passes directly to the normal interrupt handler. If the marquee is to be updated, IRQSRV, invokes SCROLL before allowing control to revert to the normal path.

Figure 2:

In this drawing, interrupts are represented by marbles rolling downhill through troughs. When the marquee is not active, IRQSRV is disconnected from the interrupt system and control passes directly to the normal interrupt handler. Activating the marquee inserts IRQSRV and SCROLL into the interrupt path.

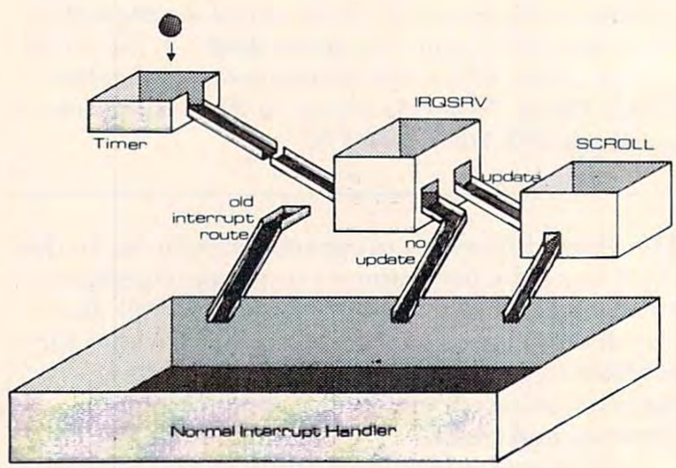
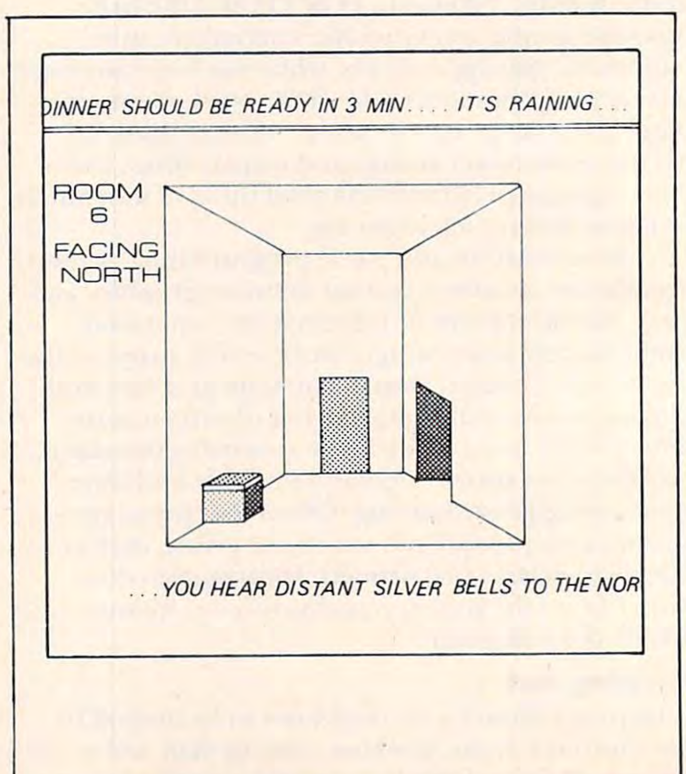


Figure 3.



chips...chips...chips...chips...chips...chips...

Limited Time: FREE Disk-O-Pro.

Special: You can get this Disk-O-Pro **absolutely free**... when you order the new Commodore 2031 single disk drive from Skyles. (See page 8.)

Command-O or Command-O-Pro?

It's called the Command-O-Pro in Europe, Command-O in the U.S.

But whatever you call it, this 4K byte ROM will provide your CBM BASIC 4.0 (4016, 4032) and 8032 computers with 20 additional commands including 10 Toolkit program editing and debugging commands and 10 additional commands for screening, formatting and disk file manipulating. *(And our technical writer dug up 39 additional commands in the course of doing a 76-page manual!)*

Extends Commodore's 8032 advanced screen editing features to the ultimate. **SCROLL** up and down, insert or delete entire lines, delete the characters to the left or right of the cursor, select **TEXT** or **GRAPHICS** mode or ring the PET's bell. Redefine the window to adjust it by size and position on your screen. And define any key to equal a sequence of up to 90 key strokes.

Resides in hexadecimal address $9000, the rightmost empty socket in 4016 and 4032 or the rearmost in 8032. If space conflicts, Socket-2-ME available at a very special price.

Command-O from Skyles Electric Works	$75.00
Complete with **Socket-2-ME**	95.00

Disk-O-Pro ROM:

25 new commands at $75.00, that's only $3.00 a command!

The powerful chip for all "classic" PETs. In any PET with Version III (BASIC 2.0) ROMs (### **COMMODORE BASIC** ###) Disk-O-Pro will give 19 software compatible disk instructions*: 15 identical with the new BASIC 4.0 (or with 8032 ROMs) compatible with both old and new DOS. Plus 4 additional disk commands...including program appending (**MERGE**), program overlaying (**MERGE #**) and **PRINT USING**, allowing formatting output of strings and numbers on the PET screen or on any printer.

* **NOTE:** *Old DOS doesn't recognize 3 commands.*

Plus softtouch key (**SET**) which allows you to define a key to equal a sequence of up to 80 keystrokes; **SCROLL** whereby all keys repeat as well as slow scrolling and extra editing features; **BEEP** which allows you to play music on your PET.

Completely compatible with the BASIC Programmer's Toolkit. *(The chip resides in the socket at hexadecimal address $9000, the rightmost empty socket in most PETS.)* And for the owners of "classic" (or old) PETs, we do have interface boards.

(For those owning a BASIC 4.0 or 8032, even though the Disk-O-Pro may not be suitable, the Command-O is. We have never abandoned a PET owner.)

Complete with 84-page manual written by Greg Yob.

Disk-O-Pro from Skyles Electric Works	$75.00
Complete with **Interface Board** (for "Classic" PETs)	95.00

"The PET with the new PicChip...is like home movies."

The new ROM that took Europe by storm is available only from Skyles Electric Works in the U.S. and Canada.

PicChip, a ROM extension of the BASIC Version III (BASIC 2), BASIC 4.0 or BASIC 8032 interpreter. Over 40 commands that allow you to create programs with dynamic graphics displays: plots, bar graphs, pictures; and rolling, scrolling, shifting and inverting. All instantly and easily added to your BASIC program.

The address for the 2000/3000 (which would require PicChip module PC2), for the 4000 (PC4) and for the 8000 (PC8) is $A000...unless you have a Mikro, Word Pro 3 or 4, or Jinsam, which occupy that same address. In those cases, you will need the PicChip on an interface board... for the 2000/3000 series, PCB2, for the 4000, PCA4, and for the 8000, PCA8. In all cases, the Mikro or WordPro would be switchable manually using the Skyles Socket-2-ME...available at a very special price.

PicChip from Skyles Electric Works (Please indicate PC2, PC4, PC8)	$60.00
Complete with Socket-2-ME (Please indicate PCB2, PCA4, PCA8)	80.00

The New Mikro:

Now you can program in machine language as easily as writing BASIC.

4K machine language assembler ROM plugs into your main board. Does all the machine language work for you; all you have to do is start laying down the code.

Retains all the great screen editing features of the PET... even all the Command-O or Toolkit commands. Write your own machine language subroutine. The program you write is the source code you can save. Machine language monitor saves the object code. Not as professional as the Skyles MacroTeA...not as expensive, either.

A great learning experience for those new to machine language programming but who want to master it easily. Superb 50-page manual—including brief introduction to machine language programming—included. But we also recommend several of the books in the Skyles Book Barn.

Skyles Mikro Machine Language Assembler MK 2/3 for 2000 or 3000 Series PETs MK 4/8 for 4000 or 8000 Series PETs and CBMs	$80.00
Complete with **Socket-2-ME**	95.00

Skyles Catalogue Page 1

For PET and CBM owners only:

This is just 1 of 20 pages of the newest and biggest Skyles catalog, hot off the press.

We know you'll want this page, in its full 8½ x 10 splendor, and another 19 pages of peripherals, software and books that will make your PET or CBM computer even nicer to live with. So, if we missed sending you your very own copy within the last few weeks, call us at **(800) 227-9998** **(unless you live in California, in which case call (415) 965-1735.**

From Skyles Electric Works, the oldest and largest professional specialists in the business.

Skyles Electric Works
231-E S. Whisman Road
Mountain View, California 94041

Program 1 presents an implementation of the marquee system for the Upgrade ROM PET. The user's program calls START when it wants to put a message on the marquee. START initializes several variables and constants and, most importantly, routes all further interrupt requests *via* the marquee update controller IRQSRV.

Program 1.

```
800 FOR ADRES=864TO1023:READ DATTA:
    POKE ADRES,DATTA:NEXT ADRE
    S
864 DATA 40, 0, 5, 85, 228, 7
870 DATA 5, 40, 160, 1, 185, 0
876 DATA 128, 153, 255, 127, 200, 2
    04
882 DATA 96, 3, 208, 244, 32, 161
888 DATA 3, 205, 97, 3, 240, 15
894 DATA 192, 255, 240, 11, 200, 14
    0
900 DATA 101, 3, 172, 96, 3, 153
906 DATA 255, 127, 96, 172, 96, 3
912 DATA 169, 32, 153, 255, 127, 23
    8
918 DATA 103, 3, 173, 103, 3, 205
924 DATA 96, 3, 176, 48, 96, 172
930 DATA 101, 3, 177, 0, 41, 191
936 DATA 96, 141, 0, 0, 142, 1
942 DATA 0, 169, 0, 141, 103, 3
948 DATA 141, 101, 3, 173, 144, 0
954 DATA 141, 99, 3, 173, 145, 0
960 DATA 141, 100, 3, 120, 169, 223

966 DATA 141, 144, 0, 169, 3, 141
972 DATA 145, 0, 88, 96, 120, 173
978 DATA 99, 3, 141, 144, 0, 173
984 DATA 100, 3, 141, 145, 0, 88
990 DATA 96, 206, 102, 3, 16, 9
996 DATA 32, 104, 3, 173, 98, 3
1002 DATA 141, 102, 3, 108, 99, 3
1008 DATA 162, 3, 169, 248, 32, 169
1014 DATA 3, 96, 77, 65, 82, 81
1020 DATA 85, 69, 69, 0, 0, 70
```

The PET's 6522 timer generates interrupts 60 times per second. While the marquee is active, these interrupt requests invoke IRQSRV. This routine decides whether or not it's time to update the marquee; the speed of the marquee display is determined by the variable RATE, which specifies the number of interrupts which will occur between marquee updates. By adjusting RATE, the marquee's progress may be speeded up or slowed down.

If IRQSRV decides not to update the display, it jumps to the computer's normal interrupt handler, whose address is stored in OLDIRQ. If IRQSRV decides to update the display, it calls SCROLL before returning control to the machine's normal procedures.

SCROLL alone actually writes and updates the marquee. SCROLL first moves each character on the marquee line one space to the left. Next, SCROLL calls GETCHAR, which locates the next character in the message. The new character is tacked onto the right-hand edge of the message before SCROLL returns to IRQSRV.

A special character, END (usually 00, the ASCII NUL character), marks the end of each marquee message. When SCROLL encounters the end of a message, it starts tacking blanks onto the end of the marquee line. Eventually all the text will travel off the left edge of the screen, leaving the marquee blank; at this point, SCROLL automatically invokes STOP to disable future marquee updates.

For Other Computers

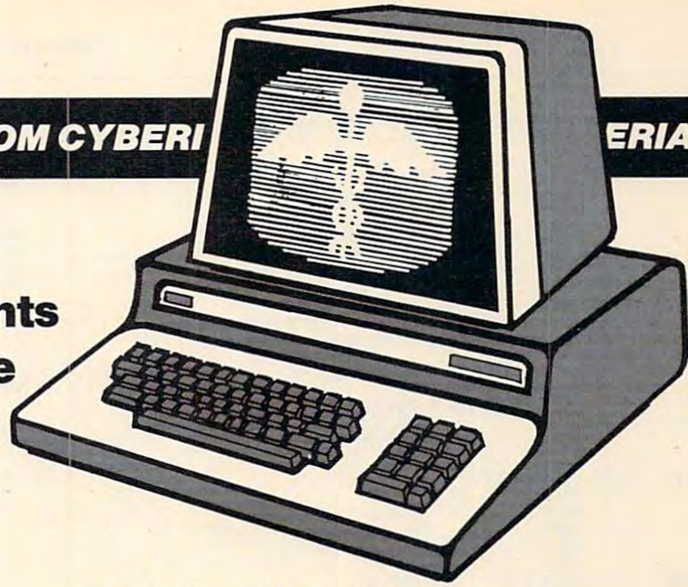
The marquee routines described here can be used on many 6502 systems with little or no change.

Different model PET's are easily accomodated. The only ROM-dependent instruction is the address IRQVEC, the page-zero location through which the PET vectors its interrupts.(It's the same, $90, 81 in 4.0 BASIC. For Original PETs, use $0219, 021A] 80-column computers, of course, can have 80-character marquees; simply change the value in LENGTH to 80.

Other computers should also be able to use this marquee system. The basic requirements are a memory-mapped display and a source of periodic interrupts. Many single-board computers, for example, use the 6522 VIA/timer which does this job admirably. Apple users will need to add an expansion board if one of their current accessories won't do the job. Several Apple parallel port I/O boards include the 6522; additionally, some time-of-day clock boards can generate periodic interrupts to drive the marquee.

Finally, note that marquees might be used in several different ways. They need not occupy an entire line; to use only a part of a line, simply change LINE (the address of the left end of the marquee) and LENGTH (the length of the marquee). The marquee may appear anywhere on the screen, although the top (used here) and bottom lines are likely to be most popular. Several marquees might appear on the same screen! Finally, note that marquees may move very rapidly (for speed reading practice?), and are not limited to text, suggesting several interesting possibilities for unusual graphics.

NEW FROM CYBERIA ★ NEW FROM CYBERIA ★ NEW FROM CYBERIA



Mari

A comprehensive accounts receivable and insurance billing system for modern health care offices and clinics

DESIGNED FOR COMMODORE 8000 SERIES COMPUTERS AND DISK DRIVES

- Help functions are always on-line
- Supports CPT, ICD and RVS medical coding
- Supports open item or balance forward accounting systems
- Interfaces to popular word processing programs
- Multiple terminals may be added with Superbus 4.0
- Multiple disk drives may be used—no limit on number of patients or accounts
- Includes a data base and forms generator to fill out any insurance form
- Includes a computer aided instruction program to train new users

Includes these standard reports:

- A/R aging
- A/R transactions
- General ledger
- Instant cash receipt
- Income analysis by physician
- Doctor referral report
- Patient/account cross reference
- Standard SuperBill insurance form

NEW FROM CYBERIA ★ NEW FROM CYBERIA

NEW ★ NEW ★ NEW

SUPERBUS 4.0

Commodore Computer Networking System

- Up to 18 computers can be interconnected
- Multiple disk drives, printers and other devices may be added to the network
- Sophisticated security system prevents unauthorized use
- WordPro and Wordcraft programs are supported to allow multiple terminal word processing

*WordPro is a trademark of Professional Software, Inc.

CYBERIA INC. 2330 LINCOLN WAY, AMES, IA 50010 • 515-292-7634

Program 2.

```
                0010              .BA $7000
                0020              .OS
                0500 ;================================
                0501 ;    GLOBAL VARIABLES
                0502 ;================================
                0503 ;
                0504 ;
                0510 ; LINE = ADDRESS OF LEFT END OF
                0511 ;        THE MESSAGE DISPLAY LINE
                0512 ;
                0513 LINE      .DE $8000
                0514 ;
                0520 ; LENGTH = NUMBER OF CHARACTERS
                0521 ;          PER LINE FOR THIS
                0522 ;          COMPUTER'S VIDEO DISPLAY
                0523 ;
7000- 28        0524 LENGTH    .BY 40
                0525 ;
                0530 ; END  =  SYMBOL TO INDICATE
                0531 ;         THE END OF THE MESSAGE
                0532 ;         ( ASCII NUL )
                0533 ;
7001- 00        0534 END       .BY 00
                0535 ;
                0540 ; RATE =  NUMBER OF 1/60'THS SECONDS
                0541 ;         TO ELAPSE BETWEEN DIS-
                0542 ;         PLAY UPDATES.
                0543 ;
7002- 05        0544 RATE      .BY 5
                0545 ;
                0550 ; IRQVEC = ADDRESS WHERE THE
                0551 ;          COMPUTER STORES THE
                0552 ;          ADDRESS OF ITS INTERRUPT
                0553 ;          SERVICE ROUTINE.
                0554 ;
                0556 IRQVEC    .DE $90
                0557 ;
                0560 ; OLDIRQ = BUFFER FOR STORING THE
                0561 ;          COMPUTER'S NORMAL
                0562 ;          NORMAL INTERRUPT SERVICE
                0563 ;          ROUTINE.
                0564 ;
7003- 00 00     0565 OLDIRQ    .BY 0 0
                0566 ;
                0570 ; POINTR = PAGE ZERO POINTER
                0571 ;          TO THE START OF THE
                0572 ;          STRING WHICH THE
                0573 ;          COMPUTER WILL DISPLAY.
                0574 ;
                0575 POINTR    .DE 0
                0576 ;
                0580 ; CHAR = COUNT OF CHARACTERS
                0581 ;        WRITTEN TO THE MARQUEE
                0582 ;        TO DATE.  RESET BY START,
                0583 ;        UPDATED BY SCROLL, USED
                0584 ;        BY GETCHAR.
                0585 ;
7005- 00        0586 CHAR      .BY 0
                0587 ;
                0590 ; IRQCNT = COUNT OF INTERRUPT
                0591 ;          REQUESTS TO BE SKIPPED
                0592 ;          BEFORE ADVANCING THE
                0593 ;          MARQUEE.  USED ONLY BY
                0594 ;          IRQSRV.
                0596 ;
7006- 00        0597 IRQCNT    .BY 0
                0598 ;
                0599 ;==================================
                0600 ; AFTER = COUNT OF CALLS TO 'DONE'
                0601 ;         'DONE' APPENDS BLANKS TO
                0602 ;         THE MARQUEE UNTIL THE
                0603 ;         ENTIRE MARQUEE IS BLANK.
                0604 ;         'DONE' THEN DISABLES THE
                0605 ;         MARQUEE BE INVOKING
                0606 ;         'STOP'.
                0607 ;
7007- 00        0608 AFTER     .BY 0
                0609 ;
                0999 ;--------------------------------
                1000 ; SCROLL
                1001 ;
                1002 ; DISPLAY A 'TIMES-SQUARE' STYLE
                1003 ;   MESSAGE LINE.
                1004 ; 'SCROLL' IS CALLED PERIODICALLY
                1005 ;  BY AN INTERRUPT SERVICE
                1006 ;  ROUTINE, AND ADVANCES THE
                1007 ;  DISPLAY ONE NOTCH PER CALL.
                1008 ;
                1009 ;
                1010 ; SCROLL IS NOT USUALLY CALLED
                1011 ;  BY THE USER.
                1012 ;
                1013 ; INSTEAD, THE DISPLAY IS TUR'ED ON
                1014 ;  BY CALLING 'START'.
                1015 ;
                1016 ; AFTER THE ENTIRE MESSAGE IS
                1017 ;  DISPLAYED, THE DISPLAY ROUTINE
                1018 ;  TURNS ITSELF OFF.  IT CAN BE
                1019 ;  DEACTIVATED AT ANY TIME BY
                1020 ;  CALLING 'OFF'.
                1038 ;--------------------------------
                1039 ;
                1140 SCROLL
                1150 ;
7008- A0 01     1160          LDY #1        ; LEFT-MOST CHARACTER
                1170 ;
                1180 SCROLL1
700A- B9 00 80  1190          LDA LINE,Y
700D- 99 FF 7F  1210          STA LINE-1,Y
7010- C8        1220          INY
7011- CC 00 70  1240          CPY LENGTH    ; MOVED ENTIRE LINE ?
7014- D0 F4     1250          BNE SCROLL1   ; REPEAT 'TIL DONE
                1260 ;
                1270 ; GET THE NEXT CHARACTER
                1271 ;   TO BE DISPLAYED
                1272 ;
7016- 20 41 70  1280          JSR GETCHAR
                1290 ;
                1291 ; GETCHAR RETURNS THE NEXT CHARAC-
                1292 ;  TER IN A AND THE TOTAL NUMBER
                1293 ;  OF CHARACTER DISPLAYED SO FAR
                1294 ;  IN Y.
                1295 ;
                1296 ; NOW WE CHECK FOR THE END OF
                1297 ;  THE MESSAGE, WHICH HAPPENS
                1298 ;  AFTER THEN 'END' CHARACTER
                1299 ;  OR AFTER 256 CHARACTER HAVE
                1300 ;  BEEN DISPLAYED.
                1301 ;
7019- CD 01 70  1304          CMP END
701C- F0 0F     1310          BEQ DONE
                1312 ;
701E- C0 FF     1320          CPY #$FF
7020- F0 0B     1322          BEQ DONE
                1324 ;
                1330 ; UPDATE AND SAVE CHARACTER POINTER
7022- C8        1340          INY
7023- 8C 05 70  1350          STY CHAR
                1360 ;
                1370 ; PUT THE NEXT CHARACTER ON SCREEN
7026- AC 00 70  1380          LDY LENGTH
7029- 99 FF 7F  1400          STA LINE-1,Y
                1410 ;
702C- 60        1470          RTS
                1500 ;-----------------
                1501 ; DONE
                1502 ;-----------------
                1503 ;
                1520 DONE
702D- AC 00 70  1530          LDY LENGTH    ; END OF LINE
7030- A9 20     1540          LDA #$20      ; BLANK
7032- 99 FF 7F  1550          STA LINE-1,Y
                1560 ;
                1561 ; INCREMENT AFTER
                1562 ; QUIT WHEN AFTER>LENGTH
                1563 ;
7035- EE 07 70  1570          INC AFTER
7038- AD 07 70  1580          LDA AFTER
703B- CD 00 70  1590          CMP LENGTH
703E- B0 30     1600          BCS STOP
                1610 ;
7040- 60        1620          RTS
                1800 ;-------------------------
                1801 ; GETCHAR
                1802 ;-------------------------
                1803 ;
                1804 ; GET NEXT CHARACTER FROM
                1805 ; MESSAGE STRING
                1806 ;
                1807 ; RETURNS THE CHARACTER IN A
                1808 ;  AND THE TOTAL CHARACTER COUNT
                1809 ;  IN Y.
                1810 ;
7041- AC 05 70  1830 GETCHAR  LDY CHAR
7044- B1 00     1840          LDA (POINTR),Y
                1850 ;
                1851 ; THE FOLLOWING CONVERSION APPLIES
                1852 ; ONLY TO THE COMMODORE PET.
                1853 ; IT TRANSLATES ASCII STRINGS INTO
                1854 ; THE PET'S "SCREEN CODE"
                1856 ;
7046- 29 BF     1860          AND #$BF
                1870 ;
7048- 60        1880          RTS
                3000 ;------------
                3001 ;  START
                3002 ;------------
                3003 ;
                3004 ;  SET UP A NEW MESSAGE TO BE
                3005 ;  SCROLLED ACROSS THE SCREEN.
                3006 ;
                3007 ;  THE FOLLOWING REGISTERS MUST
                3008 ;  BE LOADED BEFORE CALLING START
                3009 ;
                3010 ;  A : LSB OF STRING ADDRESS
                3011 ;  X : MSB OF STRING ADDRESS
                3012 ;
                3013 ;
                3014 ;  THE DISPLAY WILL PROCEED
                3015 ;  AUTOMATICALLY UNTIL THE
                3016 ;  END-OF-MESSAGE CHARACTER
                3017 ;  (STORED IN 'END') IS FOUND.
                3018 ;  AFTER THE COMPLETE MESSAGE HAS
                3019 ;  BEEN DISPLAYED; THE DISPLAY
                3020 ;  WILL TURN ITSELF OFF.
                3021 ;  THE DISPLAY CAN BE DISABLED
```

SYSRES™

THE ULTIMATE RESIDENT PROGRAM MANIPULATION SYSTEM FOR PET™/CBM™ MICROCOMPUTERS

SYSRES™ EXTENDED DOS SUPPORT COMMANDS

@ (type "N" keyboard)
← (type "B" keyboard)
! (original keyboard)
> (for 'wedge' users)

These commands may be used interchangably, to perform the following dos support functions.

Disk	Printer	Tape	Directory	Modes	Command	Function
x				3	@	Display disk status / send command
x					@N	Format (header) a new diskette
x					@I	Force initialize diskette
x					@V	Validate diskette (collect)
x					@D	Duplicate diskette
x			x	4	@C	Copy or concatenate disk file(s)*
x					@R	Rename file
x			x	3	@S	Scratch file(s)*
x					@$	List directory**
x					@U:	Reset disk drive
x	x	x	x	6	@L	List disk file or BASIC program**

Note: Some of the disk utility command set may also be used, if an appropriate direct access channel has been opened.

* Standard command with added options.
** Added disk command.

SYSRES™ EXTENDED EDITOR COMMANDS

Disk	Printer	Tape	Directory	Modes	Command	Function
x			x	4	/	Quick load from disk
x			x	4	↑	Quick load from disk with auto run
x			x	2	APPEND	Append from disk to end of current program
				4	AUTO	Auto line number (allows header)
x			x	3	BLOAD	Load machine language (binary) file
x			x	3	BRUN	Load and execute machine language program
	x			776	CHANGE	Change pattern to another pattern
				2	CLOSE	Close one or all files
				1	CMD	Set output to file (does not send "READY.")
				4	DELETE	Delete a range of lines from program
	x			1	DUMP	Dump all scalar variables to screen or file
x			x	2	EXEC	Execute a file as keyboard commands
	x			240	FIND	Find occurances of a pattern
x		x	x	3	GET	Read a sequential file into editor
				7	KEY	Define a key as a special function
				1	KEYS	Turn key functions on
				1	KILL	Disable SYSRES™
				1	KILL*	Disable SYSRES™ and unreserve memory
	x			10	LIST	Improved BASIC LIST command
x		x	x	3	LOAD	Defaults to disk drive
x			x	2	MERGE	Merge from disk into current program
	x			1	MON	Break to current machine language monitor
				1	OLD	Restore program after "NEW"
x	x	x	x	24	PUT	Send program to disk as text file
				6	RENUMBER	Renumber all or part of program
				2	RUN	Run current program, ignores screen garbage
x		x	x	3	SAVE	Defaults to disk drive, allows replace
x		x		1	SETD	Set disk device #, allows multiple drives
	x			4	SETP	Set printer channel, format mode, paging
				4	TRACE	Select 1 of 3 trace/step modes and speed
x		x	x	3	VERIFY	Compare current program against disk/tape
				1	WHY	Print position of last error
				1	WHY?	List line of break or error
x	x				*	Send output to printer
	x			1	#	Display current version of SYSRES™

JUST A FEW OF THE FEATURES OF SYSRES™

- * Fast up/down scrolling which works!
- * Advanced repeat-key routine!
- * Re-define any or all keys as any keyword (full or short form) or as any string up to 255 charactors long!
- * Auto line numbering which can feed a string of up to 127 charactors as well!
- * Extended DOS support (requires DOS 2A or greater)!
- * Never enter another file name! All file commands work from the directory!
- * Supports multiple disk drives!
- * List BASIC programs, sequential and relative files without loading them into memory!
- * TRUE PROGRAM MERGE (overlay). Supports subroutine libraries!
- * Load and run machine language programs with parameter passing!
- * Supports multiple printers!
- * Automatic printer output with paging plus formatted listings with full ASCII code conversion including cursor control and special charactors for non-CBM™ printers!
- * Edit text files and assembler source code without leaving BASIC!
- * Renumber part of a program or even change the order of lines!
- * Over 700 FIND/CHANGE commands including variable names ("A$" will not match "BA$"), pattern matching with "wild-cards", and even commands to remove spaces and REM's!
- * Three TRACE modes including trace variables!
- * Does not affect BASIC program operation!
- * One AUTO-BOOT DISKETTE works for ALL PET™ or CBM™ computers (BASIC 2.0 or greater with at least 16k of RAM.). SYSRES™ requires NO ROM SPACE or extra boards, so you can take it with you if you want to use another computer. It may be put above the screen if you have RAM there. It boots automatically without disturbing any program in RAM!
- * Diskette and Extensive Manual – only $75 (cdn)
Documented Source Code (YES!) – $150 (cdn)
Custom versions are available for unique systems!

Dealer enquiries are welcome.

VISA MasterCard

CANSOFT

CANSOFT DATA INC.

MICROCOMPUTER SERVICES

PROGRAMMING * CONSULTING * DESIGNING * EVALUATING

BOX 76752, STN.S, VANCOUVER, B.C. V5R 5S7 TEL:(604)437-6667

```
                3022 ;  BY CALLING 'STOP' AT ANY TIME.
                3023 ;
                3024 ;---------------------------------
                3025 ;
                3026 START
                3027 ; STORE THE STRING'S ADDRESS
                3028 ;
7049- 8D 00 00  3030           STA POINTR
704C- 8E 01 00  3040           STX POINTR+1
                3050 ;
                3051 ; RESET 'AFTER' AND 'CHAR'
                3052 ;
704F- A9 00     3060           LDA #0
7051- 8D 07 70  3070           STA AFTER
7054- 8D 05 70  3072           STA CHAR
                3080 ;
                3081 ; SAVE OLD IRQ SERVICE VECTOR
                3082 ;
7057- AD 90 00  3090           LDA IRQVEC
705A- 8D 03 70  3100           STA OLDIRQ
705D- AD 91 00  3110           LDA IRQVEC+1
7060- 8D 04 70  3120           STA OLDIRQ+1
                3130 ;
                3131 ; SET UP NEW IRQ VECTOR
                3132 ;
7063- 78        3138           SEI
7064- A9 7F     3140           LDA #L,IRQSRV
7066- 8D 90 00  3150           STA IRQVEC
7069- A9 70     3160           LDA #H,IRQSRV
706B- 8D 91 00  3170           STA IRQVEC+1
706E- 58        3180           CLI
                3190 ;
706F- 60        3200           RTS
                3500 ;--------------------
                3501 ; STOP
                3502 ;--------------------
                3503 ;
                3504 ; DISABLE THE AUTOMATIC MESSAGE
                3505 ;  DISPLAY.
                3506 ;
                3507 ; STOP CAN BE CALLED DIRECTLY BY
                3508 ;  THE USER.  IT IS ALSO INVOKED
                3509 ;  BY 'DONE' WHEN THE COMPLETE
                3510 ;  MESSAGE HAS BEEN DISPLAYED.
                3511 ;
                3520 STOP
                3521 ;
                3522 ; RESTORE THE ORIGINAL IRQ VECTOR
                3523 ;
7070- 78        3530           SEI
7071- AD 03 70  3540           LDA OLDIRQ
7074- 8D 90 00  3542           STA IRQVEC
7077- AD 04 70  3550           LDA OLDIRQ+1
707A- 8D 91 00  3552           STA IRQVEC+1
707D- 58        3560           CLI
                3570 ;
707E- 60        3580           RTS
                4000 ;-----------------
                4001 ; IRQSRV
                4002 ;-----------------
                4003 ;
                4004 ; THIS ROUTINE IS CALLED WHENEVER
                4005 ;
                4006 ;    A) THE PET TIMER CREATES AN
                4007 ;        INTERRUPT REQUEST
                4008 ;
                4009 ;    B)  THE MESSAGE ROUTINE HAS
                4010 ;         BEEN ENABLED BY CALLING
                4011 ;         'START',  AND HAS NOT
                4012 ;         YET BEEN DISABLED BY
                4013 ;         CALLING 'STOP'.
                4014 ;
                4015 ; THE PET TIMER REQUESTS AN
                4016 ; INTERRUPT 60 TIMES PER SECOND.
                4017 ;
                4022 ;
                4029 ;---------------------------------
                4030 ;
                4031 IRQSRV
707F- CE 06 70  4040           DEC IRQCNT
7082- 10 09     4050           BPL NORMAL
                4060 ;
                4061 ; CALL DISPLAY UPDATE
                4062 ;
7084- 20 08 70  4070           JSR SCROLL
                4080 ;
                4081 ; RESET IRQ COUNTER
                4082 ;
7087- AD 02 70  4090           LDA RATE
708A- 8D 06 70  4100           STA IRQCNT
                4110 ;
                4120 ; EXIT THROUGH THE STANDARD
                4121 ;  INTERRUPT SERVICE ROUTINE, WHOSE
                4122 ;  ADDRESS IS STORED IN 'OLDIRQ'.
                4123 ;
                4130 ;
708D- 6C 03 70  4140 NORMAL    JMP (OLDIRQ)
                5000 ;-------------
                5001 ; SAMPLE PROGRAM
                5002 ;-------------
                5003 ;
                5005 ;
                5006 TEST
                5007 ; LOAD THE ADDRESS OF THE MESSAGE
                5008 ; TO BE DISPLAYED INTO THE X AND
                5009 ; A REGISTER.
                5010 ;
7090- A2 70     5012           LDX #H,STRING
7092- A9 98     5020           LDA #L,STRING
                5022 ;
                5030 ;
                5031 ; CALL 'START' TO BEGIN DISPLAY
                5032 ;
7094- 20 49 70  5040           JSR START
                5050 ;
7097- 60        5110           RTS
                9000 ;-------------------------------
                9001 ;
                9002 ; STRING = SAMPLE STRING FOR
                9003 ;          TEST PROGRAM.
                9004 ;
                9005 ;-------------------------------
                9006 ;
7098- 4D 41 52  9010 STRING    .BY 'MARQUEE DISPLAYS ARE '
709B- 51 55 45
709E- 45 20 44
70A1- 49 53 50
70A4- 4C 41 59
70A7- 53 20 41
70AA- 52 45 20
70AD- 45 41 53  9011           .BY 'EASY TO PROGRAM AND '
70B0- 59 20 54
70B3- 4F 20 50
70B6- 52 4F 47
70B9- 52 41 4D
70BC- 20 41 4E
70BF- 44 20
70C1- 43 4F 4E  9012           .BY 'CONVENIENT FOR THE USER.
70C4- 56 45 4E
70C7- 49 45 4E
70CA- 54 20 46
70CD- 4F 52 20
70D0- 54 48 45
70D3- 20 55 53
70D6- 45 52 2E
70D9- 20 20
70DB- 54 48 45  9013           .BY 'THEY ALLOW LOTS OF INFOR'
70DE- 59 20 41
70E1- 4C 4C 4F
70E4- 57 20 4C
70E7- 4F 54 53
70EA- 20 4F 46
70ED- 20 49 4E
70F0- 46 4F 52
70F3- 4D 41 54  9014           .BY 'MATION TO BE DISPLAYED IN
70F6- 49 4F 4E
70F9- 20 54 4F
70FC- 20 42 45
70FF- 20 44 49
7102- 53 50 4C
7105- 41 59 45
7108- 44 20 49
710B- 4E 20
710D- 41 20 53  9015           .BY 'A SMALL SCREEN AREA.' 0
7110- 4D 41 4C
7113- 4C 20 53
7116- 43 52 45
7119- 45 4E 20
711C- 41 52 45
711F- 41 2E 00
                9999           .EN

LABEL FILE:  [ / = EXTERNAL ]

/LINE=8000            LENGTH=7000           END=7001
RATE=7002             /IRQVEC=0090          OLDIRQ=7003
/POINTR=0000          CHAR=7005             IRQCNT=7006
AFTER=7007            SCROLL=7008           SCROLL1=700A
DONE=702D             GETCHAR=7041          START=7049
STOP=7070             IRQSRV=707F           NORMAL=708D
TEST=7090             STRING=7098
//0000,7122,7122
```

©

DR. DALEY Introduces...

THE WIZ

Data Management System

THE WIZ is here!!!

THE WIZ, a system powerful enough to manage most of your data storage and manipulation needs - yet is easy to use. A system we are so sure of that we have an offer you can't resist. First though, let's take a look at a few of the many features of this program.

Feature	Benefit
1. ON-LINE help	At your fingertips is the equivalent of a 60 page manual. At any time the computer is waiting for a response from you, you may press the 'h' key or type 'help'. THE WIZ will then provide you with an explanation of the function you are working with.
2. Plotting capability	This is a feature unique to THE WIZ. It can produce a bar graph with up to 18 bars or a histogram with up to 100 points plotted. Graphically presented data is easy to interpret.
3. Wordpro interface	This option is standard with THE WIZ. With many of the competing data managers, if available, it is an extra cost option.
4. Read a sequential file	You may reorganize your files or even read sequential files generated by other data management systems.
5. Search for keywords	Here you can search for a word in ANY field in your record. It can even ignore differences due to upper case and lower case characters.
6. Constants in data entry	You may store up to three separate sets of contant fields. Each set can have as many fields as you like filled with information. Then two keystrokes will call the appropriate set.

And there is more. There is not room enough to tell you all the features in a one page ad!

SPECIAL OFFER

You will like our system. In fact we are so convinced of this, that we are going to pay you to try it! If you have another commercially available data management program, you can receive TRADE-IN credit for your purchase of THE WIZ. Call us on our toll free number (800) 548-3289 for our offer on your present system. Remember that this offer expires March 15, 1982.

DR. DALEY'S SOFTWARE

Water St. Darby, MT. 59829

(800) 548-3289 (for Orders & Information)

(406) 821-3924 (Montana and technical assistance)

Call between 9 a.m. and 6 p.m. Mountain time Monday through Friday

Disk Disassembler

George H. Watson, Jr.
Physics Dept.
University of Delaware
Newark, DE

Editor's Note: This program works on either BASIC 3.0 or 4.0 and any 2040 DOS. It uses a printer. On some systems, the question "SKIP BASIC?" should be answered "NO" even though the program under disassembly is entirely in machine language.

There are several fine disassemblers available (in BASIC and in machine language) which disassemble programs while they reside in PET memory. Problems arise though when the program to be disassembled normally resides in the same memory space allocated to the disassembler. By relocating the disassembler (moving it to different memory space) it may still be used, although with a bit more difficulty. This problem may be circumvented by using a disassembler which does not require the program to be in PET memory. Instead, the program can be disassembled directly from the diskette on which it is stored by transferring the machine code byte-by-byte (reading the program) and translating into mnemonics, but not storing the bytes in memory.

A computer program is a set of instructions which are stored in the computer's memory in the form of bytes (8-bit words). A machine language program is a set of bytes which the microprocessor in your computer understands directly. On the other hand, a BASIC program consists of bytes which represent the various BASIC statements. When you RUN a BASIC program, each byte is interpreted and the microprocessor acts according to machine language subroutines which exist in the computer's ROMs. When you LIST a BASIC program, the operating system of your computer translates the bytes into BASIC statements, which are then displayed on the screen. Unfortunately no such LIST command is available for machine language programs on the PET microcomputer. But something *is* available which will translate the bytes into a form more understandable to a human. By allowing a disassembler to operate on the machine code, the program will be "LISTED" as 6502 microprocessor mnemonics, the heart of every PET.

DISK DISASSEMBLER opens a file to be read (the program to be disassembled) in the disk drive. The first two bytes which are read will contain the address at which the file is normally loaded into PET memory. The remaining bytes to be read comprise the program. All bytes will be translated into mnemonics until an end-of-file marker is detected (through the error word, ST), at which point the disassembly is finished.

Many programs which you may be interested in disassembling will be a combination of BASIC and machine code. DISK DISASSEMBLER handles the case where the machine code follows the BASIC program. All bytes are skipped over until three consecutive zeroes are detected which indicates the end of the BASIC program. All subsequent bytes will be disassembled.

As much as possible, I have attempted to make the output resemble the source code used by assemblers. (Source code for an assembler consists of the mnemonics for the microprocessor which the assembler converts into machine code.) One major benefit of an assembler is its ability to represent addresses with labels. Thus the machine language programmer is not required to calculate relative addresses needed for conditional branches — a tedious chore. DISK DISASSEMBLER does not provide the option of inputting labels (too time-consuming) but relative branches ARE converted to absolute branches, which makes understanding the disassembly easier.

DIS TEST is a compilation of all legal opcodes (instructions) available to the 6502 microprocessor. When disassembled, an alphabetical listing of the mnemonics along with their addressing modes will be printed out. If there are errors in the mnemonics or addressing modes, carefully check the DATA statements in lines 9000-9155. If the relative branches are wrong, check lines 670-675. Check all lines containing the address counter, AD, if the memory locations in the first column are incorrect.

Try DISK DISASSEMBLER on your favorite game or utility. You can learn much about machine language programming by studying the tricks used by others. You may also be able to learn more about the routines available in the PET's ROMs by examining how other programmers use them.

One option available in DISK DISASSEMBLER is the ability to change a legal opcode to an illegal opcode. Why do this? Some programs which you may disassemble use a legal opcode (unused otherwise) as filler between subroutines. I suppose this is to thwart disassembly since a simple NOP would also do the job. You may overcome this limitation by making the opcode illegal. How? Find the mnemonic in the DATA statement; make sure you find the one with the correct addressing mode. Now simply replace the number immediately following the mnemonic with a zero.

DISK DISASSEMBLER was written on a 32K PET (3.0) with 2040 disk drive. The program as written is slightly less than 7K in length, while variables, arrays, and strings require slightly less than 8K, so the program will run on a 16K PET; remove the REM statements if there is a problem. DISK DISASSEMBLER will also run on 4.0 PETs and with the new disk drive ROMs. For readers not inclined to type in long programs, contact me at the above address and I will provide tape copies at $3 each. (Include SASE, mailer, and tape.) Happy disassembling!

Speeding up BASIC

Some notes on DISK DISASSEMBLER:

1. Most frequently-used subroutines and the working part of the program should be placed at the beginning of the program (lower line numbers). When a GOSUB or GOTO is executed, BASIC begins at the first line of the program and compares each following line number until a match is obtained with the desired line number. Thus fewer line numbers need to be scanned for subroutines which are placed at the beginning. Disadvantage: a program may seem less structured.

2. Variables should be dimensioned as in lines 2000–2020 and the most-used variables should be initialized first. Similar to 1), when a variable is encountered, BASIC begins at the first variable in the table of variables and compares each following variable with the desired variable until a match is made. Dummy variables (constantly changing value and heavily used in subroutines) are good candidates for the first positions in the table. The variables should then be used as often as permitted.

3. When possible, use arrays of constants in place of conversions made with time-consuming subroutines. The biggest timesaving in DISK DISASSEMBLER was made by using an array of 256 hex characters, HG$( ), in place of a subroutine which converted the decimal value of a byte to the hex value. Disadvantage: more memory consumed.

4. Use IF FG THEN ... rather than IF FG<>0 THEN ... and IF ST-64 THEN ... rather than IF ST<>64 THEN ... The branch will be made if the argument of the IF .. THEN .. is nonzero.

5. Replace numbers with defined variables. In lines 300 and 400, B = 256. Time is saved since the conversion of the number 256 into the representation used by BASIC need not be done over and over; it was done once at initialization. Disadvantage: larger variable table.

I would also like to mention two shorthand tricks which are available.

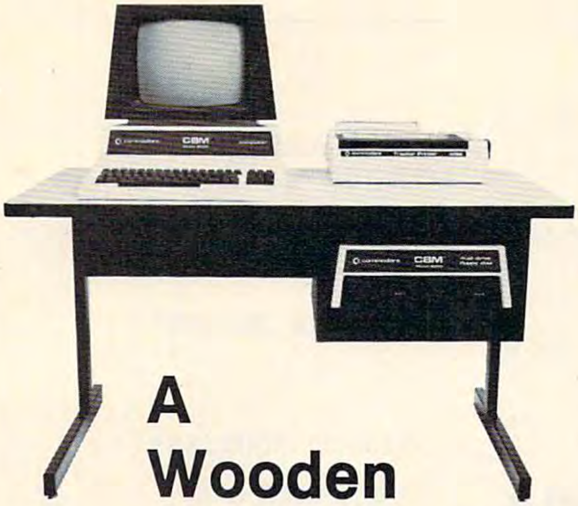
6. Since any statement following a GOTO or RETURN on the same line is never executed, a remark may be placed there with no time lost and with no REM statement. See lines 10 and 100.

7. When DATA statements are read, if all that is seen is another comma (no data), then a variable is read to be zero and a string is read to be null.

Program 1.

```
10 GOTO1000:                                             ******* COMMENTS *******
100 IFST-64THENRETURN:                                   *CHECK FOR END-OF-FILE*
110 FG=1:RETURN
200 GET#5,D$:GOSUB100:IFD$=""THEND=0:D$="00":RETURN:*BYTE GET & CONVERSION*
210 D=ASC(D$):D$=H$(D):RETURN
300 A%=AD/B:AD$=H$(A%)+H$(AD-A%*B):RETURN:               *ADDRESS CONVERSION*
400 A%=D/B:C$=H$(A%)+H$(D-A%*B):RETURN:                  *DECIMAL -> 4-DIGIT HEX*
490 TI$="000000":REM                                     *BEGIN DISASSEMBLY*
500 IFFGTHENRETURN:                                       CHECK END-OF-FILE FLAG
510 GOSUB200:ONB%(D)GOTO540,600,700:REM                   GET 1ST BYTE & BRANCH
520 REM                                                  *1-BYTE INSTRUCTION*
530 D$=D$+"*":M$="":GOTO550:                              -ILLEGAL OPCODE
540 M$=M$(D):REM                                          -ACCUMULATOR,IMPLIED
550 PRINT#4,AD$"  "D$"     ",M$
560 AD=AD+1:GOSUB300:GOTO500
590 REM                                                  *2-BYTE INSTRUCTION*
600 B1=D:B1$=D$:M$=M$(D)+" ":GOSUB200:REM                 GET 2ND BYTE
605 ONA%(B1)GOTO610,620,630,640,650,660,670:              ADDRESSING MODE
610 M$=M$+"# $"+D$:GOTO680:                               -IMMEDIATE
620 M$=M$+"* $"+D$:GOTO680:                               -ZERO PAGE
630 M$=M$+"($"+D$+",X)":GOTO680:                          -INDEXED INDIRECT
```

```
640 M$=M$+"($"+D$+"),Y":GOTO680:                    -INDIRECT INDEXED
650 M$=M$+"* $"+D$+",X":GOTO680:                    -ZERO PAGE INDEXED BY X
660 M$=M$+"* $"+D$+",Y":GOTO680:                    -ZERO PAGE INDEXED BY Y
670 IFD<128THEND=AD+D+2:GOTO675:                    -RELATIVE -> ABSOLUTE
672 D=AD+D-254
675 GOSUB400:M$=M$+"TO $"+C$
680 PRINT#4,AD$"  "B1$" "D$"   ",M$
690 AD=AD+2:GOSUB300:GOTO500
695 REM                                            *3-BYTE INSTRUCTION*
700 B1=D:B1$=D$:GOSUB200:B2$=D$:GOSUB200:REM        GET 2ND & 3RD BYTES
710 M$=M$(B1)+" $"+D$+B2$
720 ONA%(B1)GOTO760,730,740,750:                    ADDRESSING MODE
730 M$=M$+",X":GOTO760:                             -ABSOLUTE INDEXED BY X
740 M$=M$+",Y":GOTO760:                             -ABSOLUTE INDEXED BY Y
750 M$=LEFT$(M$,4)+"($"+D$+B2$+")":REM              -INDIRECT
760 PRINT#4,AD$"  "B1$" "B2$" "D$,M$:REM            -ABSOLUTE
770 AD=AD+3:GOSUB300:GOTO500
780 :
1000 GOSUB2000:REM INITIALIZE                      *BEGIN EXECUTION*
1100 GOSUB2100:REM SELECT PRINTER
1200 GOSUB2200:REM SELECT FILE
1300 GOSUB2300:REM OPEN FILE
1350 IFFETHEN1200: DISK ERROR
1400 GOSUB2400:REM GET LOAD ADDRESS
1500 GOSUB2500:REM SKIP BASIC
1600 GOSUB490: REM DISASSEMBLE FILE
1700 GOSUB2600:REM STOP?
1800 GOTO1100: REM REPEAT
1900 :
2000 DIMD,D$,AD,A%,B,B1,FG,C$,J:B=256:REM           *INITIALIZATION*
2010 DIMAD$,M$,B1$,B2$,DR$,FL$,FY,FP
2020 DIMDV,FD,DA$,FE,EN,EN$,EM$,ES$,ET$
2030 DIMD$(15),H$(255),M$(255),B%(255),A%(255),C$(13)
2040 FORJ=0TO15:READD$(J):NEXT:REM                   FILL ARRAYS
2050 FORJ=0TO13:READC$(J):NEXT
2060 PRINT"ĥr>>"C$(0)C$(10):PRINT"↓"C$(11):PRINT"↓"C$(12)
2070 FORJ=0TO15:FORD=0TO15:H$(J*16+D)=D$(J)+D$(D):NEXT:NEXT
2080 FORJ=0TO255:READM$(J),B%(J),A%(J):NEXT:RETURN
2090 :
2100 IFFPTHENRETURN:                               *OPEN PRINTER*
2110 D=1:GOSUB6000:DV=A%:IFDV<3ORDV>30THEN2110
2120 FP=1:CLOSE4:OPEN4,DV:IFDV-3THENPRINT#4
2130 IFFDTHENRETURN:                                ENTER DATE
2140 D=2:GOSUB6000:DA$=D$:FD=1:RETURN
2150 :
2200 D=3:GOSUB6000:DR$=D$:IFA%ANDA%-1THEN2200:REM   *SELECT FILE*
2210 D=4:GOSUB6000:FL$=D$:IFDV=4ORDV=3THENPRINT#4,"ĥ
2220 PRINT#4,FL$,,DA$:PRINT#4:PRINT#4:RETURN
2230 :
2300 CLOSE15:CLOSE5:REM                            *INITIALIZE DISK DRIVE*
2310 OPEN15,8,15,"I"+DR$:GOSUB7000:IFFETHENRETURN:   OPEN COMMAND CHANNEL
2320 OPEN5,8,5,DR$+":"+FL$+",P,R":GOSUB7000:REM     OPEN FILE FOR READ
2330 RETURN
2340 :
2400 GOSUB200:AD=D:AD$=D$:REM                      *GET LOAD ADDRESS*
2410 GOSUB200:AD=AD+D*B:AD$=D$+AD$:RETURN
2420 :
2500 PRINT"ĥ":D=5:GOSUB6000:IFFY=0THEN2570:        *SKIP BASIC*
2510 IFFY-1THEN2500
2520 PRINTC$(0)C$(6):J=0:REM                        CHECK FOR 3 ZEROES
2530 GET#5,D$:IFD$THENJ=J+1:GOTO2530
```



A Wooden Computer?

Not from Commodore!

So why should the desk look like wood? A pleasant cream and charcoal trimmed desk looks so much better with Commodore systems. One look and you'll see. Interlink desks are right. By design.

The specifications only confirm the obvious:

- Cream and charcoal color beautifully matches the Commodore hardware and blends with your decor.
- An ideal 710 mm (28") keyboard height yet no bumping knees because a clever cutout recesses the computer into the desk-top.
- High pressure laminate on both sides of a solid core for lasting beauty and strength.
- Electrostatically applied baked enamel finish on welded steel legs—no cheap lacquer job here.
- T-molding and rounded corners make a handsome finish on a durable edge that won't chip.
- Knocked down for safe, inexpensive shipment.
- Patented slip joints for quick easy assembly.
- Leveling glides for uneven floors.
- Room enough for a Commodore printer on the desk, yet fits into nearly any den or office niche— H: 660 mm (26") W: 1170 mm (46") D: 660 mm (26").
- Matching printer stand available with slot for bottom feeding.

Price: $299

In short, as Commodore dealers, we won't settle for anything that looks good only in the catalog! Our customers won't let us. They don't buy pictures. And neither should you. This is why **we will let you use one of our desks for a week** and then decide. If for any reason you don't like it, just return it in good condition for a cheerful refund.

If your Commodore dealer doesn't carry our desks yet, send a check for $299 and we will ship your desk freight paid!

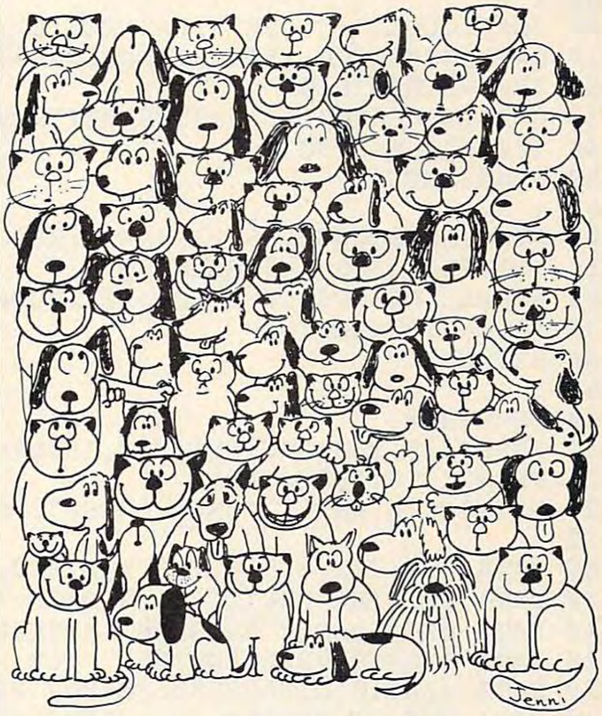
Name____________________

Address____________________

City__________ **St**_____ **Zip**______

Interlink, Inc., Box 134, Berrien Springs, MI 49103
Master Charge and Visa welcome. Call our order line:
616-473-3103

Every PET Needs a Friend.



CURSOR is the best friend your Commodore PET will ever have. Since July, 1978 we have published 150 of the most user-friendly programs for the PET available anywhere. When we write or edit a program, we spend lots of time fussing about how it will treat you. We pay attention to lots of little things that help make using a computer a pleasure instead of a pain.

Naturally, **CURSOR** programs are technically excellent. Each program that we purchase is extensively edited or re-written by a professional programmer. But *imagination* is just as important as being user-friendly and technically good! We delight in bringing you off-beat, unusual programs that "show off" the abilities of your PET or CBM.

CURSOR is user-friendly, technically great and full of imaginative programs. And every issue of **CURSOR** is still available! We continue to upgrade previously published programs so that they'll work on the three varieties of Commodore ROM's (Old, New, and 4.0). New issues also work on the 80 column CBM.

For only $4.95 you can buy a sample issue and judge for yourself. Or send $18 for a four-issue subscription. Each **CURSOR** comes to you as a C-30 cassette with five programs and a graphic Front Cover, ready to LOAD and RUN on your PET.

Who knows? After your PET meets **CURSOR**, things may never be the same!

AUTHORIZED DISTRIBUTORS:

Great Britain
AUDIOGENIC, Ltd.
P.O. Box 88
Reading, Berkshire

Holland
COPYTRONICS
Bergemeester
Van Suchtelenstraat 46
7413 XP Deventer

Japan
SYSTEMS FORMULATE CORP.
Shin-Makicho Bldg. 1-8-17
Yaesu, Chuo-ku, Tokyo 103

Australasia
MICROCOMPUTER HOUSE, LTD.
133 Regent Street
Chippendale, Sydney
N.S.W. Australia 2008

THE CODE WORKS
Box 550
Goleta, CA 93116
805-683-1585

```
254Ø GET#5,D$:IFD$THENJ=J+2:GOTO253Ø
255Ø GET#5,D$:IFD$THENJ=J+3:GOTO253Ø
256Ø AD=AD+J+3:GOSUB3ØØ:REM                        FIX ADDRESS
257Ø PRINT"ĥ"C$(Ø)C$(7)"↓↓":RETURN
258Ø :
26ØØ FG=Ø:CLOSE5:CLOSE15:PRINTC$(Ø)INT(TI/6)/1Ø"SEC
261Ø D=8:GOSUB6ØØØ:IFFY=ØTHENCLOSE4:END:REM       *DO ANOTHER FILE?*
262Ø IFFY-1THEN261Ø
263Ø D=9:GOSUB6ØØØ:IFFY=ØTHENRETURN:REM            CHANGE PRINTER?
264Ø IFFY-1THEN263Ø
265Ø FP=Ø:RETURN
266Ø :
6ØØØ FY=2:PRINTC$(Ø)C$(D)"←←←";:REM               *INPUT ROUTINE*
6Ø1Ø INPUTD$:IFD$="-"THEN6ØØØ
6Ø2Ø A%=VAL(D$):C$=LEFT$(D$,1):IFC$="N"THENFY=Ø
6Ø3Ø IFC$="Y"THENFY=1
6Ø4Ø RETURN
6Ø6Ø :
7ØØØ FE=Ø:REM                                     *CHECK FOR DISK ERROR*
7Ø1Ø INPUT#15,EN$,EM$,ET$,ES$:IFEN$="ØØ"THENRETURN
7Ø2Ø PRINTC$(Ø)"r"C$(13)
7Ø3Ø PRINTC$(Ø)EN$", "EM$", "ES$","ET$:FE=1:RETURN
7Ø4Ø :
8ØØØ DATA Ø,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,"↓↓↓→→→→→→→→→→"
85ØØ DATA PRINTER DEVICE #  3,DATE (MO/DA/YR)  -,DRIVE #  Ø,FILENAME  -
851Ø DATA SKIP BASIC PROGRAM  N,SKIPPING BASIC ....,DISASSEMBLING ....
852Ø DATA DISASSEMBLE ANOTHER FILE  N,DIFFERENT PRINTER  N,DISK DISASSEMBLER
853Ø DATA -BASIC DISASSEMBLER FOR PET DISK FILES-
854Ø DATA OUTPUT RESEMBLES ASSEMBLER SOURCE CODE.,DISK ERROR
9ØØØ DATA BRK,1,,ORA,2,3,,,,,,,,,,ORA,2,2,ASL,2,2,,,
9ØØ5 DATA PHP,1,,ORA,2,1,ASL A,1,,,,,,,,ORA,3,1,ASL,3,1,,,
9Ø1Ø DATA BPL,2,7,ORA,2,4,,,,,,,,,,ORA,2,5,ASL,2,5,,,
9Ø15 DATA CLC,1,,ORA,3,3,,,,,,,,,,ORA,3,2,ASL,3,2,,,
9Ø2Ø DATA JSR,3,1,AND,2,3,,,,,,,BIT,2,2,AND,2,2,ROL,2,2,,,
9Ø25 DATA PLP,1,,AND,2,1,ROL A,1,,,,,BIT,3,1,AND,3,1,ROL,3,1,,,
9Ø3Ø DATA BMI,2,7,AND,2,4,,,,,,,,,,AND,2,5,ROL,2,5,,,
9Ø35 DATA SEC,1,,AND,3,3,,,,,,,,,,AND,3,2,ROL,3,2,,,
9Ø4Ø DATA RTI,1,,EOR,2,3,,,,,,,,,,EOR,2,2,LSR,2,2,,,
9Ø45 DATA PHA,1,,EOR,2,1,LSR A,1,,,,,JMP,3,1,EOR,3,1,LSR,3,1,,,
9Ø5Ø DATA BVC,2,7,EOR,2,4,,,,,,,,,,EOR,2,5,LSR,2,5,,,
9Ø55 DATA CLI,1,,EOR,3,3,,,,,,,,,,EOR,3,2,LSR,3,2,,,
9Ø6Ø DATA RTS,1,,ADC,2,3,,,,,,,,,,ADC,2,2,ROR,2,2,,,
9Ø65 DATA PLA,1,,ADC,2,1,ROR A,1,,,,,JMP,3,4,ADC,3,1,ROR,3,1,,,
9Ø7Ø DATA BVS,2,7,ADC,2,4,,,,,,,,,,ADC,2,5,ROR,2,5,,,
9Ø75 DATA SEI,1,,ADC,3,3,,,,,,,,,,ADC,3,2,ROR,3,2,,,
9Ø8Ø DATA ,,,STA,2,3,,,,,,,STY,2,2,STA,2,2,STX,2,2,,,
9Ø85 DATA DEY,1,,,,,TXA,1,,,,,STY,3,1,STA,3,1,STX,3,1,,,
9Ø9Ø DATA BCC,2,7,STA,2,4,,,,,,,STY,2,5,STA,2,5,STX,2,6,,,
9Ø95 DATA TYA,1,,STA,3,3,TXS,1,,,,,,,,STA,3,2,,4,,,,
91ØØ DATA LDY,2,1,LDA,2,3,LDX,2,1,,,,LDY,2,2,LDA,2,2,LDX,2,2,,,
91Ø5 DATA TAY,1,,LDA,2,1,TAX,1,,,,,LDY,3,1,LDA,3,1,LDX,3,1,,,
911Ø DATA BCS,2,7,LDA,2,4,,,,,,,LDY,2,5,LDA,2,5,LDX,2,6,,,
9115 DATA CLV,1,,LDA,3,3,TSX,1,,,,,LDY,3,2,LDA,3,2,LDX,3,3,,,
912Ø DATA CPY,2,1,CMP,2,3,,,,,,,CPY,2,2,CMP,2,2,DEC,2,2,,,
9125 DATA INY,1,,CMP,2,1,DEX,1,,,,,CPY,3,1,CMP,3,1,DEC,3,1,,,
913Ø DATA BNE,2,7,CMP,2,4,,,,,,,,,,CMP,2,5,DEC,2,5,,,
9135 DATA CLD,1,,CMP,3,3,,,,,,,,,,CMP,3,2,DEC,3,2,,,
914Ø DATA CPX,2,1,SBC,2,3,,,,,,,CPX,2,2,SBC,2,2,INC,2,2,,,
9145 DATA INX,1,,SBC,2,1,NOP,1,,,,,CPX,3,1,SBC,3,1,INC,3,1,,,
915Ø DATA BEQ,2,7,SBC,2,4,,,,,,,,,,SBC,2,5,INC,2,5,,,
```

```
 9155 DATA SED,1,,SBC,3,3,,,,,,,,,,SBC,3,2,INC,3,2,,,
 10000 *********************
 10010 *                   *
 10020 * DISK DISASSEMBLER *
 10030 *                   *
 10040 * G.H.WATSON   3/81 *
 10050 *                   *
 10060 *********************
 11000 ----- VARIABLE TABLE ------
 11010 J,D     DUMMY INDEX/VARIABLE
 11020 A%      DUMMY INTEGER
 11030 D$,C$   DUMMY STRINGS
 11040 FG      END-OF-FILE FLAG
 11050 B1      OPCODE (DEC)
 11060 AD,AD$  ADDRESS (DEC,HEX)
 11070 M$      INSTRUCTION
 11080 B1$,B2$ 1ST & 2ND BYTES OF CODE
 11090 DR$,FL$ DRIVE #,FILENAME
 11100 DA$     DATE
 11110 D$()    HEX NUMERALS
 11120 H$()    HEX FOR BYTES
 11130 M$()    6502 MNEMONICS
 11140 B%()    # BYTES IN INSTRUCTION
 11150 A%()    ADDRESSING MODE
 11160 C$()    PRINT STRINGS
 11170 EN,EN$  ERROR #
 11180 EM$     ERROR MESSAGE
 11190 ET$,ES$ ERROR TRACK/SECTOR
 11200 FE      DISK ERROR FLAG
 11210 FP,FD   PRINTER/DATE FLAG
 11220 FY      FLAG FOR YES/NO
 11230 DV      PRINTER DEVICE #
 11240 B       CONSTANT = 256
READY.
```

Program 2.

```
5000 69 00 6D 00 00 65 00 61
5008 00 71 00 75 00 7D 00 00
5010 79 00 00 29 00 2D 00 00
5018 25 00 21 00 31 00 35 00
5020 3D 00 00 39 00 00 0E 00
5028 00 06 00 0A 16 00 1E 00
5030 00 90 00 B0 01 F0 7F 2C
5038 00 00 24 00 30 80 D0 FE
5040 10 FF 00 50 00 70 00 18
5048 D8 58 B8 C9 00 CD 00 00
5050 C5 00 C1 00 D1 00 D5 00
5058 DD 00 00 D9 00 00 E0 00
5060 EC 00 00 E4 00 C0 00 CC
5068 00 00 C4 00 CE 00 00 C6
5070 00 D6 00 DE 00 00 CA 88
5078 49 00 4D 00 00 45 00 41
5080 00 51 00 55 00 5D 00 00
5088 59 00 00 EE 00 00 E6 00
5090 F6 00 FE 00 00 E8 C8 4C
5098 00 00 6C 00 00 20 00 00
50A0 A9 00 AD 00 00 A5 00 A1
50A8 00 B1 00 B5 00 BD 00 00
50B0 B9 00 00 A2 00 AE 00 00
50B8 A6 00 BE 00 00 B6 00 A0
50C0 00 AC 00 00 A4 00 B4 00
50C8 BC 00 00 4E 00 00 46 00
50D0 4A 56 00 5E 00 00 EA 09
50D8 00 0D 00 00 05 00 01 00
50E0 11 00 15 00 1D 00 00 19
50E8 00 00 48 08 68 28 2E 00
50F0 00 26 00 2A 36 00 3E 00
50F8 00 6E 00 00 66 00 6A 76
5100 00 7E 00 00 40 60 E9 00
5108 ED 00 00 E5 00 E1 00 F1
5110 00 F5 00 FD 00 00 F9 00
5118 00 38 F8 78 8D 00 00 85
5120 00 81 00 91 00 95 00 9D
5128 00 00 99 00 00 8E 00 00
5130 86 00 96 00 8C 00 00 84
5138 00 94 00 AA A8 BA 8A 9A
5140 98 AA AA AA AA AA AA AA

S "DIS TEST",08,5000,5141
```

©

Line Input For The PET

Robert Lando
Toronto, Canada

Unfortunately, as many users will agree, the INPUT command on the Commodore Pet contains several undesirable features. First of all, if the RETURN key is pressed before any data is entered, the program will abruptly end and the user will be left with a READY. message. Although there are several "tricks" that the programmer can use to prevent this from happening, they do not alleviate another major problem.

No matter how an INPUT is programmed, when it is encountered the computer waits for a key to be typed, echoes it back to the screen, and waits for another until the RETURN key is pressed. The problem is that if the user enters a cursor movement key, its function will be echoed back to the screen. The user could, for example, clear the screen, and have no way of recovering the lost information.

Some computers offer a command that will accept one line of input from the user. The only acceptable keys are un-shifted letters from A-Z, digits from 0-9, the space bar, the delete key, and the return key. All other keys are ignored. Pressing RETURN, DELete, or SPACE before something else is typed will have no effect, and trailing spaces are ignored. This command, usually called INLINE (INput LINE), or LINPUT (Line INPUT), can be used in place of INPUT.

Below is a program written in machine language, that when called with a SYS command to its starting location, will do a LINPUT on the Commodore Pet with "upgrade" ROMs. Whatever the user types will be returned in the basic variable IN$. The program is completely relocatable, and occupies 305 bytes. The program is presented in assembler, and as a Basic loader. You may locate the program anywhere in memory, or have the loader program POKE it into the end of memory, and adjust the necessary pointers to protect it from being erased by string storage.

Program 1.

```
                       .OS
                       .BA $027A
              LEN      .DE $B1       ;LENGTH OF STRING
              BUL      .DE $B3       ;POINTER TO BUFFER
              CUR      .DE 167       ;CURSOR STATUS
              MAX      .DE 37        ;MAX # OF CHAR
              BOS      .DE $30       ;BOTTOM OF STRINGS PTR
              GCOLL    .DE $D400     ;GARBAGE COLLECT
              CHRREC   .DE $B5       ;LAST KEY RECEIVED
              TESTDIR  .DE 53888     ;TEST DIRECT MODE
              VAR      .DE $2A       ;START OF VARIABLES PTR
              SEARCH   .DE $B7       ;START OF SEARCH PTR
              ENDV     .DE $2C       ;END OF VARIABLES PTR
              ENDS     .DE $B9       ;END OF SEARCH PTR
              SLEN     .DE $BB       ;FINAL STRING LENGTH
              SNAME    .DE $42       ;VARIABLE NAME
              ADDSTR   .DE $D001     ;ADD A VARIABLE
027A- 20 80 D2         JSR TESTDIR   ;EXIT IF DIRECT MODE
027D- A9 02            LDA #$2       ;BUFFER POINTER = $0200
027F- 85 B4            STA *BUL+1
0281- A9 00            LDA #00
0283- 85 B3            STA *BUL
0285- A9 8D            LDA #141
0287- 20 D2 FF         JSR $FFD2     ;DO CRLF
028A- A9 3E            LDA #62
028C- 20 D2 FF         JSR $FFD2     ;PRINT PROMPT
028F- A9 00            LDA #0
0291- 85 B1            STA *LEN      ;SET LENGTH TO ZERO
```

80 COLUMN GRAPHICS

The Integrated Visible Memory for the PET has now been redesigned for the new 12" screen 80 column and forthcoming 40 column PET computers from Commodore. Like earlier MTU units, the new K-1008-43 package mounts inside the PET case for total protection. To make the power and flexibility of the 320 by 200 bit mapped pixel graphics display easily accessible, we have designed the Keyword Graphic Program. This adds 45 graphics commands to Commodore BASIC. If you have been waiting for easy to use, high resolution graphics for your PET, isn't it time you called MTU?

The image on the screen was created by the program below.

```
 10 VISMEM: CLEAR
 20 P=160: Q=100
 30 XP=144: XR=1.5*3.1415927
 40 YP=56: YR=1: ZP=64
 50 XF=XR/XP: YF=YP/YR: ZF=XR/ZP
 60 FOR ZI=-Q TO Q-1
 70 IF ZI<-ZP OR ZI>ZP GOTO 150
 80 ZT=ZI*XP/ZP: ZZ=ZI
 90 XL=INT(.5+SQR(XP*XP-ZT*ZT))
100 FOR XI=-XL TO XL
110 XT=SQR(XI*XI+ZT*ZT)*XF: XX=XI
120 YY=(SIN(XT)+.4*SIN(3*XT))*YF
130 GOSUB 170
140 NEXT XI
150 NEXT ZI
160 STOP
170 X1=XX+ZZ+P
180 Y1=YY-ZZ+Q
190 GMODE 1: MOVE X1,Y1: WRPIX
200 IF Y1=0 GOTO 220
210 GMODE 2: LINE X1,Y1-1,X1,0
220 RETURN
```

K-1008-43M Manual only $10 (credited toward purchase)
k-1008-43 Complete ready to install package $495

Mastercharge and Visa accepted

Write or call today for our full line catalog describing all MTU 6502 products, including our high speed 8" Floppy Disk Controller for up to 4 megabytes of PET storage.

MTU
Micro Technology Unlimited
2806 Hillsborough Street
P.O. Box 12106
Raleigh, NC 27605, U.S.A.
(919)833-1458

NOW 80 COLUMN PETS CAN HAVE MTU HIGH RESOLUTION GRAPHICS

```
0293- 85 A7       0290          STA *CUR      ;TURN ON CURSOR
0295- 20 E4 FF    0300 GET      JSR $FFE4
0298- F0 FB       0310          BEQ GET       ;WAIT FOR A KEY
029A- 85 B5       0320          STA *CHRREC   ;SAVE KEY PRESSED
029C- C9 0D       0330          CMP #13       ;RETURN KEY?
029E- D0 19       0340          BNE SKIP1     ;BRANCH IF NOT
02A0- A4 B1       0350          LDY *LEN      ;IS LENGTH ZERO?
02A2- F0 15       0360          BEQ SKIP1     ;BRANCH IF YES
02A4- A9 01       0370          LDA ##$01
02A6- 85 A7       0380          STA *CUR      ;TURN OFF CURSOR
02A8- A9 92       0390          LDA #146
02AA- 20 D2 FF    0400          JSR $FFD2     ;PRINT RVS/OFF
02AD- A9 20       0410          LDA #32
02AF- 20 D2 FF    0420          JSR $FFD2     ;PRINT SPACE
02B2- A9 8D       0430          LDA #141
02B4- 20 D2 FF    0440          JSR $FFD2     ;DO CRLF
02B7- D0 4B       0450          BNE ADDVAR    ;BRANCH ALWAYS
02B9- A5 B1       0460 SKIP1    LDA *LEN      ;CHECK STRING LENGTH
02BB- C9 25       0470          CMP #MAX      ;MAXIMUM LENGTH?
02BD- D0 06       0480          BNE SKIP2     ;NO
02BF- A5 B5       0490          LDA *CHRREC   ;YES, CHECK CHAR
02C1- C9 14       0500          CMP #20       ;DELETE?
02C3- D0 D0       0510          BNE GET       ;NO, NOT DELETE
02C5- 18          0520 SKIP2    CLC
02C6- A5 B5       0530          LDA *CHRREC
02C8- C9 41       0540          CMP #65       ;IS CHAR 65 OR MORE
02CA- F0 04       0550          BEQ OK1       ;YES
02CC- B0 02       0560          BCS OK1
02CE- D0 12       0570          BNE SKIP3     ;IT WAS LESS THAN 65

02D0- 18          0580 OK1      CLC
02D1- C9 5B       0590          CMP #91       ;IS IT LESS THAN 91?
02D3- B0 0D       0600          BCS SKIP3     ;NO
02D5- A4 B1       0610 OK2      LDY *LEN
02D7- A5 B5       0620          LDA *CHRREC
02D9- 91 B3       0630          STA (BUL),Y   ;STORE CHAR IN BUFFER
02DB- E6 B1       0640          INC *LEN      ;INCREMENT LENGTH
02DD- 20 D2 FF    0650 DEL      JSR $FFD2     ;PRINT CHARACTER
02E0- D0 B3       0660          BNE GET       ;BRANCH ALWAYS
02E2- A5 B1       0670 SKIP3    LDA *LEN      ;CHECK THE LENGTH
02E4- F0 06       0680          BEQ SKIP4     ;BRANCH IF LENGTH 0
02E6- A5 B5       0690          LDA *CHRREC   ;IS IT A SPACE?
02E8- C9 20       0700          CMP #32
02EA- F0 E9       0710          BEQ OK2       ;YES, THE SPACE IS OK
02EC- A5 B5       0720 SKIP4    LDA *CHRREC   ;IS IT A DIGIT?
02EE- C9 30       0730          CMP #48
02F0- 90 04       0740          BCC SKIP5
02F2- C9 3A       0750          CMP #58
02F4- 90 DF       0760          BCC OK2       ;YES, IT'S A DIGIT
02F6- C9 14       0770 SKIP5    CMP #20       ;IS IT A DELETE?
02F8- D0 9B       0780          BNE GET       ;BRANCH IF IT ISN'T
02FA- A5 B1       0790          LDA *LEN      ;CHECK THE LENGTH
02FC- F0 97       0800          BEQ GET       ;BRANCH IF IT'S ZERO
02FE- C6 B1       0810          DEC *LEN      ;DECREMENT LENGTH
0300- A9 14       0820          LDA #20       ;LOAD A WITH DELETE
0302- D0 D9       0830          BNE DEL       ;BRANCH ALWAYS
0304- 20 00 D4    0840 ADDVAR   JSR GCOLL     ;GARBAGE COLLECT
```

```
0307-  C6 B1     0850           DEC *LEN
0309-  A4 B1     0860 CHECKSP   LDY *LEN
030B-  B1 B3     0870           LDA (BUL),Y
030D-  C9 20     0880           CMP #$20       ;PEEL OFF TRAILING SPACES
030F-  D0 04     0890           BNE MOVEIT
0311-  C6 B1     0900           DEC *LEN       ;DECREMENT LENGTH COUNTER
0313-  D0 F4     0910           BNE CHECKSP    ;BRANCH ALWAYS
0315-  A4 B1     0920 MOVEIT    LDY *LEN
0317-  C8        0930           INY
0318-  84 BB     0940           STY *SLEN      ;SAVE CURRENT LENGTH
031A-  C6 30     0950 MOVE1     DEC *BOS       ;DEC STRING PTR
031C-  A5 30     0960           LDA *BOS
031E-  C9 FF     0970           CMP #255
0320-  D0 02     0980           BNE POS
0322-  C6 31     0990           DEC *BOS+1
0324-  A4 B1     1000 POS       LDY *LEN
0326-  B1 B3     1010           LDA (BUL),Y
0328-  A0 00     1020           LDY #$00
032A-  91 30     1030           STA (BOS),Y    ;MOVE FROM INPUT
032C-  C6 B1     1040           DEC *LEN       ; BUFFER TO BOTTOM OF
032E-  A5 B1     1050           LDA *LEN       ; STRINGS
0330-  C9 FF     1060           CMP #255
0332-  D0 E6     1070           BNE MOVE1
0334-  A9 01     1080           LDA #$01
0336-  85 A7     1090           STA *CUR       ;MAKE SURE CURSOR OFF
0338-  A5 2A     1100 LINKTAB   LDA *VAR       ;SET SEARCH PTR
033A-  85 B7     1110           STA *SEARCH    ;TO LOCATIONS POINTED
033C-  A5 2B     1120           LDA *VAR+1     ;TO BY START OF
```

```
033E- 85 B8              STA *SEARCH+1        ;VARIABLES POINTER
0340- A5 2C              LDA *ENDV   ;SET END-OF-SEARCH
0342- 85 B9              STA *ENDS   ;POINTER TO LOCATIONS
0344- A5 2D              LDA *ENDV+1 ;POINTED TO BY END-OF-
0346- 85 BA              STA *ENDS+1 ;VARIABLES POINTER
0348- A0 00     HUNT     LDY #$00    ;HUNT FOR IN$ IN TABLE
034A- B1 B7              LDA (SEARCH),Y
034C- C9 49              CMP #$49    ;FOUND I?
034E- D0 18              BNE INCSEVEN         ;NO
0350- C8                 INY
0351- B1 B7              LDA (SEARCH),Y
0353- C9 CE              CMP #$CE    ;FOUND N?
0355- D0 11              BNE INCSEVEN         ;NO
0357- A5 BB              LDA *SLEN
0359- C8                 INY
035A- 91 B7              STA (SEARCH),Y       ;SET LENGTH OF IN$
035C- C8                 INY
035D- A5 30              LDA *BOS
035F- 91 B7              STA (SEARCH),Y       ;POINTER TO STRING
0361- C8                 INY
0362- A5 31              LDA *BOS+1
0364- 91 B7              STA (SEARCH),Y
0366- D0 28              BNE RET     ;BRANCH ALWAYS
0368- A2 07     INCSEVEN LDX #$07    ;INCREMENT SEARCH
036A- E6 B7     NBYTE    INC *SEARCH ; POINTER BY 7
036C- D0 02              BNE SAMEPAGE
036E- E6 B8              INC *SEARCH+1
0370- CA        SAMEPAGE DEX
0371- D0 F7              BNE NBYTE
0373- A5 B8     DONE?    LDA *SEARCH+1        ;CHECK FOR END
0375- C5 BA              CMP *ENDS+1 ; OF VARIABLE TABLE
0377- 90 CF              BCC HUNT
0379- D0 06              BNE ADDIN
037B- A5 B7              LDA *SEARCH
037D- C5 B9              CMP *ENDS
037F- 90 C7              BCC HUNT    ;MORE TO GO
0381- A9 49     ADDIN    LDA #$49    ;ADD IN$ TO TABLE
0383- 85 42              STA *SNAME  ;SET UP NAME OF STRING
0385- A9 CE              LDA #$CE
0387- 85 43              STA *SNAME+1
0389- 20 01 D0           JSR ADDSTR
038C- F0 AA              BEQ LINKTAB ;NOW GO LOOK FOR IT
038E- D0 A8              BNE LINKTAB
0390- A5 2C     RET      LDA *ENDV   ;POINTERS OK?
0392- C5 2A              CMP *VAR
0394- D0 15              BNE BASIC   ;YES
0396- A5 2D              LDA *ENDV+1
0398- C5 2B              CMP *VAR+1
039A- D0 0F              BNE BASIC
039C- A2 07              LDX #$07
039E- E6 2C     ADJUST   INC *ENDV   ;NO, FIX POINTERS
03A0- E6 2E              INC *ENDV+2
03A2- D0 04              BNE SAPAGE
03A4- E6 2D              INC *ENDV+1
03A6- E6 2F              INC *ENDV+3
03A8- CA        SAPAGE   DEX
```

```
03A9- D0 F3     1690           BNE ADJUST
03AB- 60        1700 BASIC     RTS  ;RETURN TO BASIC
                1710           .EN
```

Program 2.

```
100 REM   *** RELOCATABLE LINE INPUT
110 REM   *** FOR UPGRADE ROM PETS
120 REM
130 REM   *** BY ROBERT LANDO
140 REM       146 VAN HORNE AVENUE
150 REM       TORONTO, CANADA
160 REM
170 REM
180 PRINT"▯▮LINE INPUT■                ROBERT LANDO"
190 PRINT"▮▮PLEASE ENTER THE DECIMAL LOCATION THAT"
200 PRINT"▮YOU WOULD LIKE THE LINE INPUT PROGRAM"
210 PRINT"▮TO START AT. IF YOU ENTER AN ASTERISK,"
220 PRINT"▮THE PROGRAM WILL BE PACKED INTO THE"
230 PRINT"▮END OF AVAILABLE MEMORY AND THE"
240 PRINT"▮NECESSARY POINTERS WILL BE ADJUSTED TO"
250 PRINT"▮PROTECT IT FROM BEING OVERWRITTEN BY"
260 PRINT"▮STRINGS.▮▮"
270 INPUT"   *■■■";S$
280 IFS$="*"THEN310
290 S=VAL(S$):IFS<512ORS>36559THENPRINT"▯";:GOTO270
300 E=S+305:GOTO330
310 M=PEEK(52)+PEEK(53)*256:E=M:S=M-305
320 M=S:POKE53,M/256:POKE52,M-256*PEEK(53)
330 FORX=STOE:READV:POKEX,V:NEXTX
340 PRINT"▯THE LINE INPUT PROGRAM IS NOW IN"
350 PRINT"▮MEMORY. WHEN YOU USE THE COMMAND"
360 PRINT"▮SYS (";S;") IN A PROGRAM, WHATEVER"
370 PRINT"▮THE USER ENTERS WILL BE TRANSFERRED"
380 PRINT"▮TO THE BASIC VARIABLE IN$"
390 END
400 DATA32,128,210,169,2,133,180,169,0,133,179,169,141,32,210,255,169
410 DATA62,32,210,255,169,0,133,177,133,167,32,228,255,240,251,133,181
420 DATA201,13,208,25,164,177,240,21,169,1,133,167,169,146,32,210,255,169
430 DATA32,32,210,255,169,141,32,210,255,208,75,165,177,201,37,208,6
440 DATA165,181,201,20,208,208,24,165,181,201,65,240,4,176,2,208,18,24
450 DATA201,91,176,13,164,177,165,181,145,179,230,177,32,210,255,208
460 DATA179,165,177,240,6,165,181,201,32,240,233,165,181,201,48,144,4,201
470 DATA58,144,223,201,20,208,155,165,177,240,151,198,177,169,20,208
480 DATA217,32,0,212,198,177,164,177,177,179,201,32,208,4,198,177,208
490 DATA244,164,177,200,132,187,198,48,165,48,201,255,208,2,198,49,164
500 DATA177,177,179,160,0,145,48,198,177,165,177,201,255,208,230,169,1
510 DATA133,167,165,42,133,183,165,43,133,184,165,44,133,185,165,45,133
520 DATA186,160,0,177,183,201,73,208,24,200,177,183,201,206,208,17,165
530 DATA187,200,145,183,200,165,48,145,183,200,165,49,145,183,208,40,162
540 DATA7,230,183,208,2,230,184,202,208,247,165,184,197,186,144,207,208
550 DATA6,165,183,197,185,144,199,169,73,133,66,169,206,133,67,32,1
560 DATA208,240,170,208,168,165,44,197,42,208,21,165,45,197,43,208,15
570 DATA162,7,230,44,230,46,208,4,230,45,230,47,202,208,243,96
READY.
```

©

Measure Time Intervals With The Pet Parallel User Port

Robert Macnaughton
Rexdale, Canada

This article describes a machine language program that can be used to measure seven successive small time intervals, using the CBM Parallel User Port (PUP), and eight phototransistors, to the nearest 1/10000s.

Since no page zero locations are used, this program should run on any PET (except 4.0, since it would need to be moved above 864 decimal for 4.0 BASIC).

The PUP, located at the back of the CBM, consists of 24 contacts to the main logic board, labelled as follows:

1	2	3	4	5	6	7	8	9	10	11	12
A	B	C	D	E	F	H	J	K	L	M	N

Only the bottom row of contacts will be used. The top row of contacts are for use by CBM diagnostic routines during servicing.

On the bottom row of contacts, Pin M is the CB2 line, used in many programs for sound effects; contacts A and N are grounds, and contact B is the CA1 line.

We will use contacts C,D,E,F,H,J,K and L, known as PA0, PA1, PA2, PA3, PA4, PA5, PA6 and PA7, the programmable input/output lines, to receive information from eight phototransistors, the detectors of the position of some moving object.

The eight lines are treated by the PET as a single memory location, 59471 in decimal or $E84F in hexadecimal. It is known as the ORA, the output register for I/O Port A, without handshaking. At any time, a PEEK(59471) will indicate the condition of the ORA.

The DDR A, the data direction register for Port A, is used to designate which are the input and which are the output lines of the ORA. Its address is 59459 or $E843. A zero in bit three would make PA3 an input line and a one would make it an output line. If you POKE 59459,76 then PA2, PA3 and PA6 will be output lines and the rest input lines, since 76 in binary is 01001100.

In this timer, all eight lines are made inputs by POKE 59459,0. A PEEK(59459) when the CBM is first turned on will show that all the lines are initially inputs.

When running, the timer program looks at the contents of the ORA again and again. To understand the result, the contents of 59471 must be expressed as a binary number. Each of the eight I/O lines corresponds to one bit in this number. Any line grounded will be represented as a 0. If not grounded, it will be represented as a 1. More exactly, if a resistance of less than about 2000Ω is connected from a PA line to GND, the state of the line will be interpreted as a 0. If the resistance is more than 2000Ω, it will be interpreted as a 1.

If you PEEK(59471) with nothing connected to the PUP, you will get 255. If you short out all eight lines, you will get a 0. (First make sure that they are all input lines.)

	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
bit	7	6	5	4	3	2	1	0
value	128	64	32	16	8	4	2	1
59471								
255	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	0
4	0	0	0	0	0	1	0	0
8	0	0	0	0	1	0	0	0
16	0	0	0	1	0	0	0	0
32	0	0	1	0	0	0	0	0
64	0	1	0	0	0	0	0	0
128	1	0	0	0	0	0	0	0
214	1	1	0	1	0	1	1	0

The collectors of eight FPT100 phototransistor's are connected to the eight PA lines, and their emitters to ground at contact N. When enough light strikes a phototransistor such as the FPT100, its resistance falls to about 200Ω. This is interpreted as a 0 in the ORA. When the light is cut off, the resistance increases dramatically and is interpreted as a 1. As an object passes by a phototransistor, the state of that PA line will change from 0 to 1 and back to 0 as the light is temporarily interrupted.

I have placed the phototransistors in holes drilled in a meter stick 15 cm apart. The position of the first phototransistor must be adjustable to start the timer at the correct moment. Opposite each phototransistor is a small flashlight bulb attached to a second meter stick. The two meter sticks are placed on either side of a ramp. A large

PETTERM

INTELLIGENT TERMINAL

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

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

forth

for PET/CBM

FORTH is a new concept in programming, with the speed of compilers and interactive ease of BASIC. Programs become a part of FORTH extending the power of FORTH and your PET.

8050,4040 disk, cassette all PET-CBMs 16k+

Starter	fig-FORTH w editor assembler	$35
Personal	floating point; strings; source	$75
Professional	turnkey development/data base	$259

F S S

software for small computers

1903 Rio Grande
Austin, Texas
78705

1-512-477-2207

P.O. Box 8403
Austin, Texas
78712

master charge

DEALERS INQUIRE

VISA

CBM/PET INTERFACES

RS-232 SERIAL PRINTER INTERFACE – addressable – baud rates to 9600 – switch selectable upper/lower, lower/upper case – works with WORDPRO, BASIC and other software – includes case and power supply.
MODEL – ADA1450 149.00

CENTRONICS/NEC PARALLEL INTERFACE - addressable – high speed – switch selectable upper/lower, lower/upper case – works with WORDPRO, BASIC and other software – has Centronics 36 pin ribbon connector at end of cable.
MODEL – ADA1600 129.00

CENTRONICS 730/737 PARALLEL INTERFACE – as above but with Centronics card edge connector at end of cable.
MODEL – ADA730 129.00

COMMUNICATIONS INTERFACE WITH SERIAL AND PARALLEL PORTS – addressable – software driven – true ASCII conversion – selectable reversal of upper-lower case – baud rates to 9600 – half or full duplex – X-ON, X-OFF – selectable carriage return delay – 32 character buffer – centronics compatible — much more.
MODEL – SADI 295.00

ANALOG TO DIGITAL CONVERTER – 16 channels – 0 to 5.12 volt input voltage range – resolution is 20 millivolts per count – conversion time is less than 100 microseconds per channel.
MODEL – PETSET1 295.00

REMOTE CONTROLLER WITH CLOCK/CALENDAR – controls up to 256 devices using the BSR X10 remote control receivers – 8 digital inputs, TTL levels or switch closure — 8 digital outputs, TTL levels.
MODEL – PETSET2 295.00

All prices are in US dollars for 120VAC.
Prices on 220 VAC slightly higher.
Allow $5.00 shipping & handling, foreign orders add 10% for AIR postage.
Connecticut residents add 7½% sales tax.
All prices and specifications subject to change without notice.
Our 30 day money back trial period applies.
MASTER CHARGE/VISA accepted.

MENTION THIS MAGAZINE WITH YOUR ORDER AND DEDUCT 5% FROM TOTAL.

IN CANADA order from: Batteries Included, Ltd., 71 McCaul Street, F6 Toronto, Canada M5T2X1, (416)596-1405.
IN THE USA order from your local dealer or direct: Connecticut microComputer, Inc., 34 Del Mar Drive, Brookfield, CT 06804, (203)775-4595.

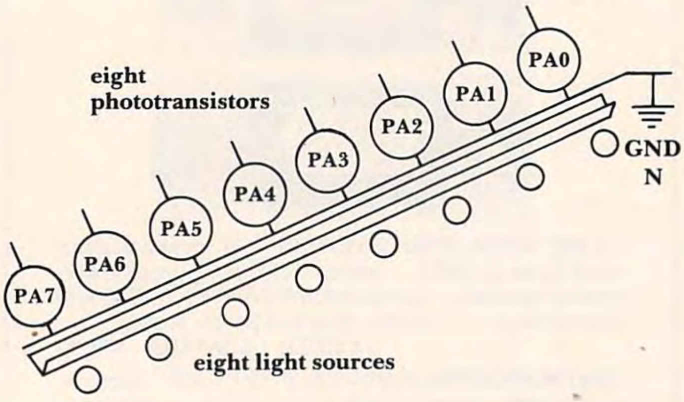
Dealer inquiries invited.

CmC

Connecticut microComputer, Inc.

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

ball bearing rolling down the ramp will be timed as it interrupts each light beam in turn.



If the times you wished to measure were long, you could write a BASIC program to measure these time intervals, using the internal "jiffy" clock of the PET. The light to each phototransistor would have to be cut off long enough that it would still be cut off when the program got around to checking the state of 59471.

To fully utilize the 1 megacycle clock in the CBM, a machine language program must be used.

The program begins by setting the interrupt flag. This will ensure that the timmig will not be interrupted by the CBM as it performs its normal interrupt every 1/60 s, to update its clock, flash the cursor if needed, etc.

It then goes into a loop to load all the various memory locations used to store the times, with zeros. At the same time it prints a ? at the top left of the screen. It then goes into a second loop to wait for PA? to become 1 when the ball is rolled into place at the top of the ramp. An R for READY now appears on the screen.

The following table shows how the ORA changes as the ball rolls down the ramp.

SCREEN	BINARY	DECIMAL	
?	00000000	0	ball not on ramp
R	00000001	1	ball in place at top of ramp
T	00000000	0	ball rolling
1	00000010	2	passes PA1
1	00000000	0	ball rolling
2	00000100	4	passes PA2
2	00000000	0	ball rolling
3	00001000	8	passes PA3
3	00000000	0	ball rolling
4	00010000	16	passes PA4
4	00000000	0	ball rolling
5	00100000	32	passes PA5
5	00000000	0	ball rolling
6	01000000	64	passes PA6
6	00000000	0	ball rolling
7	10000000	128	passes PA6

When 59471 becomes 0, the timer enters a timing loop. Each time through the loop it checks 59471 for a 0, then adds 1 to a counter. When 59471 has the next expected value, the contents of this counter are stored, and the timing resumes, continuing until all seven times have been measured. When the program returns to BASIC, the contents of the memory locations containing the count can be recalled and converted to seconds.

Since each timing loop takes 43 cycles of the CBM's internal 1 megacycle clock, each count represents 43 microseconds.

The count is contained in three locations. The first is incremented in each loop. The second is incremented only when the first passes 255 and becomes 0 again. The third is incremented only when the second passes 255 and becomes 0 again. The largest count possible is then (255x256x256) + (255x256) + 255 or 16777215. This is slightly more than 12 minutes.

I have included a second copy of the machine language program which shows the timing loop. Beside each step I have written the number of cycles of the PET's internal clock that are needed to complete each step. The total number of cycles is 43. Some extra time is used to store the count as each phototransistor is passed. If you wish, this could be calculated and added on to the total time as a correction.

I have also included a BASIC program to operate the clock in an organized fashion. It asks you how many runs you wish to make down the ramp, then stores the seven times for each run. Eventually, the average time for each part of the run is calculated. With a few minor changes, this program can be used in almost any situation where accurate timing is needed.

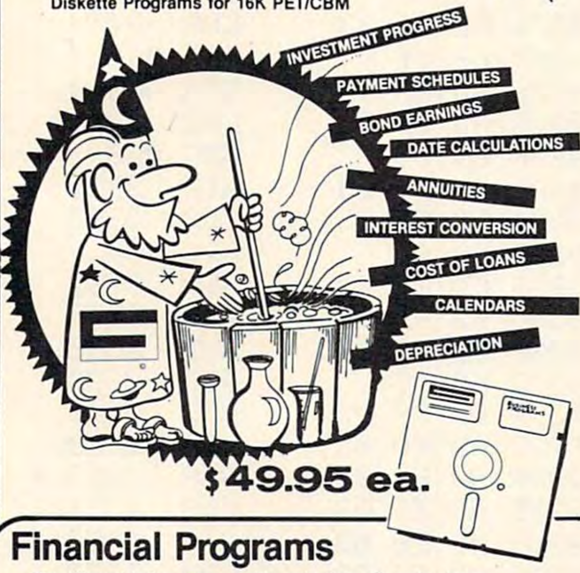
TIMER COMMENTS

1 Disable the interrupt flag
2 Load the accum with the code for ?
3 Store at top left corner of screen
4 Load the x-register with a 2
5 Store the 2 at 0336
6 Load the y-register with decimal 25
7 Load the accum with a 0
8 Store 0 at all locations from 03DF to 03DF + 25 by looping until y = 0
10 Compare y with zero
11 12 If y isn't zero then loop to step 7
12 Load accum with the contents of 59471
13 Check if PA0 is a 1 or a 0
14 If PA0 = 0 then loop and check again
15 Now PA0 is 1: R for Ready into accum
16 store R on the screen
17 This is a time delay while things
18 settle down. Load x and y with 255
19 and decrement them both to zero.
20 Each time x decrements from 255 to 0
21 y decrements by one. Finally both are
22 zero
23 Load accum with 59471 once more
24 Test to see if PA0 is still a 1
25 If so, loop back to 23 and try again
26 Now PA0 is a 0, the timing must start

Introducing The

FINANCIAL WIZARD

Diskette Programs for 16K PET/CBM



$49.95 ea.

Financial Programs

The Wizard helps you quickly calculate 12 major types of Financial Transactions with ease and accuracy.

Created for businesses and individuals, this DISK PACKAGE has 9 programs plus instructions. The Wizard delivers answers on the screen or printer.

CASCADE COMPUTERWARE • (206) 355-6121

Box 2354 • Everett, WA 98203 — Dealer inquiries invited

---

80 × 25

PET/CBM™

2000/3000/4000 Series

not using a CRT, or display controller chip

$275.00*

Select either **80 × 25** or **40 × 25** On The Built-in Display

From the keyboard or program

Displays the full, original character set

Available from your local dealer or:

EXECOM CORP.
1901 Polaris Ave.
Racine, WI 53404
Ph. 414-632-1004

*Plus installation charge of $75.00

Available only for Basic 3.0 & Basic 4.0

PET & CBM™ a trademark of Commodore Business Machines

Commodore 3.0/4.0/2.1 Dos Source Code!

Complete annotated source code for 3.0 ROMs, 4.0 ROMs and 2.1 DOS (4040-,8050-disk) now available. All entry points, routine/variable names, info on routine operation, register contents, etc. included. 4.0 source code comes with 3.0 correspondence codes so you can change 3.0 machine-language programs to 4.0 and vice versa. Hardcopy only. $129. each. All 3 (3.0, 4.0, 2.1 DOS) for $310.00. Source files on diskette on special order.

Software/Firmware

- EXTRAMON extended monitor in 2K EPROM for any slot, 3.0/4.0 ROMs, plus hardcopy documentation. $19.95
- DISK MONITOR© - reads T/S from disk to screen at 0.21 seconds/blk. Allows editing like the resident monitor, updates disk, traces links, etc. FAST! 31.1 Kbytes/sec. $49.95
- BULLETPROOF!© - full-screen input routine. Uses machine-language to get user input, places it in a user-defined variable. Text on screen outside fields impossible to disturb. Documented source code incl. with disk. $29.95

THE GREAT WESTERN SOFTWARE CO.

Authorized dealer for RPL© compilers

1624 N.W. 9th Ave., Ft. Lauderdale, FL 33311 (305) 523-1351

NEW PET/CBM SOFTWARE

Let Computer Mat turn your Pet into a Home Arcade!

BLAST-IT — Its your ship vs. a swarm of killer gammaroidz. You are on a collision course and must destroy them before they blast you into the next galaxy. Four levels of play. Has hyperspace keys that move you around. Arcade style entertainment at its finest. GREAT GRAPHICS AND SOUND.

Cass. 8K $9.95

MUNCHMAN — How many dots can you cover? It's you against the computer munchers ZIP and ZAP. Can you clear the maze first or will they get you? Number keys move you up, down, right and left. GREAT GRAPHICS AND SOUND.

Cass. 8K $9.95

LAZER-ATTACK — Its you against a barrage of enemy lazers that are aimed at your ammo dumps. Sight in on the targets and score as many hits as you dare. As your skill increases so does the the difficulty — (10 levels to select). This is an arcade-style game with great graphics and sound effects. A must for your PET/CBM.

Cass. 8K $9.95

ALL OUR SOFTWARE RUNS IN 8K
OLD-NEW ROM — 40 CHR. SCREEN
WRITE FOR FREE CATALOG OF VIC/PET SOFTWARE
PLEASE ADD $1.00 PER ORDER FOR SHIPPING

COMPUTER MAT • BOX 1664C • LAKE HAVASU CITY, AZ. 86403

27 Store a T on the screen
28 Begin timing loop by clearing the
29 carry flag, then load accum with 03E0
32 Add 1 to the contents of this
30 location and then store it back there
31 Add zero to the contents of 03E1
32 (and 1 if the carry flag was set by
33 the previous addition) and store it
34 Add zero(+ 1 if the carry flag is set)
35 to the contents of 03E2
36 03E0, 03E1, 03E2 contain the total time
37 Check 57471 to see if the next PA
38 line is a 1 or a 0 using 0336
39 0336 contains a 2: binary 00000010
40 If a 0, loop: if a 1, then arithmetic
41 shift left the value in 0336: see text
42 Store the three values representing
43 the elapsed time using the current
44 value of y (It is 0 to start with)
45
46
47
48 Clear the carry flag before addition
49 Transfer y, a counter, to the accum
50 Add 177 to it to make it the ASCII
51 code for y and store it on screen
52 Increment y (next time measurement)
53 Compare y with 7: If y is less than 7
54 then go back to start of timing cycle
55 If y is 7, the program is over: clear
56 interrupt flag and return to BASIC

Program 1.

```
 TIMER                     $033A SYS 826

 1 033A 78                 SEI
 2 033B A9 BF              LDA #BF
 3 033D 8D 00 80           STA  8000
 4 0340 A2 02              LDX #02
 5 0342 8E 36 03           STX  0336
 6 0345 A0 19              LDY #19
 7 0347 A9 00              LDA #00
 8 0349 99 DF 03           STA  03DF,Y
 9 034C 88                 DEY
10 034D C0 00              CPY #00
11 034F D0 F6              BNE  0347
12 0351 AD 4F E8           LDA  E84F
13 0354 29 01              AND #01
14 0356 F0 F9              BEQ  0351
15 0358 A9 92              LDA #92
16 035A 8D 00 80           STA  8000
17 035D A0 FF              LDY #FF
18 035F A2 FF              LDX #FF
19 0361 CA                 DEX
20 0362 D0 FD              BNE  0361
21 0364 88                 DEY
22 0365 D0 FA              BNE  0361
23 0367 AD 4F E8           LDA  E84F
24 036A 29 01              AND #01
25 036C D0 F9              BNE  0367
26 036E A9 94              LDA #94
27 0370 8D 00 80           STA  8000
28 0373 18                 CLC
29 0374 AD E0 03           LDA  03E0
30 0377 69 01              ADC #01
31 0379 8D E0 03           STA  03E0
32 037C AD E1 03           LDA  03E1
33 037F 69 00              ADC #00
34 0381 8D E1 03           STA  03E1
35 0384 AD E2 03           LDA  03E2
36 0387 69 00              ADC #00
37 0389 8D E2 03           STA  03E2
38 038C AD 4F E8           LDA  E84F
39 038F 2D 36 03           AND  0336
40 0392 F0 DF              BEQ  0373
41 0394 0E 36 03           ASL  0336
42 0397 AD E0 03           LDA  03E0
43 039A 99 E3 03           STA  03E3,Y
44 039D AD E1 03           LDA  03E1
45 03A0 99 EA 03           STA  03EA,Y
46 03A3 AD E2 03           LDA  03E2
47 03A6 99 F1 03           STA  03F1,Y
48 03A9 18                 CLC
49 03AA 98                 TYA
50 03AB 69 B1              ADC #B1
51 03AD 8D 00 80           STA  8000
52 03B0 C8                 INY
53 03B1 C0 07              CPY #07
54 03B3 D0 BE              BNE  0373
55 03B5 58                 CLI
56 03B6 60                 RTS
```

Program 2.

```
READY.

 10 REM TIMER BASIC
 20 REM ROBERT MACNAUGHTON OCT 5/80
 25 REM 2124 GREENHURST AVE
 30 REM MISSISSAUGA    L4X 1J6
 35 REM THE MACHINE LANGUAGE PROGRAM ¬
      ¬MEASURES 7 TIMES DURING A SINGLE ¬
      ¬TRIP
 40 REM UP TO 8 PHOTOTRANSISTORS ARE ¬
      ¬CONNECTED TO PA0-7
 45 REM SYS 826 ACTIVATES THE TIMER AND ¬
      ¬? APPEARS
 50 REM WHEN PA0 IS BLOCKED OFF, R ¬
      ¬APPEARS AND THE TIMER IS READY TO ¬
      ¬START
 60 REM WHEN LIGHT AGAIN FALLS ON PA0,
      ¬ THE TIMER STARTS AND T APPEARS
 70 REM AS EACH OF PA1-7 IS CUT OFF,
      ¬ THE TOTAL ELAPSED TIME IS STORED
 75 REM AS EACH MEASUREMENT IS MADE,
      ¬ ITS NUMBER APPEARS (1-7)
 80 REM UNUSED PA LINES SHOULD BE OPEN ¬
      ¬CIRCUITS
 200 PRINT"ħ"
```

```
205 INPUT"↓NUMBER OF RUNS";NR
210 FORJ=1TONR
215 SYS826
220 FOR I=0TO6
225 REM THE NEXT STATEMENT CALCULATES ¬
    ¬THE TIMES
226 REM THE MEMORY LOCATIONS FOR THE ¬
    ¬TIMES ARE (995,1002,1009)(996,
    ¬1003,1010),
227 REM CONTINUING UP TO (1001,1008,
    ¬1015)
228 REM EACH TIMING CYCLE TAKES 43 ¬
    ¬MACHINE LANGUAGE STEPS OR 43 ¬
    ¬MICROSECONDS
230 T(I,J)=43*(PEEK(995+I)+PEEK(1002+I)*
    ¬256+PEEK(1009+I)*256*256)/1000000
240 REM THE NEXT STATEMENT ROUNDS OFF ¬
    ¬THE TIMES TO 1/10000 S
250 T(I,J)=INT(T(I,J)*10000)/10000
260 PRINT T(I,J),
270 NEXT:PRINT:PRINT:NEXT
280 REM CALCULATE THE AVERAGE TIMES
290 PRINT"AVERAGE TIMES"
300 FOR I=0TO6:TM(I)=0:FOR J=1TONR
310 TM(I)=TM(I)+T(I,J)
320 AV(I)=TM(I)/NR
330 AV(I)=INT(AV(I)*10000)/10000
340 NEXT:NEXT
350 FOR I=0TO6:PRINTAV(I),:NEXT:PRINT
400 GOTO 205
```

©

COMPUTE!'s Listing Conventions

Many programs which are listed in **COMPUTE!** use cursor control keys, color keys, and so forth. We have established a listing convention which we believe eases the task of typing programs in accurately.

PET/CBM/VIC Conventions

Generally, PET/CBM/VIC programs will contain bracketed words for any special characters: [DOWN] means the cursor-down key; [3 DOWN] means type the cursor-down key three times.

If a program line runs over onto the next line down, the ~ symbol indicates where the line broke (in case the number of spaces is unclear between quotes). An underline means that that key is shifted.

8032/Fat 40 Conventions

SET WINDOW TOP	[SET TOP]
SET WINDOW BOTTOM	[SET BOT]
SCROLL UP	[SCR UP]
SCROLL DOWN	[SCR DOWN]
INSERT LINE	[INST LINE]
DELETE LINE	[DEL LINE]
ERASE TO BEGINNING	[ERASE BEG]
ERASE TO END	[ERASE END]
TOGGLE TAB	[TGL TAB]
TAB	[TAB]
ESCAPE KEY	[ESC]

All Commodore Machines

CLEAR SCREEN	[CLEAR]
HOME CURSOR	[HOME]
CURSOR UP	[UP]
CURSOR DOWN	[DOWN]
CURSOR RIGHT	[RIGHT]
CURSOR LEFT	[LEFT]
INSERT CHARACTER	[INST]
DELETE CHARACTER	[DEL]
REVERSE FIELD ON	[RVS]
REVERSE FIELD OFF	[OFF]

Review: Screen Pro

Edward K. Crossman, Ph.D.
Logan, UT

As a behavioral scientist with both teaching and research responsibilities, I am always looking for ways to save time. Yet, so many of the applications touted by those in the computer field could be more quickly accomplished with a pencil and a stack of 3x5 cards. However, Screen Pro from Kansas City Computers, Inc. has saved me time, and, in this article, I will describe some of the features of this program, and how I applied it.

My home system consists of a CBM 8032 computer, a CBM 8050 disk drive, and an Epson MX-80 printer.

The data from my behavioral experiments (and many others) are expressed in a relative frequency distribution: 10% of the subjects exhibited behavior X, 20% exhibited behavior Y, etc. Typically, these data are represented in tabular form. When you have multiple experimental conditions and, thus, many tables, it is difficult to see trends or changes in the data. So, a bar graph, or histogram (see Figure 1) is often used instead. It is much easier to visually scan a series of histograms and detect changes in the data than with many tables. But how could I convert my frequency data into a histogram quickly and simply and have it printed out on my MX-80? That is where Screen Pro came to the rescue.

Screen Pro, written in BASIC and machine language, allows you to create text or graphics on the screen, and then prints out what was on the screen. Also, it can save the information as a screen file on disk; each screen file occupies eight blocks on disk. Once the screen file is on disk it can be recalled and edited as you see fit. I have found the editing functions adequate for my purposes. These functions for the CBM 8032 (in addition to the normal editing functions) include:

Set upper case
Set lower case
Scroll screen down
Expand Screen
Compress Screen
Insert a line
Delete a line
Erase to end of line
Erase from beginning of line
Enbable/disable quote mode (for graphics)
Send screen contents to printer (normal mode)
Send screen contents to printer (squeezed)
Abort current file on screen, retaining original if editing
Normal exit of editor

Screen Pro has some other nice features. When editing a screen file, it uses a temporary scratch space on disk, so you can either save or scrap the screen file you are currently editing, and at the same time keep a backup of the original screen file as it was before you started editing. Also the author of Screen Pro, Keith Peterson, has gone to some trouble to explain how, if you understand BASIC programming, to create your own programs using his machine language subroutine. Being an amateur programmer, I don't fully grasp his instructions, but perhaps you can. He has taken a very refreshing approach by not protecting any of the software, so you can examine it and change it to your specifications. Essentially, if you write your own program, you can create hundreds of screen files with the ability to branch from one file to many others in the series. Mr. Peterson has provided the would-be programmer with several demonstrations of how to do this. In essence, then, you could create a sort of information management program, although it would lack many features of a typical data base management program, such as Create-A-Base, or Commodore's Ozz.

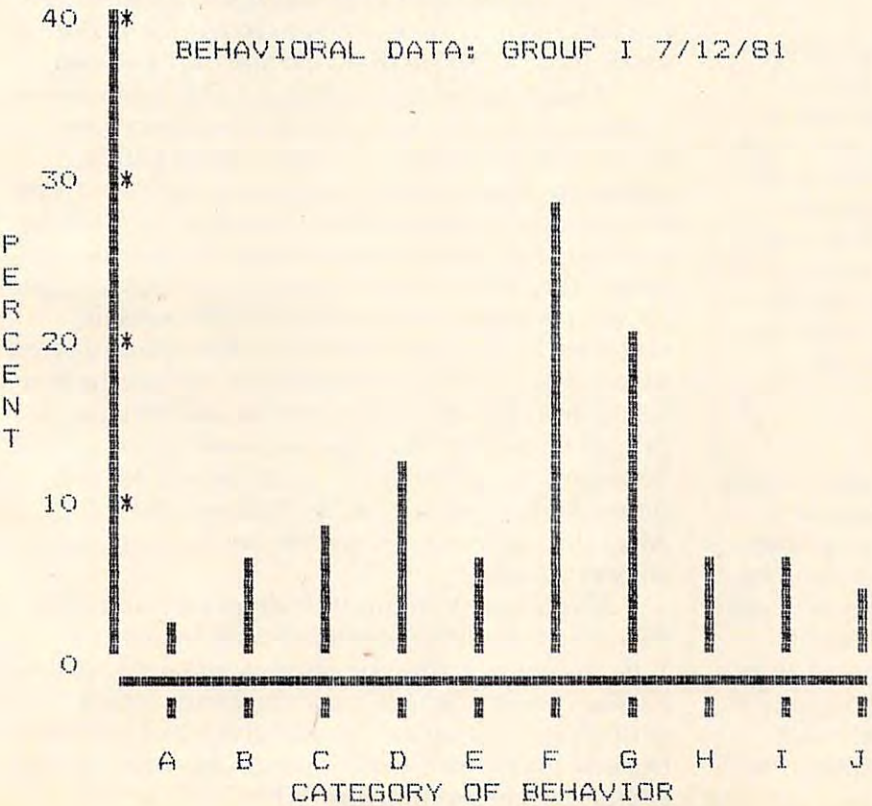
There are a couple of things I have not been happy with, however. First, the documentation, while better than some I have seen, is written by the programmer for people who already have some knowledge of programming and computers. My impression of many in the micro industry is that they consider quality documentation unimportant, yet there are people around who are trained to explain computerese in everyday language; usually, however, they are not programmers!

The second problem concerns the Epson printer. Screen Pro was designed only for the Commodore 2022 and 4022 printers, and the author states so and shows you the section of the program where you can write a routine for your own printer. Without such a routine, however, when text is placed on the screen the Epson prints upper case characters as Epson graphic characters; lower case characters are printed correctly, but capitalized. This is a serious problem for me, and I guess the only thing to do is to write a routine to handle the case conversion.

The Epson also presents a problem for graphics applications, such as mine, because it lacks the Commodore graphics characters. However, I did figure out a simple solution for my histogram application. When one of the Commodore graphics characters is placed on the screen and printed on the Epson, it comes out as one of the Epson block-type graphics characters. So I placed all of the

Commodore graphics characters, available from the 8032 keyboard, on the screen and then printed these on the Epson. From the mish-mash that resulted I was able to pick out suitable symbols to create the simple vertical and horizontal lines that are required in constructing a histogram. For example, the letter "J" on the 8032 keyboard produces a vertical bar on the Epson when the 8032 is in the graphics mode. This method produces funny pictures on the 8032 screen, but this has proved to be only a minor annoyance since there is enough resemblance to the final product produced by the Epson.

Figure 1. Histogram produced by Screen Pro



Overall, Screen Pro is a welcome addition to the software collection for the Commodore systems, and at $39.95 is very reasonably priced. It is fast, relatively easy to use, and fills a very specific need that I have had for some time. Now, if someone would just come up with a similar package which would allow the user to produce a graph rapidly, such as a sine wave, on the screen and have the image come out on an X-Y plotter, such as the MIPLOT! ©

VIC-20 Update

Extended VIC-20 Input Devices: Paddles And The Keyboard

Mike Bassman and Salomon Lederman
Woodside, NY

The VIC-20 has some remarkable capabilities not documented by the manual. Specifically, you can use game paddles with the VIC-20 as well as making better use of the keyboard.

The Paddles

Have you ever seen the little nine pin port right next to the power switch? This port can be used with paddles. To make life easy, it can be used with the widely available Atari game paddles (which are used with their video games and home computers). Just plug in a pair, and we'll be ready to begin. These paddles are *linear* devices. What is meant by this is that the paddle is a much more sensitive device than a directional joystick, which can only point in eight or so directions. You may think the paddle is not even as good, pointing only left or right. This is not true.

How It Works

What the paddle actually does is isolate one position out of the 256 possible ones. When the paddle is turned to the far right, this value is 0. Every time you turn the paddle in either direction the number is increased or decreased accordingly. The VIC-20 allows us to use up to two paddles. For each of them, we can obtain a position value. These values are in memory locations $9008 for the first paddle, and $9009 for the second. In decimal these are 36872 and 36873, respectively (A number preceded by a "$" signifies that it is hexadecimal).

How To Do It

Shown below is a quick one liner that prints out the values of both the paddle registers.

```
10 PRINT PEEK(36872);PEEK(36873): GOTO 10
```

Try typing and running this program now. You should see a continuous stream of two numbers flying by. Fiddle with the paddles. The numbers should change accordingly. The more you turn a paddle left, the higher the number goes (the opposite for right, of course).

Next, we'll try something a little more complicated and which might be more applicable. Program 1 will move a little ball across the screen according to your paddle position. It will also slide a musical tone up and down at the same time. Here are some notes which will explain some of the program. The first two lines are just set-up, setting volume for the tone generator and clearing the screen. Line 20 gets the initial paddle position. The next line, 30, determines the position of the ball on the screen. The ball can move from the far left edge of the screen (7900) to the far right (7921). Logically, the thing to do is to move the ball a little bit left whenever the paddle value goes a little bit up (turning towards the left). The problem is that the paddle is much more accurate than one line of the screen.

While the paddle has 256 possible calibrations, one line of the screen is only 22 characters long. What we do is to make a proportion of paddle calibrations per screen character, in this case 11.64 (obtained by dividing 256 by 22). Now we have the position of the ball on the screen. Line 40 does almost the same thing, finding an appropriate tone for the paddle position. We have 128 possible tones, so the proportion of the calibrations to tones is only 2 to 1. The next three lines just put the ball on the screen, tack a color onto it, and turn on the proper tone. Only the clean-up work remains to be done now: a small delay loop so the ball doesn't flicker badly, and erasing the ball and the color. After this, we get a new paddle reading and start all over again.

If you have run this little demonstration, the advantage of a linear device should be obvious. You can just whip the paddle back and forth without having to worry whether the computer is fast enough to keep up with you, and the ball will follow because the paddles determine an absolute position, rather than just a direction. This could be very

SOFTWARE · HARDWARE FOR THE VIC 20®

VIC 20 is a Registered Trademark of Commodore Business Machines

TAPE CASSETTE GAMES

- **GLOBBLER** — $24.95
  The Conqueror Worm gobbles globs under your control.
- **THE ALIEN** — $24.95
  Fast work with inflatable alien traps makes THE ALIEN an exciting game.
- **AMOK** — $18.95
  The Halls of AMOK are populated by crazed killer robots bent on your destruction!
- **ALIEN BLITZ** — $24.95
  Split second reactions at the superhuman 9th level of ALIEN BLITZ will find how good you really are.
- **FABULOUS FOUR I** — $59.95
  Four of our most popular and exciting game cassettes in one package.
  STAR WARS Blast that tie fighter!
  DRAGON MAZE Lurking man-eating dragon.
  LASER WAR Blast away space debris and aliens.
  INVADER FALL Alien paratroopers descend from space.

EDUCATIONAL TAPES

- **SUPER HANGMAN** — $18.95
  Superior version of the popular spelling game.
- **SKYMATH** — $14.95
  Decimal addition and subtraction with super graphics.

CARTRIDGE PROGRAMS

- **UMI RS232 COMMUNICATIONS INTERFACE** — $49.95
  Our RS232 Communicator Interface enables the VIC-20 to talk to most RS232 devices.
- **UMI 3K RAM MEMORY EXPANDER** — $79.95
  Our 3K Ram Memory Expander adds two ROM slots and 3K of RAM memory to the VIC-20
- **UMI 8K RAM CARTRIDGE** — $89.95
  Our 8K RAM Cartridge adds 8K of user memory to the VIC-20.

DEALER INQUIRIES INVITED

VISA AND MASTERCARD ACCEPTED

United Microware Industries, Inc.
3431 H Pomona Blvd.
Pomona, CA. 91768

Please send me my FREE CATALOG.

NAME____________________
ADDRESS____________________
STATE____________________
CITY____________ ZIP____________

FOR ADDITIONAL PRODUCTS SEND FOR FREE CATALOG *

"Probably the best-documented programs I've seen for the PET/CBM."

- Robert Baker, September 1981 KILOBAUD

HESBAL : Assembler/editor for PET and VIC
HESCOUNT, HESLISTER : Debugging tools for PET and VIC
HESPLOT : Hi-res graphics subroutines for VIC
HESCAT : Disk cataloging system for PET
HESCOM : Exchange data, programs between PET and VIC

VIC

HES

Human Engineered Software
3748 Inglewood Blvd. Room 11
Los Angeles, California 90066

(213) 398-7259
24 HOURS
SOURCE ID:
TCK308

PET, CBM, and VIC are trademarks of Commodore convenient in games where speed is at a premium. In the near future, I'm sure we'll see many clever and innovative ways to use the paddles.

The Keyboard

There are two types of keyboards: ASCII, or hardware keyboards, and *polled,* or software keyboards. The ASCII keyboard is a separate device from the computer which just sends out the ASCII value of the key being pressed. The polled keyboard is a little more subtle. A polled keyboard is split up into sections of eight keys, called rows. Generally, a polled keyboard has eight rows. The computer can test one row at a time, and detect which key along with row is being pressed, if any. The polled keyboard can also detect any number of key combinations along any particular row. Consequently, polled keyboards need a fair amount of system software to do what comes naturally to an ASCII keyboard.

Most microcomputers today, the VIC-20 included, use polled keyboards because of the added flexibility and lower price. Unfortunately, the VIC-20 does not let us normally get at some of those nice features. To us, from BASIC, it seems just like an ASCII keyboard. We can only obtain one character at a time using the GET command. If two keys are being pressed down at once, the GET command will almost randomly choose one of those two as the value that gets sent back to the user. If you wanted to do a two player game or a game requiring simultaneous depressing of more than one key, life would be very difficult. But here's how it can be done.

Polled Keyboard Encoding

The VIC-20 polled keyboard has eight rows of eight keys each. Each row can be selected by a particular value. The eight values for the eight rows are all shown in Table 1. These values are by no means arbitrary. If you examine the table, you can see that the values are given in binary, as well as decimal and hexadecimal. Row values were made by turning on all the bits in the byte, then turning off the bit which the row represents. For example, the first row has all the bits on (set to 1) except for the one on the far left, which is off (or 0). Then this binary number is simply used in its hexadecimal or decimal form to represent the row. Each key along the row is handled in exactly the same manner as the rows (for example, the value representing the first row would be the same as the one representing the first key in that row). This is a little confusing, but it works out well in the end. Table 2 is the keyboard encoding matrix. It shows all the row values going down, and all the keys along each row, and their value. For instance, the keys on row 223 are F3,=,:,K,H,F,S, and Commodore. The value of the Commodore key would be 254.

Implementing Keyboard Theory

Using an individual row on the keyboard is accomplished as follows. You select a row by POKEing its value into a memory location we'll call the row select register. Then you can get the information as to which key(s) is hit by PEEKing another location, the keyboard data register. The row select register is located at $9120 (37152), and the data register at $9121 (37153).

Things don't work out as easily as doing just one POKE, then another PEEK. The problem, in this case, is the RUN STOP ROUTINE. This part of BASIC is the one that checks if you hit this key during the execution of a program. If you have, the program stops. What the routine does is, after every command executed, it puts a 247 in the row select register (the row which has the RUN STOP key) and checks the data register for a value of 254 (eighth key over). If the data register is 254, then you have hit the RUN STOP key, and program execution terminates.

What this means for us is that, even after we have just chosen a row by POKEing a value into the select register, the RUN STOP routine will change it right back to a 247. Very bad news indeed, unless you only want to use row 247. Not only that, but you can't use the RUN STOP key for your own purposes. There is a way to disable the RUN STOP key. POKEing 808 with 114 turns off the RUN STOP key, and POKEing 808 with 112 turns it back on again. This does not solve our problem. Turning off the RUN STOP key will prevent it from ending program execution when that key is hit, but the routine still stores that 247 in the select register. However, when we clear up the major problem, turning off the RUN STOP key will allow us to use that key in our programs.

A Solution

The way to solve this problem is by noticing that this routine operates after every BASIC command. What must be done is to POKE in our select value, then PEEK the data register, all in the time of less than one BASIC command. Machine language is the answer. The VIC-20 can use machine language even though it has no direct facilities for entering or saving it. *[See Jim Butterfield's Tinymon in* **COMPUTE!** *#20, pg. 176 which provides a monitor for VIC – Ed.]* We are going to use a very short machine language routine that simply puts our row into the select register, looks at the data register, then puts the contents of the data register into a RAM location that the BASIC program can look into. Program 2 shows just such a machine language program. Not much to it at all, just five lines OF CODE. The first instruction loads in the row value, in this case a $7F (127). The second stores it in the select register. The next picks up a value from the data register. The last two just store that value in accessible RAM (at $1DFF, or 7679), then returns back to BASIC.

This routine will do the trick because it does what we want in less than one BASIC command. Even though the VIC-20 has no real method for entering machine code data, it can be done anyway. You just take the machine code values, convert them into decimal, and stick them into a BASIC DATA statement. Then just add a line of BASIC that reads the values and puts them in the correct place. In Program 3, we have a complete deomonstration. Lines 30 and 40 are the aforementioned DATA statement and reader/POKER. Line 5 turns off the RUN STOP.

Lines 10 and 20 need a little bit of explanation. We are going to put the machine language routine into the top of available memory. Unfortunately, BASIC also wants to use this space. These lines tell BASIC not to use the highest 21 bytes of RAM. Locations 51 and 52, as well as 55 and 56 contain the top of BASIC RAM in low, high format. Low, high format is when the low byte of an address

Table 1.

127 - 7F - 0111 1111
191 - BF - 1011 1111
223 - DF - 1101 1111
239 - EF - 1110 1111
247 - F7 - 1111 0111
251 - FB - 1111 1011
253 - FD - 1111 1101
254 - FE - 1111 1110

Table 2: Keyboard Matrix Table

Column (POKE) ↓ Row (PEEK) → of Keys

	127	191	223	239	247	251	253	254
127	F7	Home	—	∅	8	6	4	2
191	F5		@	O	U	T	E	Q
223	F3	=	:	K	H	F	S	COMMODORE
239	F1	RIGHT SHIFT	.	M	B	C	Z	SPACE
247	CURSOR	/	,	N	V	X	LEFT SHIFT	RUN STOP
251	CURSOR	;	L	J	G	D	A	CTRL
253	RETURN	*	P	I	Y	R	W	
254	DEL	£	+	9	7	5	3	1

precedes the high byte of it. To calculate an address from this format, just use this formula: (256*high byte + low byte = address). Normally the low and high byte for the top of BASIC are 00 and 30, respectively (yielding an address of 7680). These we change to 235 and 29, giving an address of 7659. Line 50 goes to our machine code routine, line 60 prints the result, and 70 repeats the process. Try it now. I'll wait. If you press one of the keys from the first row, the appropriate value will be printed. No key is indicated by its printing 255. As it is now, this program will print first row values. To change the row, just change the second item of data in line 30. I used this program, incidentally, to make the keyboard matrix chart.

All this may seem pretty useless to you at this point. Our next program will do something that cannot be done with regular old BASIC. Program 4 will play a tone of varying pitch depending on which of two keys you hit. Doesn't sound too exciting, but it will play the two tones one after the other even if both keys are pressed at the same time. This is the basis of two-player games, where the computer can fairly give one turn to each player. All the material in this program should be old hat to you now, so I won't bother to explain it.

Hopefully you've learned to use your paddles and keyboard now. Put them to good use!

Program 1.

```
1 REM Listing 1
5 POKE36878,3
10 PRINT"[Shift/Home]"
20 X=PEEK(36872)
30 L=7921-INT(X/11.64)
40 T=255-INT(X/2):IF T=255 THEN T=
   254
50 POKEL,81
55 POKEL+30720,2
60 POKE36874,T
70 FOR K=1 TO 10:NEXT
80 POKEL,32:POKEL+30720,1
90 GOTO20
OK
```

Program 2.

```
A9 7F     LDA #$7F
8D 20 91  STA $9120
AD 21 91  LDA $9121
8D FF 1D  STA $1DFF
60        RTS
```

Program 3.

```
5 POKE 808,114
10 POKE 51,235:POKE 52,29
20 POKE 55,235:POKE 56,29
30 DATA 169,127,141,32,145,173,33,
   145,141,255,29,96
40 FOR K=1 TO 12:READ X:POKE 7659
   +K,X:NEXT K
50 SYS 7660
60 PRINT PEEK(7679);
70 GOTO50
OK
```

Program 4.

```
1 REM Listing 4
10 POKE808,114:POKE51,235:POKE52,29
   :POKE55,235:POKE56,29:POKE36878,
   3
20 DATA169,127,141,32,145,173,33,
   145,141,255,29,96
30 FOR K=1 TO 12:READ X:POKE 7659
   +K,X:NEXT K
40 POKE 7661,127:SYS 7660
50 IF PEEK(7679)=254THENPOKE36874,
   200:FORK=1TO500:NEXTK:POKE36874,
   0
60 POKE7661,191:SYS 7660
70 IF PEEK(7679)=127THENPOKE36875,
   200:FORK=1TO500:NEXTK:POKE36875,
   0
80 GOTO 40
OK
```

©

Timekeeping

Keith Schleiffer
Annandale, VA

The friendly computer guide that comes in the box with your VIC 20 mentions several interesting features that the casual reader can easily miss. In my most recent rereading, I discovered the timekeeping feature of the VIC. The computer can keep real clock time, count elapsed time, or time controlled pauses during program execution.

The clock is available as the reserved variables TI and TI$. TI actually counts time passing. TI$ is a string variable, which depicts this time count in HHMMSS format (hours, minutes, and seconds, without any punctuation) on a twenty-four hour clock.

How does the VIC do this timekeeping? When the computer is first turned on, the timekeeper initializes at 000000 (midnight). You can then set it to act as a clock by assigning to TI$ a string representing the correct time. For instance, if I initialize the timekeeper as a clock at 1:29:30 in the afternoon, I would enter the statement:

```
TI$ = "132930"
```

The VIC would convert this to 48570 seconds after midnight, multiply by sixty, and assign:

```
TI = 2914200
```

and continue counting from there. TI is counted in one-sixtieth second intervals; that is, when TI has increased by sixty, one second has passed. The time count is kept in memory locations 160, 161, and 162.

Once you have set the correct time, you can check it whenever you wish by entering:

```
PRINT TI$
```

and the VIC will display the time, again in HHMMSS format. I like to set TI$ to keep clock time, and check it occasionally, so my wife doesn't have to complain about getting less attention than the computer. The timekeeper can be used in programming to control operations at scheduled times during the day, such as periodic data-collection from an experiment, or to control your lights in a household security program.

To use the VIC to count elapsed time, you cannot start and stop the time counter. To get around this problem, you must run a second variable to count time in parallel with TI, then stop counting with that second variable when the timed period is over. The following program uses the "hit any key" concept to start and stop timing:

```
100 GET A$ : IF A$ = " " THEN 100
110 TS = TI
120 PRINT "TIMING"
130 TC = TI : GET A$ : IF A$ = " " THEN 130
140 TE = (INT ((TC-TS) / 6 + 0.5)) / 10
150 GOSUB 400 : PRINT T$
160 END
400 REM CONVERTS SECONDS TO HH:MM:SS.S
    FORMAT
410 H1 = INT(((TE /60/60/24) - (INT(TE /60/60/24)))
    * 24)
420 B1 = STR$(H1)
430 H$ = MID$(B$,2,2) : IF H1<10 THEN H$ = "0"
    + MID$(B$,2,1)
440 T3 = TE - (H1 * 60 * 24)
450 M1 = INT(((T3 /60/60) - (INT(T3 /60/60))) * 60)
460 B$ = STR$(M1)
470 M$ = MID$(B$,2,2) : IF M1<10 THEN M$ = "0"
    + MID$(B$,2,1)
480 T2 = T3 - (M1 * 60)
490 S1 = INT(((T2 /60) - (INT(TE/60))) * 60)
500 B$ = STR$(S1)
510 S$ = MID$(B$,2,4) : IF S1<10 THEN S$ = "0"
    + MID$(B$,2,3)
520 T$ = H$ + ":" + M$ + ":" + S$ : RETURN
```

Line 130 converts TE to the elapsed time in seconds and rounds off to the nearest tenth. The subroutine starting at line 400 will convert this to "clock" display, complete with colons in HH:MM:SS.S format, down to tenths of seconds. A simpler approach would use TI$ by assigning to it the elapsed time value and immediately printing it:

```
140 TE = TC-TS
150 TI = TE : PRINT TI$ : END
```

You won't want to use this method if you are using TI as a real clock, or if you're relying on the timekeeper to track more than one period at once.

You can use the timekeeper for the scoreboard in a game, either by displaying stopwatch time or TI$, to show time passing, or by calculating time remaining and displaying a countdown timer. The followinng program is a version of the countdown timer.

```
100 PL = 5 : REM PERIOD LENGTH 5 MINUTES
110 PS = TI : REM PERIOD STARTS NOW
120 PF = PS + PL * 60 : REM PERIOD FINISH TIME
130 TR = PF-TI : REM TIME REMAINING
140 GOSUB 400
150 PRINT "[clr]" T$
160 IF TI<PF THEN 130
170 END
400 REM CONVERTS SECONDS TO MM:SS
    FORMAT
440 T3 = INT (TR / 60 + 0.5)
450 M1 = INT(((T3 /60/60) - (INT(T3 /60/60))) * 60)
460 B$ = STR$(M1)
470 M$ = MID$(B$,2,2) : IF M1<10 THEN M$ = "0"
    + MID$(B$,2,1)
480 T2 = T3 - (M1 * 60)
```

```
490 S1=INT(((T2 /60) - (INT(TE /60))) * 60)
500 B$=STR$(S1)
510 S$=MID$(B$,2,2) : IF S1<10 THEN S$="0"
    + MID$(B$,2,1)
520 T$=M$+":"+S$ : RETURN
```

The most valuable feature of the timekeeper is the ability to control the length of pauses made during execution, independent of the program lines being executed. The friendly computer guide shows how to make delays by using a FOR ... NEXT loop with the statements:

```
FOR I=1 TO 100 : NEXT I
```

The major problem with this method is that it ties up the whole program while you pause. You can insert program lines for execution during the loop, but then some guesswork and experimenting will be necessary every time you program to obtain the desired pause. Frequently you will have to compromise between the statements you want to execute and the time you can allot to the pause. Finally, if the lines executed during the pause contain the decisions with varying amounts of program to be executed based on the decision, the length of the pause becomes unpredictable.

Getting Control Over Pause

The timekeeper counts independently, on a steady basis, and allows you to assume control of the length of a pause, while permitting other parts of the program to continue. To do this you simply note the time the pause begins and add the desired pause length, giving the time the pause will end. An IF decision watches for the clock to exceed that end time, and you can run other parts of the program while the pause is in progress. The decision watching for the end of the pause must be made with a reasonable frequency, so the number of statements you can execute between repetitions of the end-time decision will depend on how long the pause is and how exact you want the measurement of the pause to be.

As a very conservative rule-of-thumb, allow twenty eighty-character (multiple statement) program lines to reach the end-time decision at an interval of about ten percent of the total pause length. For example, if I pause for about ten seconds, I can allow up to one second, or about twenty program lines. Similarly, a two-second pause will allow up to four program lines between repetitions of the end-time decision. You can use a greater number of lines if they do not contain several statements each.

These time estimates are very rough: do some experimenting yourself to find how many statements you can squeeze in and still get accurate control of the pause length. Once you have established some rules for yourself, they should be useful in all your programming.

As an example of the pause, let's say that I'm writing a game program in which we explore a dungeon. If someone casts a magic spell of darkness, then I want to give no visual clues for the length of the spell – say twenty seconds – while the action of the program continues. The following segment of a program will provide that effect:

```
100 DEF FN PS(T2)=TI+(T2 * 60)
350 REM   THE SPELL IS CAST
360 GOSUB 900 : P1=FN PS(20)
370 REM P1=TIME TO END BLACKOUT
380 REM THE
390 REM  PROGRAM
400 REM    CONTINUES
410 REM     RUNNING
420 REM      WITH A
430 REM       BLACK
440 REM        SCREEN
490 REM  (UP TO FORTY PROGRAM LINES)
775 IF TI>P1 THEN GOSUB 902 : GOTO 800
780 GOTO 380
800 END
900 POKE 36879,8 : FOR I = 38400 TO 38906 :
    POKE I,0 : NEXT I
905 RETURN : REM BLACKOUT MAKER
920 POKE 36879,78 : RETURN : REM BLACKOUT
    LIFTER
```

This application uses the function PS to relate the desired pause length (T2) to a future time value (P1) which defines the end of the blackout.

Another application of the pause timer can limit how often I may perform an action. I'm writing a game in which the player fires a laser cannon that takes five seconds to recharge before it can be fired again. The line which times the firing interval looks like this:

```
350 IF PEEK(197)=35 AND TI>P1 THEN GOSUB
    800 : P1=TI+(5*60)
800 RETURN : REM VISUAL AND SOUND EFFECT
FOR LASER FIRING
```

Here there is no need to worry about running the end-time decision within a set interval – the next time I want to fire the cannon, the logical AND in the decision checks to see if it has recharged. This pause method can also be used in an education program, to limit how soon the student may answer after a question appears, or may try a second time after an incorrect first answer has been entered.

If you're interested in converting existing programs to timekeeper pauses, the statement:

```
FOR I = 1 TO 100 : NEXT I
```

is worth about eight counts on the timekeepeer, or 0.13 seconds. There will be some difference between this statement and a longer loop. For instance, modifying the statement to:

```
FOR I = 1 TO 1000 : NEXT I
```

This is worth 72 counts, or 1.2 seconds, not the eighty counts one might expect. This is because of the "overhead" time needed to establish the loop during execution. There may even be differences between machines. You can check your own timing with this simple program:

```
10 BT = TI
20 FOR J = 1 TO 1000 : NEXT J
30 FT = TI
40 ET = FT - BT
50 PRINT ET, ET/60
```

This displays the time passed in both counts and seconds. Try varying the length of the loop in line 20 to get a general idea of what the "overhead" time is on your computer.

You need to do nothing to the timer to use it as a basis for pauses. However, if you have the VIC on for long periods, or if you set TI$ to keep clock time and run the program near midnight, be careful: if the pause starts before midnight and ends after, you may never reach the end of the pause, since the clock resets to 000000 at midnight. You can put in additional statements to watch for this problem and compensate for it; you can have the program reset the clock to 000000 before timing any pauses; or you can ignore the possibility and hope for the best. The third option, technically unsound as it is, requires the least effort and presents no great threat.

These pause techniques have two important features: controllable pause lengths and the ability to run other, unrelated parts of the program while the pause is in effect. When you develop a program, you can select a length of pause that will not change as you add, change, remove, or relocate program statements. The pause can also be lengthened or shortened to suit your needs, without major changes in the program itself. You have made your pause independent of the program that contains it. At the same time, you can execute lines of an unrelated portion of the program while the pause is in progress, making the program independent of the pause it executes. The timekeeper in the VIC gives the programmer much better control of realism in his game and simulation programs. ©

An Easy Way To Relocate VIC Programs On Other Commodore Computers

Greg Sherwood and Ross Sherwood
Manhattan, KS

BASIC programs that are generated on Commodore's new color computer, the VIC-20, start at memory location 4097 rather than at 1025 as in Commodore's other computers. Thus, if you wish to use features available for some of the other computers such as Toolkit, etc. to edit or modify a program written or stored from a VIC, you need to relocate the program so it starts at memory location 1025.

The following is a description of a quick and simple method of relocation of VIC programs, I will describe two versions, one using the built in monitor and the other done in direct mode.

To relocate a VIC program from the monitor, load the program from tape and then enter the monitor with SYS1024. Next, look at the first part of BASIC memory by typing M 0400 0400. Make the following changes to the displayed memory:

'M 0400 00 01 10 00 00 99 00 XX'
(XX means *doesn't matter.*)

Next exit the monitor by typing an X. Now type LIST and the VIC program should list out with an additional line (line 0) at the beginning: 0 PRINT Finally type "0" and RETURN and the VIC program is relocated and can be edited or modified at will.

To accomplish the same change in direct mode, the following six POKEs are entered:

POKE 1025,1:POKE 1026,16
change link pointers to VIC program
POKE 1027,0:POKE 1028,0
create line #0
POKE 1029,159
put PRINT on line 0
POKE 1030,0
end of line indicator

Now, as above, type LIST and the VIC program will list with the additional line 0 PRINT.

Last, type "0" and the line 0 will be eliminated so the VIC program can be edited.

This method works with both BASIC 3.0 and 4.0 Commodore computers and, through it hasn't been tested on other versions, it should work on those as well. It has been successfully used on both 40 and 80 column machines.

If you should wish to relocate several VIC programs in succession, the following assembly language subroutine can be used. It begins at location 926 in the second cassette buffer and can be called by SYS926. To load this program, enter the monitor and type M 039E 03C8 annd change the memory as follows:

```
039E A9 00 8D 03 04 8D 04 04
03A6 8D 06 04 A9 30 8D 6F 02
03AE A9 01 8D 01 04 A9 10 8D
03B6 02 04 A9 99 8D 05 04 A9
03BE 0D 8D 70 02 A9 02 85 9E
03C6 60 00 XX XX XX XX XX XX
```

This program can be saved on tape or disk by saving from 039E to 03C8 and then can be loaded in anytime and used to relocate VIC programs with a SYS926 command until the machine is turned off or the second cassette buffer is used for some other purpose. This subroutine is located high enough in the second cassette buffer that disk operations don't overwrite it.

This subroutine automatically erases line 0 and so that, when you return to BASIC, the VIC program is moved and ready to be edited, etc. without the necessity of removing line 0.

```
039E A9 00        LDA #$00
03A0 8D 03 04     STA $0403
03A3 8D 04 04     STA $0404
03A6 8D 06 04     STA $0406
03A9 A9 30        LDA #$30
03AB 8D 6F 02     STA $026F
03AE A9 01        LDA #$01
03B0 8D 01 04     STA $0401
03B3 A9 10        LDA #$10
03B5 8D 02 04     STA $0402
03B8 A9 99        LDA #$99
03BA 8D 05 04     STA $0405
03BD A9 0D        LDA #$0D
03BF 8D 70 02     STA $0270
03C2 A9 02        LDA #$02
03C4 85 9E        STA $9E
03C6 60           RTS
03C7 00           BRK
03C8 00           BRK
```

©

Review:

UMI Amok For VIC

Harvey B. Herman
Associate Editor

One of the reasons people buy personal computers is to play games. I confess that this reviewer is no exception. In contrast to all the serious applications of computers, it is still fun to relax and play an occasional interesting and challenging game. The Commodore VIC is particularly suited for game playing as it comes with an interface for a joystick and can play sound effects through the TV speaker. Unusual displays are also possible because one is not limited to a standard character set composed of letters, numbers, and graphics. A knowledgeable user can define a new set for special effects.

Until now, the VIC games I have previewed have, for the most part, been good, but nevertheless have not taken full advantage of all the VIC's capabilities. All of them have been written in BASIC which can be too slow for good animated displays. Machine language usually looks much more realistic. The AMOK program was a pleasant surprise. The author, Roger Merritt, seems to have done everything right. His machine language program uses the features of the VIC to advantage. My kids, on whom I rely to advise me on game programs, rated it a 9 (out of 10 possible). I did enjoy playing with it myself, but not as much as they did.

You Against The Angry Robots

The game works like this: you are in a partitioned room with angry robots. The robots, shown in various colors, are shooting at you and you, of course, are dodging and returning their fire. You get three chances before the game is over. You lose a chance whenever your character touches the walls, partitions, or robots. If the robots hit you with their fire, you also lose a chance. Your character is controlled with a joystick or the keyboard. Other rooms can be entered (the door closes behind you) where you encounter a new set of differently colored robots. The object of the game is to score the most points. The color of the robot determines its point score and there are bonus scores. The difficulty of the game is set at the beginning.

I think, in all fairness, our enthusiasm is partly due to some of the relatively pedestrian VIC programs which we have previously seen. Your character in this program is a sight to behold. I have not seen graphics this good on either the VIC or the PET and it approaches animations I've seen on dedicated video games. I recommend this program highly, particularly if you have game playing kids. I am told it is similar to the arcade game BEZERK. An adult may not stay interested for hours, but a kid will. Think of all the quarters you will save.

Hints by Herman – if you have added the 3K memory expansion, a special load sequence is necessary. Use:

```
LOAD " ",1,1
SYS 4110
```

This is not in the instructions.

United Microware Industries
3431 H Pomona Blvd.
Pomona, CA 91768
$18.95

©

Review:

UMI 3K VIC Memory Expansion

Harvey B. Herman
Associate Editor

This small circuit board is designed to plug into the 44 contact female edge connector inside the VIC. It adds 3K of RAM memory to the 5K normally present. When the VIC is powered up, the bytes free message should now total 6655 instead of 3583. This means that programs loaded from tape can be almost twice as long as before without running out of memory. The board also has two empty ROM sockets which allow up to 16K of programs on chips to reside permanently in memory. These programs do not disappear when power is turned off or lost. Initial startup with ROM-based software is much more convenient compared with a long program loading from tape. Many ROM games and other interesting ROM programs will be marketed by UMI and other companies. This board will allow you to use them without additional expense.

The circuit board is easily installed. The VIC case does not even have to be opened as the board fits through the opening in the rear. If you read and follow the quite explicit directions, I predict you won't have any difficulty.

It's Solidly Constructed

I have several positive comments. The board looks solidly constructed and seems to be well thought out. Each ROM can be placed in one of two areas of memory and three ROM sizes can be accommodated. The instructions even give technical hints to advanced hobbyists who intend to program their own ROMs. The price is competitive with similar boards I have seen advertised, but have not yet examined critically.

I have two minor negative comments. Contact fingers on the board are not gold plated so corrosion could be a problem under certain conditions. This will probably not be a concern in typical household use, however. The board sticks out a little from the back of the VIC and is unprotected. Users will have to take care that it does not get knocked about. Again, I do not see this as a serious problem.

As you may have inferred, I am quite happy with this memory expansion. I felt somewhat limited by the small amount of available RAM in the original VIC. Now I can run more ambitious programs. I am also looking forward to installing the better ROM-based programs when they become available.

United Microware Industries
3431 H Pomona Blvd.
Pomona, CA 91768
$79.95

©

Alphabetizer

Jim Wilcox
Vienna, WV

The following program will alphabetize letters or put numbers in order from lowest to highest. The first thing that will happen is the screen will clear and the message "HOW MANY VARIABLES?" will appear on the screen. You then type in the number of names you wish to sort. The variable "VAR" will take on the value typed in. The statement at line number 20 will set the amount of variables of the dimensioned variable "A$". If you are stuck on dimensioned variables, read on.

DIMensioned Variables

Dimensioned variables can be compared to houses on a street. Let's say the house numbers on this block start at one and end at ten. They all belong to the street named, say "Washington." To make things easier than naming each house after a different president, they are given numbers. There might be another house with the number two, but not on the same street. The name of the street is the variable, but there is more than one house on the street, which are variables too. To get a letter to house #2 on Washington Street, one would have to write the person's name, "Jones," who would reside at 2 Washington Street. In a computer program, one could set the variable WASHINGTON$(2) = "JONES". 1 Washington Street might have the "George's" living there so the variable would be WASHINGTON$(1) = "GEORGE". So a dimensioned variable is a variable that has other variables related to it, i.e. all the people on the block have in common the fact they live on Washington Street.

I recommend that you try a small list first, such as ten of the letters of the alphabet mixed up. This will not take long to put the characters in order and the programmer can tell whether the program was typed in properly. On longer lists it becomes tempting to hit the RUN/STOP key to see if the computer is stuck in an endless loop, but the longer the list, the longer it takes.

```
10 INPUT"{CLEAR}HOW MANY VARIABLES"
   ;VAR
20 DIMA$(VAR+22)
30 FORA=1TOVAR
40 PRINT"#"A;
50 INPUTA$(A)
60 NEXT A
70 PRINT"ALPHABETIZING"
80 FORA=1TOVAR-1
90 FORB=A+1TOVAR
100 IFA$(B)<=A$(
    A)THENSM$=A$
    (B):A$(B)=A$(
    A):A$(A)=SM$
110 NEXT B
120 NEXT A
130 PRINT"FINISH
    ED ALPHABETI
    ZING"
140 POKE36878,8
150 POKE36874,25
    0
160 FORA=1TO500
170 NEXT A
180 POKE36878,0
190 POKE36874,0
200 FORA=1TOVAR
    STEP22
210 FORB=ATOA+21
220 PRINTA$(B)
230 NEXT B
240 GETA$:IFA$="
    "THEN240
250 NEXT A
260 END
```

©

COMPUTE!'s Listing Conventions

Many programs which are listed in **COMPUTE!** use cursor control keys, color keys, and so forth. We have established a listing convention which we believe eases the task of typing programs in accurately.

PET/CBM/VIC Conventions

Generally, PET/CBM/VIC programs will contain bracketed words for any special characters: [DOWN] means the cursor-down key; [3 DOWN] means type the cursor-down key three times.

If a program line runs over into the next line down, the ~ symbol indicates where the line broke (in case the number of spaces is unclear between quotes). An underline means that that key is shifted.

VIC Conventions

SET COLOR TO BLACK	[BLK]
SET COLOR TO WHITE	[WHT]
SET COLOR TO RED	[RED]
SET COLOR TO CYAN	[CYN]
SET COLOR TO PURPLE	[PUR]
SET COLOR TO GREEN	[GRN]
SET COLOR TO BLUE	[BLU]
SET COLOR TO YELLOW	[YEL]
FUNCTION ONE	[F1]
FUNCTION TWO	[F2]
FUNCTION THREE	[F3]
FUNCTION FOUR	[F4]
FUNCTION FIVE	[F5]
FUNCTION SIX	[F6]
FUNCTION SEVEN	[F7]
FUNCTION EIGHT	[F8]
ANY NON-IMPLEMENTED FUNCTION	[NIM]

CAPUTE:

Corrections and Amplifications

— Issue #16, pg. 107, "The Unwedge": in line 116, the final datum should read H259, not H25.

— Issue #18, pg. 60, "Bits, Bytes and Basic Boole" the following lines should be changed in the program listing:

```
1060 FOR A1=0 TO 3
1070 A2=2↑A1
1090 SC(0,A3,A1) = SA(A3 OR A2)
```

— Issue #18, pg. 118, "Assembler Update" was missing the following program and was inadvertently in the Atari Gazette. Mr. Brandon's Assembler for the PET (which originally appeared in Issue #13, pg. 120) can be upgraded with the following modifications to permit LOAD/SAVEs of source code to disk:

Program 2.

```
3ØØ  PRINT"{DOWN}{REV}I{OFF}NPUT
     REV}D{OFF}ELETE I{REV}N{OF
     OFF}SERT {REV}L{OFF}IST {R
     REV}S{OFF}AVE L{REV}O{OFF}
     AD"
3Ø5  PRINT"{REV}A{OFF}SSEMBLE {REV}Q
     {OFF}UIT"
36Ø  IF CM$="Ø"THEN11ØØØ
37Ø  IF CM$="S"THEN12ØØØ
11ØØØ  INPUT"FILENAME ";FL$
11Ø1Ø  OPEN8,8,2,"Ø:"+FL$+",S,R"
11Ø2Ø  FORT=1TOMEM
11Ø3Ø  GET#8,IO$:IFIO$=CHR$(13)THEN11Ø
     5Ø
11Ø4Ø  A$(T)=A$(T)+IO$:GOTO11Ø3Ø
11Ø5Ø  NEXTT
11Ø6Ø  CLOSE8
11Ø7Ø  GOTO3ØØ
12ØØØ  INPUT"FILENAME ";FL$
12Ø1Ø  OPEN8,8,2,"Ø:"+FL$+",S,W"
12Ø2Ø  FORT=1TOMEM
12Ø3Ø  PRINT#8,A$(T);CHR$(13);
12Ø4Ø  NEXTT
12Ø5Ø  CLOSE8
12Ø6Ø  GOTO3ØØ
```

— Issue #18, pg. 148, "Inversion Partitioning" will run on the Original ROM PET with the following lines changed (our thanks to Lou Sander):

Program 3.

```
Ø33A A2 ØØ EØ ØØ DØ 1D A5 87

Ø362 A5 Ø2 48 A5 66 48 A5 67

Ø372 Ø2 A9 FE 85 66 A9 3F 85
Ø37A 67 A1 Ø1 48 A1 66 81 Ø1
Ø382 68 81 66 E6 Ø1 C6 66 E4
Ø38A Ø1 DØ EE E6 Ø2 C6 67 A5
Ø392 67 C9 21 DØ E4 68 85 67
Ø39A 68 85 66 68 85 Ø2 68 85
Ø3A2 Ø1 38 A9 FF E5 7C 48 A9
Ø3AA 43 E5 7D 48 A9 FF E5 86
Ø3B2 85 7C A9 43 E5 87 85 7D
Ø3BA 68 85 87 68 85 86 EC 3D

Ø3CA 8E Ø1 Ø4 8E Ø2 Ø4 85 6F
Ø3D2 85 7D A9 Ø1 85 6E A9 Ø3
Ø3DA 85 7C A5 7C 85 7E 85 8Ø
Ø3E2 A5 7D 85 7F 85 81 A5 86
Ø3EA 85 82 A5 87 85 83 6Ø ØØ
```

©

SOFTWARE • HARDWARE

FOR THE VIC 20®

COMMUNICATIONS AND COMPUTER TOOLS

- **UMI RS232 COMMUNICATOR INTERFACE** — **$49.95**
  Our RS-232 Communicator Interface enables the VIC-20 to talk to most RS232 devices.
- **VICTERM A** — **$19.95**
  With the VICTERM A program, begin telephone communications.
- **UMI 3K RAM CARTRIDGE** — **$39.95**
  The easiest way to have 6655 bytes of program memory in your VIC.
- **UMI 3K RAM EXPANDER** — **$79.95**
  For programmers, ROM developers and advanced users, our 3K RAM EXPANDER adds two ROM slots and 3K of RAM memory to the Commodore VIC-20 to give a total of 6655 bytes of available user memory. An 8-position dipswitch provides switch - selectable base address and single-socket enable/disable to two ROM sockets.
- **UMI 8K RAM CARTRIDGE** — **$89.95**
  Our 8K RAM CARTRIDGE adds 8K of user memory to the VIC-20.
- **UMI SOCKETED 8K RAM/ROM BOARD** — **$29.95**
  Our 8K board has four sockets for mixing and matching RAMS, ROMS, or EPROMS.
- **BUTI** (BASIC PROGRAMMER'S UTILITY ROM) — **$34.95**
  A beauty of a ROM that plugs into CVH-0002, our 3K RAM EXPANDER, or CVU-0006, our RAM/ROM/EPROM Board.
- **UMI VIC-20 REFERENCE CARD** — **$4.95**
  Our 8-1/2" by 11" laminated reference card is packed with commonly needed Commodore VIC20 information.

PROGRAMS FOR THE COMMODORE PET®

- **SATELLITES AND METERORITES** — **$49.95**
  Satellites and Meterorites is a greater challenge than most space pilots have met.
- **SKYMATH** — **$12.95**
  Decimal addition and subtraction with super PET graphics!
- **MATH SERIES DISKETTE I** — **$29.95**
  Pedagogically correct programs SKYMATH, SPACEDIV, and LONGDIV, on a 5-1/4" Diskette for Commodore Disk Drives.
- **SPACEDIV** — **$12.95**
  Division practice with graphics that hold the child's interest.
- **LONGDIV** — **$14.95**
  20 problems of step-by-step long division with super graphics and 1-inch high digits.
- **SUPERHANGMAN** — **$16.95**
  Superior graphic version of the popular spelling game.

GAMES AND ENTERTAINMENT

- **ALIEN BLITZ** — **$24.95**
  Split second reactions at the superhuman 9th level of ALIEN BLITZ, will find how good you really are.
- **SPIDERS OF MARS** (CARTRIDGE) — **$59.95**
  You are a trapped space fly, fire at the horde of Martian Spiders wending their way to satisfy their urge to devour you.
- **BLASTOIDS** — **$18.95**
  Blast the cruising junk from space miner's rock piles to smithereens before it crashes into you.
- **A-MAZ-IN** — **$18.95**
  Fast action color game for one or two players.
- **AMOK** — **$18.95**
  The halls of AMOK are populated by robots that obey one instruction — Get the intruder!
- **ROBOT BLASTERS** — **$39.95**
  Only blasts from your lasers can breach the near-impregnable rows of walls circling your only means of escape.
- **ASTRO TRANSPORTER** (CARTRIDGE) — **$59.95**
  As Captain your duty is to use your powerful thrustors to guide space barges to a soft landing on Earth.
- **3D INVADERS** (CARTRIDGE) — **$59.95**
  A new view — from the wide-angle viewport of your movable laser. Tiers of attackers march back and forth above you, dropping bombs.
- **MASTERWITS** — **$9.95**
  Try to deduce the pattern of four markers, using 6 colors, MASTERWITS hides from you!
- **FABULOUS FOUR I** — **$59.95**
  Four of our most popular and exciting game cassettes in one package.
  - **STAR WARS** Blast that tie fighter!
  - **DRAGON MAZE** Lurking man-eating dragon.
  - **LASER WAR** Blast away space debris and aliens.
  - **INVADER FALL** Alien paratroopers descend from space.

EDUCATION

- **SUPER ADDITION, SUBTRACTION, DON'T FALL** — **$14.95**
  One cassette contains all three educational games.
- **SKYMATH** — **$14.95**
  With excellent graphics, SKYMATH shoots 5-digit, 3-place addition and subtraction problems into screen's "sky" in 1-inch high digits.
- **SPACEDIV** — **$14.95**
  Division practice with graphics that hold the child's interest.

GAME PROGRAM CARTRIDGES NOW AVAILABLE—ORDER NOW!

- **DEALER INQUIRIES INVITED**
- **ATTENTION SOFTWARE DEVELOPERS**
  Please contact UMI for distribution and top royalties. Cartridge kits available in quantity.
- **MASTERCARD/VISA Accepted**

united microware industries, inc.

• 3431 H POMONA BLVD. • POMONA, CA 91768 • PHONE (714) 594-1351 •

VIC 20 and PET are Registered Trademarks of Commodore Business Machines

Clip Coupon for FREE CATALOG

united microware industries inc.
3431 H Pomona Blvd
Pomona, CA 91768

Please send me my FREE CATALOG describing your Hardware and Software Products

NAME ____________

ADDRESS ____________

CITY ____________ STATE ______

ZIP ________ COMPANY ____________

New Products

Board Offers 24K Additional Memory For VIC 20

Quantum Data Inc. of Costa Mesa, CA announced the first of its new products to expand the capabilities of the Commodore VIC 20 personal computer: A Board allowing up to 24K of additional user memory. Designated the QDI Expandor, it expands the memory of the VIC to a total of 29K bytes.

Designed for those users who need more than the 3K of user RAM available on the VIC, the QDI Expandor is available in several configurations from 0K to 24K of additional memory. The board uses state of the art memories which allow the board to consume less than 150 MA even when fully loaded with RAM. These memories are also pin compatable with popular EPROMs and ROMs and they can be mixed and matched in 8K blocks.

In its standard configuration, the QDI Expandor uses memory from HEX 2000 to 7FFF but it may also be jumpered to operate one of the 8K blocks in the A000 to BFFF range, "Where the ability of the board to handle ROMs is very convenient," explains Dick Edwards, QDI's president. "That's because the VIC starts looking at location A000 on power up to see if there is a machine language program present in ROM, and, if there is, it will run that program. We expect this feature to be of special interest to systems houses, who are going to use this computer where a resident machine language program is important."

Measuring 4.5 by 6 inches, the QDI Expandor can plug directly into the VIC memory expansion port or in the expansion chassis that QDI will soon be announcing. Priced at a suggested retail of $295.00, the board is available from stock.

For further information contact: Quantum Data, Inc., 3001 Redhill, Bldg. 4, Suite 105, Costa Mesa, CA 92626.

PDI Announces Publication Of Do-It-Yourself Spelling

Program Design, Inc., the Greenwich, Connecticut, firm that specializes in the design, development, and marketing of educational courseware for microcomputers, has just published a new spelling program entitled DO-IT-YOURSELF SPELLING. Unlike other spelling programs on the market, Do-It-Yourself Spelling allows the user to add voice to the program.

Do-It-Yourself Spelling allows teachers, parents, and other individuals to create their own spelling programs. Following simple instructions, the person enters a series of 10-word lists into the computer program. The word lists might consist of a child's vocabulary assignment. It might consist of science words or musical terms or even the names of baseball players.

Do-It-Yourself Spelling comes with a list of 1950 words that every child should recognize and be able to spell by the time he or she graduates from 6th grade. The words are organized by grade. There are 50 first-grade words, 300 words for each of the second, third, and fourth grades, and 500 words for each of the fifth and sixth grades.

Do-It-Yourself Spelling is available for use on Atari microcomputers with a memory of at least 8K. The program retails for $19.95.

For additional information, contact: Laurie Hall, Program Design, Inc., 11 Idar Ct., Greenwich, CT 06830 203-661-8799.

Mountain Computer Announces RAMPLUS+ ™ For The Apple II Computer

Scotts Valley, CA — Mountain Computer has just released a new dual 16K RAM card for the Apple II®. Two banks of 16K selectable RAM expand the Apple to 80K of available memory. The second bank of 16K RAM is controlled by user-supplied software. Hardware and/or software selection of each 16K bank of RAM is controlled by the user. The card also provides its own refresh circuitry.

Card installation is simple, just install it in any I/O slot. When RAMPLUS+ is in Slot 0, it emulates an Apple Language Card®, and you can install multiple RAMPLUS+ cards into the same Apple II.

Status includes the area and bank of RAM selected, READ, READ or WRITE operation,

NOW TWO LOCATIONS

SAVE TIME • SAVE SHIPPING

ATARI Computers for people.™

800™ $699

410 Recorder	$59.00
810 Disc Drive	$444.00
822 Printer	$359.00
825 Printer	$629.00
830 Modem	$159.00
820 Printer	$269.00
850 Interface	$159.00
New DOS 2 System	$21.00
CX30 Paddle	$18.00
CX40 Joy Stick	$18.00
CX853 16K RAM	$89.00
Microtek 16K RAM	$75.00
Microtek 32K RAM	$159.00
One year extended warranty	**$50.00**

ATARI 400

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

Intec 48K Board....$249

ATARI SOFTWARE

CX404 Word Processor	$119.00
CX404 PILOT	$68.00
CX413 Microsoft Basic	$68.00
CX4101 Invitation To Programing I	$17.00
CX4102 Kingdom	$13.00
CX4103 Statistics	$17.00
CX4104 Mialing List	$17.00
CX4105 Blackjack	$13.00
CX4106 Invitation to Programing 2	$20.00
CX4107 Biorythm	$13.00
CX4108 Hangman	$13.00
CX4109 Graph It	$17.00
CX4110 Touch Typing	$20.00
CX4008 SPACE INVADERS	$32.00
CX4112 States & Capitals	$13.00
CX4114 European Countries & Capitals	$13.00
CX4115 Mortgage & Loan Analysis	$13.00
CX4116 Personal Fitness Program	$59.00
CX4117 Invitation To Programing 3	$20.00
CX4118-20 Conversational Languages (ea.)	$45.00
CX4121 Energy Czar	$13.00
CXL4001 Educational Master	$21.00
CX6001-17 Talk & Teach Series (ea.)	$23.00
CX8106 Bond Analysis	$20.00
CX8107 Stock Analysis	$20.00
CX8101 Stock Charting	$20.00
CXL4002 Basic Computing Language	$46.00
CXL4003 Assembler Editor	$46.00
CXL4004 Basketball	$24.00
CXL4005 Video Easel	$24.00
CXL4006 Super Breakout	$30.00
CXL4007 Music Composer	$45.00
CXL4009 Chess	$30.00
CXL4010 3-D Tic-Tac-Toe	$24.00
CLS4011 STAR RAIDERS	$39.00
CXL4012 MISSLE COMMAND	$32.00
CXL4013 ASTEROIDS	$32.00
CXL4015 TeleLink	$20.00
Visicalc	$149.00
Letter Perfect (Word Processor)	$109.00
Source	$89.00
CX481	$75.00
CX482	$109.00
CX483	$54.00
CX484	$319.00

PRINTERS

Centronics 739-1	$649.00
Diablo 630 Special	$1799.00
Epson	
MX70	$359.00
MX80	$469.00
MX80FT	Call
MX100	Call
NEC	
8023	$639.00
7730	Call
7720	Call
7710	Call
Okidata	
82A	$499.00
83A	$769.00
84	$1129.00
Citoh Starwriter	
25 CPS-P	$1329.00
45 CPS-P	$1699.00
Paper Tiger	
445G	$699.00
460G	$899.00
560G	$1129.00
Talley	
8024-7	$1399.00
8024-L	$1629.00
Xerox 820	
System I 5½"	$2450.00
System II 8"	$2950.00
CPM 5¼	$169.00
Word Processing	$429.00
Super Calc	$269.00

RAM SALE!

Atari	
Microtek 16K	$64.00
32K	$129.00
Intec 32K	$139.00
40K	$219.00
Ramcrom 128K	$519.00
Apple	
Microtek 16K	$94.00
32K	$129.00
Commodore	
64K upgrade	$389.00
Hewlett Packard	
16K upgrade	$249

computer mail order saves

Texas Instruments

PHM 3006 Home Financial Decisions	$26.00
PHM 3013 Personal Record Keeping	$43.00
PHD 5001 Mailing List	$60.00
PHD 5021 Checkbook Manager	$18.00
PHM 3008 Video Chess	$60.00
PHM 3010 Physical Fitness	$26.00
PHM 3009 Football	$26.00
PHM 3018 Video Games I	$26.00
PHM 3024 Indoor Soccer	$26.00
PHM 3025 Mind Challengers	$22.00
PHM 3031 The Attack	$35.00
PHM 3032 Blasto	$22.00
PHM 3033 Blackjack and Poker	$22.00
PHM 3034 Hustle	$22.00
PHM 3036 Zero Zap	$18.00
PHM 3037 Hangman	$18.00
PHM 3038 Connect Four	$18.00
PHM 3039 Yahtzee	$22.00
PHM 3017 Terminal Emulator I	$39.00
PHM 3026 Extended Basic	$88.00
PHM 3035 Terminal Emulator II	$45.00

TI-99/4A $379

PHC 004 TI-99/4 Home Computer	$399.00
PHP 1600 Telephone Coupler	$169.00
PHP 1700 RS-232 Accessories Interface	$169.00
PHP 1800 Disk Drive Controller	$239.00
PHP 1850 Disk Memory Drive	$389.00
PHP 2200 Memory Expansion (32K RAM)	$319.00
PHA 2100 R.F. Modulator	$43.00
PHP 1100 Wired Remote Controllers(Pair)	$31.00
32K Expansion	$329.00
PHP Printer Solid State	$319.00

HOW TO ORDER: Phone orders invited or send check or money order and receive free shipping in the continental United States. PA residents add 6% sales tax.

computer mail order west

800-648-3351

IN NEVADA, CALL (702) 588-5654

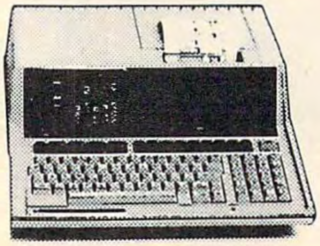
P.O. BOX 6689, STATE LINE, NEVADA 89449

TO SAVE YOU MORE!

COSTS • SAVE SALES TAX

you more money & time!

HEWLETT PACKARD



HP•85 $2479

NEW! HP•125	$2999.00
HP•83	$1699.00
HP•85 16K Memory Module	$249.00
5¼" Dual Master Disc Drive	$2129.00
Graphics Plotter (7225B)	$2079.00

NOW IN STOCK!

The new HP41CV Calculator

$259

Call for HP Software Prices & Information.
Call for Calculator prices.

CBM 8032

$1069

commodore

8032	$1069.00
4032	$969.00
4016	$769.00
8096	$1569.00
Super Pet	$1599.00
2031	$529.00
8050	$1299.00
4022	$599.00
C2N	$63.00
Word Pro 4 Plus	$299.00
Word Pro 3 Plus	$199.00

SOFTWARE

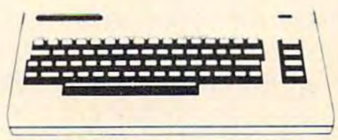
WordPro3 Plus	$299.00
WordPro4 Plus	$329.00
Commodore Tax Package	$399.00
Visicalc	$149.00
BPI General Ledger	$329.00
OZZ Information System	$289.00
Dow Jones Portfolio	$129.00
Pascal	$239.00
Legal Time Accounting	$449.00
Word Craft 80	$289.00
Power	$89.00
Socket-2-Me	$20.00
Jinsam	$Call
MAGIC	$ Call
The Manager	$209.00
Softrom	$129.00

Terminals

Televideo	
910	$579.00
912C	$699.00
920C	$749.00
950	$939.00
Call for computers	
Zenith 719	$749.00
Adds	$549.00

Monitors

Amdex 12" B&W	$149.00
12" Green	$169.00
13" Color	$349.00
Sanyo 12" B&W	$259.00
12" Green	$269.00
13" Color	$449.00
TI 10" Color	$349.00
Modems	
Novation Auto	$239.00
D Cat	$169.00
Cat	$159.00
HAyes	
Smart	$239.00
Pioneer Lazer Disk	$599.00
BSR X-10 Systems	
PK 500	$84.00
LM 501	$16.00
AM611	$17.00
AM286	$17.00



VIC 20 $259 COMPLETE

Vic-TV Modual	$19.00
Vic Cassette	$69.00
Vic 6 Pack Program	$44.00
VIC1530 Commodore Datassette	$69.00
VIC1540 Disk Drive	$499.00
VIC1515 VIC Graphic Printer	$399.00
VIC1210 3K Memory Expander	$32.00
VIC1110 8K Memory Expander	$53.00
VIC1011 RS232C Terminal Interface	$43.00
VIC1112 VIC IEEE-488 Interface	$86.00
VIC1211 VIC 20 Super Expander	$53.00
VIC1212 Programmers Aid Cartridge	$45.00
VIC1213 VICMON Machine Language Monitor	$45.00
VIC1901 VIC AVENGERS	$23.00
VIC1904 SUPERSLOT	$23.00
VIC1906 SUPER ALIEN	$19.00
VIC1907 SUPER LANDER	$23.00
VIC1908 DRAW POKER	$23.00
VIC1909 MIDNIGHT DRIVE	$23.00
VT106A Recreation Pack A	$44.00
VT107A Home Calculation Pack A	$44.00
VT164 Programmable Character/Gramegraphics	$12.00
VT232 VICTerm I Terminal Emulator	$9.00

Add 3% for VISA or MC. Equipment subject to price change and availability without notice.

computer mail order east

800-233-8950

IN PA. CALL (717) 327-9575

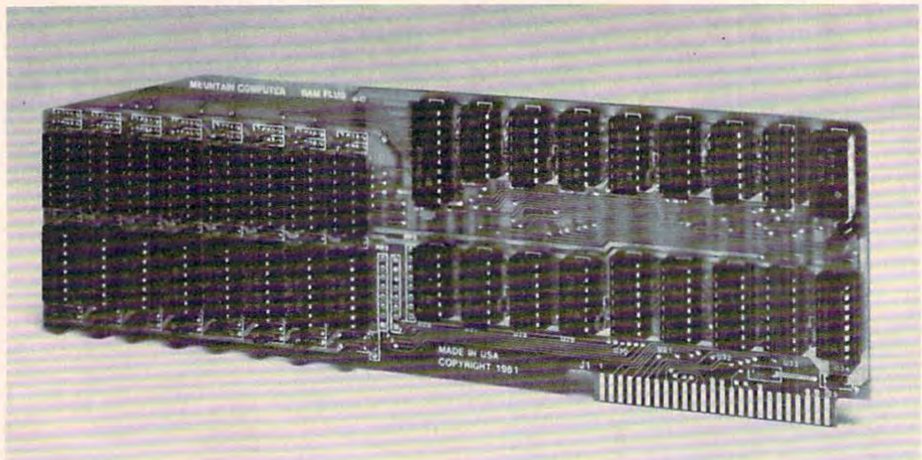
501 E. THIRD ST., WILLIAMSPORT, PA 17701

and ROM Enable or Disable information.

The card is supplied with 16K of installed RAM for $189.00 The additional 16K of plug-in RAM costs only $24.95. RAM diagnostics have been developed (on diskette) and are supplied with the product.

For additional information, contact:

Mountain Computer Inc.
300 El Pueblo Rd.
Scotts Valley, CA 95066
(408)438-6650



Calendar/Clock System For Small Computers

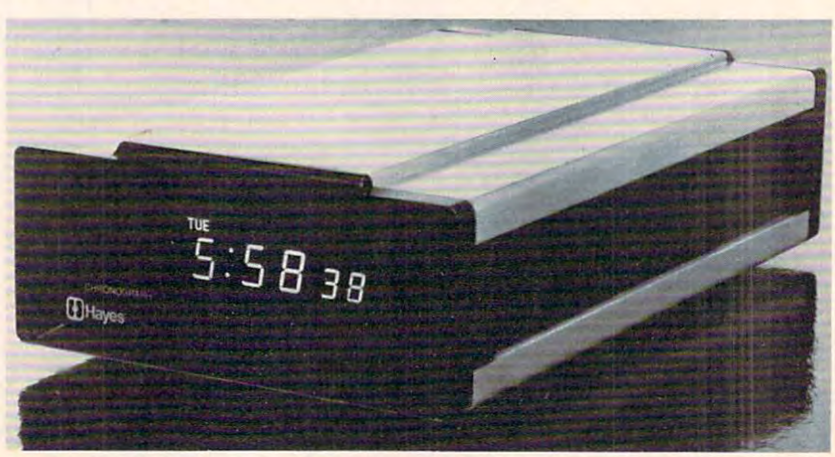
Norcross, GA — Hayes Microcomputer Products, Inc., announces the Hayes Stack™ Chronograph, an RS-232C compatible calendar/clock for small computers. The Chronograph is the latest in the Hayes Stack series of stackable microcomputer component systems.

Featuring quartz-crystal control, the Chronograph adds the dimension of precise timekeeping to computer systems. With the Chronograph and user-developed software, a computer can log programs and reports by day, date and time. Utilizing the computer alarm feature, the Chronograph can also provide a computer with information necessary to control lights, burglar alarms and sprinkler systems. To cut the cost of electronic mail, the user can combine the Chronograph with the Hayes Stack Smartmodem and a computer, then develop programs to batch messages during the day and send them at night when telephone rates are lowest.

The Chronograph is a stand-alone unit in an aluminum case with a large, easy-to-read display for time, date and weekday reporting. The display also features low battery, write-protect and alarm indicators.

The Chronograph reports the time in hours, minutes and seconds in 12- or 24-hour modes. The date is output in a year, month, day format with automatic leap year adjust, and the weekday is output as a single digit, 0-6.

Because it is powered independently, the Chronograph does not need to be reset when the computer is turned off. The Chronograph also features a battery backup to maintain time, date and weekday for up to a year when the power fails or is disconnected. For added protection, a write-protect switch on the rear panel of the Chronograph prevents accidental changing of the time and date.



ECLECTIC SYSTEMS CORPORATION

Authorized Commodore service center
Repair of the complete line of Commodore products
In a hurry? Check our modular exchange program

HARDWARE:	
CBM 8032 Computer, 80 Column	$1095
CBM 8050 Disk Drive	1340
CBM 4032 Computer, 40 Column	995
CBM 4040 Disk Drive	995
CBM 4022 Printer	649
CBM VIC 20 Computer	263
CBM VS100 Cassette	68
PET to IEEE Cable	33
IEEE to IEEE Cable	39
BASF Diskette, Box of 10	30

SOFTWARE:	
OZZ	$299
Wordcraft 80	299
Tax Preparation System	380
IRMA	380
Dow Jones Portfolio Management System	115
Personal Tax	55
Pascal	229
Assembler Development Package	77
Wordpro 4+	329

Order TOLL FREE 1+800-527-3135

10 AM to 4 PM CDT Monday through Friday

Texas residents call 1+214-661-1370

VISA, MASTER CHARGE, MONEY ORDERS, AND C.O.D. "Certified Check" accepted.
Units in stock shipped within 24 hours, F.O.B. Dallas, Texas.
All equipment shipped with manufacturer's warranty.

Residents of Texas, Louisiana, Oklahoma City and Tulsa, Oklahoma must add applicable taxes.

Eclectic shortly will be announcing products that are designed to work with CBM systems.

1. ROMIO: two RS232 ports—three parallel ports—26K EPROM memory-managed alternate character set, software controlled—EDOS (extended DOS).
2. Terminal program (options with ROMIO)
3. EPROM programmer
4. Front-end processor
5. Additional firmware to be announced

Be sure to write the address below for more information; dealer inquiries welcome.

P.O. Box 1166 • 16260 Midway Road • Addison, Texas 75001 • (214) 661-1370

The user controls the Chronograph through a command set of simple ASCII character strings. These commands allow the user to set, read and display calendar and clock data, control the computer alarm and select various options. Other features include 300 or 1200 baud operation and automatic baud rate, parity sense and word size detect.

The Chronograph system includes the Chronograph unit, power pack, 3 AA batteries and complete owner's manual. It is covered by Hayes two year limited warranty. The suggested retail price is $249.00. For further information contact Hayes Microcomputer Products, Inc., 5835 Peachtree Corners East, Norcross, GA 30092.

TYCOM Introduces Three Educational Packages

TYCOM Associates announces three new educational software packages for the Commodore PET/CBM computers, to compliment their existing educational software line. The programs are intended for drill and practice in conjunction with courses at the Junior High or High School level.

ALBEGRA *WORD PROBLEMS:* A CAI module intended to help teach algebra students to set up and solve word problems. A drill option supplies randomly chosen problems from several formats, with randomly generated values. The student is given the correct answer if a wrong answer is entered, and a score is given upon exiting the program.

SPANISH: Drill and practice programs intended as a noun and verb vocabulary building tool. User may choose passive or active mode of operation. Verb Conjugation drills are included.

GERMAN: Drill and practice programs intended as a vocabulary building tool. User may choose passive or active mode of operation. Noun drills include gender and plural forms.

The above programs are available on cassette tape only and run on all 40 or 80 column screen PET/CBM computers. Each sells for $19.95. A free list of all educational software offered by TYCOM Associates is available upon request.

Write to:

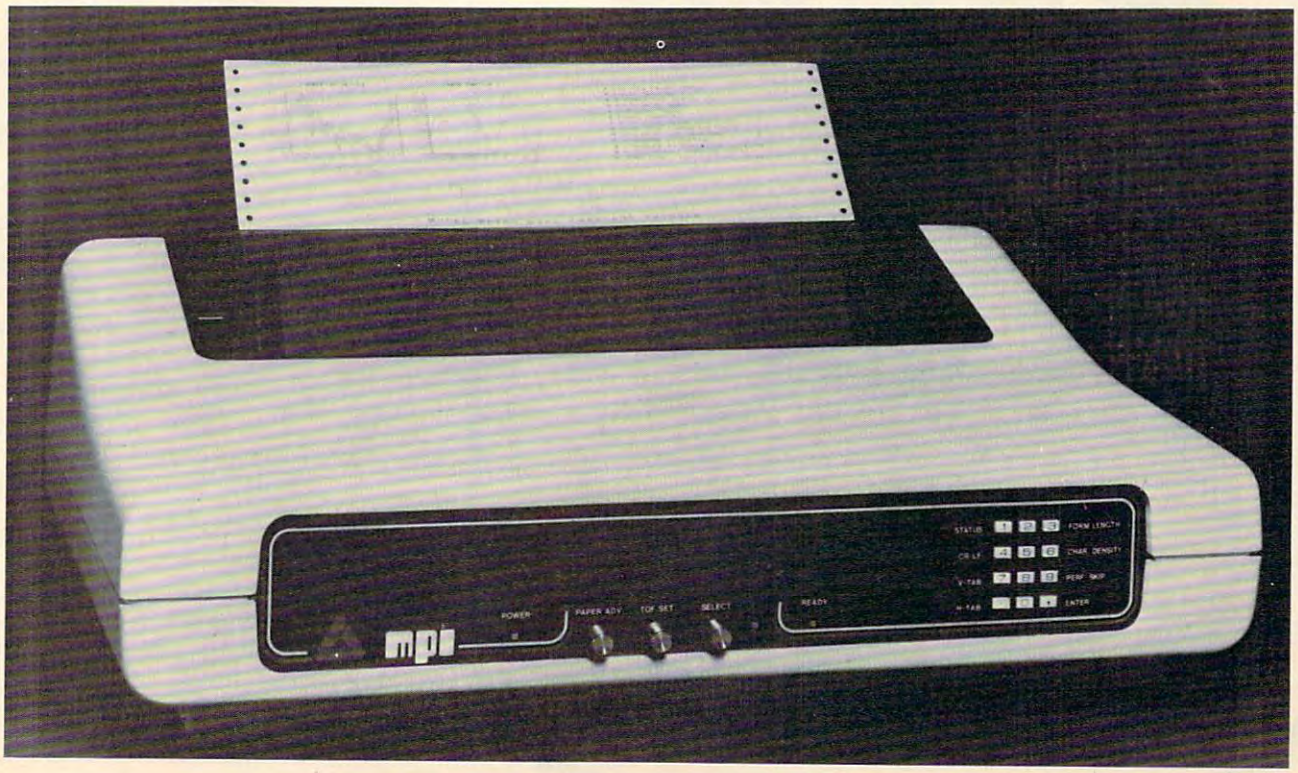
TYCOM Associates
68 Velma Avenue
Pittsfield, MA 01201

Model MP150 Wide Carriage Printer

The all new Model MP150 printer from MicroPeripherals, Inc. is the latest addition to their matrix printer line. It is the first of a series of wide carriage units designed specifically for mini and micro business systems.

The heavy duty printhead is rated for continuous duty and has an expected lifetime of over 100,000,000 characters. It forms characters bidirectionally in a logic seeking mode to optimize system thruput. Nine ballistically driven print wires form crisp, clear characters with true descenders and underlining capability. It can print a full 136 character line at 10 characters per inch or, by selecting either the 12 or 16.7 character per inch density, up to 226 columns may be printed. This allows full 136 column printouts to be condensed to fit on standard 8.5 inch wide paper. Double wide characters can be software selected in any of the character densities to give a total of six different CPI densities.

A 7x9 matrix font is used for high speed data printing while an 11x9 serif style matrix font is used for applications requiring a high quality correspondence printout. A standard 96 character USASCII set with four strap selectable foreign fonts are standard in each unit. Special fonts may be either down-line loaded into RAM memory, or permanently



located in ROM memory.

High resolution dot addressable graphics capability is included for those applications requiring plotting, printing of screen graphics, drawing of illustrations, and producing special characters or identification marks. Alphanumerics may be overprinted into the graphics area for labeling of graphs and illustrations.

Forms handling is implemented with a stepper motor drive tractor paper feed system which can be adjusted to accept forms ranging from 3 inches to 15 inches in width. Eight selectable forms lengths and a Skip-Over-Perforation feature provide for the precise paper handling required for business applications.

All printing and interface functions are placed under the direct control of a microprocessor array. The standard 1K buffer can be expanded to 8K for applications requiring additional character buffering. A Centronics type interface can accept parallel TTL level data at a transfer rate in excess of 1000 characters per second using either a Strobe/Ack or a Strobe/Busy handshake. An optional RS232C serial interface can be added and will accept data at any one of seven strapable baud rates up to 9600. Both X-ON/X-OFF and ETX/STX protocols are supported by the optional serial interface. The MP150 can also be interfaced to devices with an IEEE 488 Bus output through an optional IEEE-to-Centronics interface adapter card.

A long life mobius loop cartridge ribbon provides a minimum life of eight million characters.

An optional front console panel can be added to give greater flexibility in changing the print format parameters. It includes a non-volatile memory to store the format parameters for over three months without power.

The printer measures 23 inches wide by 16 inches deep by 7.5 inches high and allows for front, bottom or rear paper entry. It is designed to allow easy access to the electronic and mechanical components for simplicity of servicing.

The MP150 Printer, complete with graphics capability lists for $1095 with substantial discounts available for OEM quantities.

For additional information, contact:

Frank W. Irvin
Vice President, Marketing
MPI
4426 South Century Drive
Salt Lake City, UT 84107
(801)263-3081

Multi-purpose Interface For PET/CBM Computers

TEACHING TOOLS Microcomputer Services announces a new Multi-purpose Interface for PET/CBM Computers. This three-in-one interface provides the following:

1. Video monitor connector. Lets you show whatever is on the screen on a video monitor also. This is ideal for classrooms, and anywhere else a large display is needed. A high quality RF modulator (made by ATV Research) is also available, so you can use a TV in place of a video monitor. NOTE: The video adaptor is for PET/CBM computers with 9" screens only, *not* for 80 column CBMs or "Fat" 40 column PETs.

2. Sound adaptor with built in amplifier, speaker and volume control. Provides CB2 sound (the standard for PET/CBM computers). Takes its power from the PET – no batteries needed.

3. Audio tape recorder control.

commodore

Check Our New Lowest Prices!

EQUIPMENT:

	VIC-20 (Includes RF Modulator)	265
	CBM 4016 CPU (40 Col. Screen, 16K RAM)	790
	CBM 4032 CPU (40 Col. Screen, 32K RAM)	990
	CBM 8032 CPU (80 Col. Screen, 32K RAM)	1090
	CBM 8096 CPU (80 Col. Screen, 96K RAM)	1590
NEW	CBM Micro Mainframe (Super PET)	1690
NEW	CBM 2031 Single Disk Drive (170K per 5¼ Diskette)	570
NEW	CBM 4040 Dual Disk Drive (170K per 5¼ Diskette)	990
	CBM 8050 Dual Disk Drive (½ Meg per 5¼ Diskette)	1340
NEW	VIC Tractor Feed Printer (80 Col., 30 CPS)	325
	CBM 4022 Tractor Feed Printer	625
	CBM 8010 Phone Modem	225
	CBM C2N Cassette Deck (New Style)	65
	CBM CPU/IEEE Cable	35
	CBM IEEE/IEEE Cable	40
	Tally 8024-7 (7x7 Matrix) Printer	1590
	Tally 8024-9 (7x9 Matrix) Printer	1690

SOFTWARE:

Wordcraft 80 Wordprocessor	295
Wordpro 4+ Wordprocessor	325
OZZ Data Base System	295
Visicalc	175
Tax Preparation System	395
IRMA Information Retrieval	395
Dow Jones Portfolio	115
Pascal Development Package	225
Assembler Development Package	89
Personal Tax Calculator	59

PUBLICATIONS:

CBM User Guide	6.49
CBM Basic 4.0 Reference Manual	7.97
CBM Disk Manual	6.49
CBM Printer Manual	6.49
MOS Hardware Manual	5.75
MOS Programming Manual	5.75
The PET Revealed	17.49
Library of PET Subroutines	17.49
Commodore Software Encyclopedia	4.95

VIC SOFTWARE:

Six Pack A	- Includes Car Chase, Blue Meanies, Space Math, Slither/Super Slither, Biorhythm Capability	43.95
VIC Games	- Includes VIC Trap, Seawolf, Bounce Out	24.95
Household Finance	- Includes: Part I - Entering & Update; Part II - Summing & Displaying; Part III - Budgeting & Graphing; Part IV - Deductibles Analysis	14.95
Home Inventory	- Includes: Part I - Entering Inventory; Part II - Evaluating Inventory	14.95
Logic Games	- Includes Code Maker, Code Breaker	14.95
Rec/Ed I	- Includes Hangman, Hangmath	14.95
Rec/Ed II	- Includes Math Hurdler, Monster Maze	14.95
	- Inquire about other VIC software!	

COMING SOON:

More VIC Peripherals and Software
CBM 8250 Dual Disk Drive
CBM 8061/8062 8 inch Disk Drives
Data Acquisition and Control Devices

All items Insured
COD – UPS
Prepaid Orders Shipped Free
In stock items shipped within 48 hours

MASTERCHARGE OR VISA ADD 3%
GA. RESIDENTS ADD 4% SALES TAX

MART

P.O. Box 77286
Atlanta, Ga. 30357

404-981-5939
Call 9 AM-6:30 PM EST

Special Offer

8032 CPU	$1090.00
8096 CPU	$1590.00
Micro Main Frame	$1690.00

Lets you add recorded messages to your programs – puts starting and stopping a cassette recorder under program control.

The multi-purpose interface easily plugs into the back of the PET/CBM. If desired, it can be clamped from inside the PET, so that it cannot be removed. Complete instructions are included for connecting the interface, and for using CB2 sound and tape recorded messages in your own programs.

Prices: Multi-purpose interface, (requires cable with RCA phono plug at one end, connector for your monitor on the other end) $109.95. Multi-purpose interface and RF modulator (includes all cables and batteries for modulator) $149.95. Sample copy of instructions (credited toward later purchase) $1.00.

Please add $3.00 for shipping and handling.

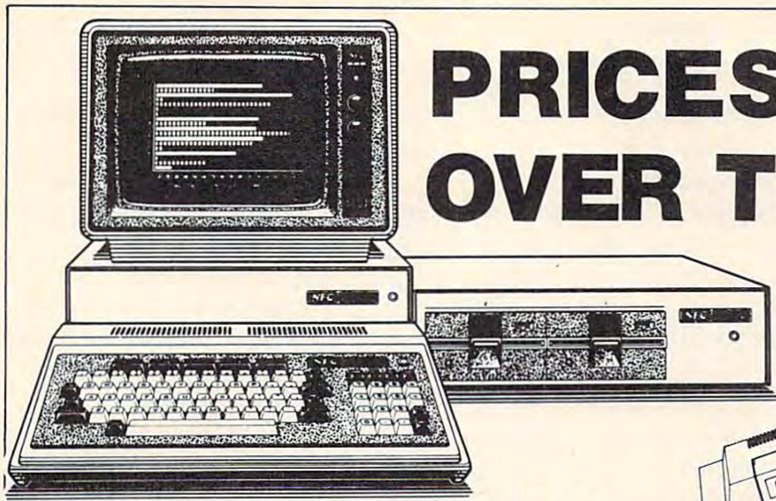
For prices in Canada, contact SES Computing, 465 King Street East, Suite 9, Toronto, Ontario M5A 1L6 (416-336-4242).

Axlon Personal Communication Terminal Announced

Sunnyvale, CA – November 19, 1981 – Axlon Incorporated of Sunnyvale, today announced the release of its portable personal communication terminal for the home and business market.

Called the Axlon HOTLINE™ Personal Communication Terminal, it can be used to transmit and retrieve information from data bases as varied as a personal telephone directory to the New York Stock Exchange.

The Axlon HOTLINE Personal Communication Terminal measures $1\frac{5}{8}$" x $3\frac{9}{16}$" x $6\frac{3}{4}$" and weighs less than 11 ounces providing a truly portable means to communicate with host computer



PRICES QUOTED OVER THE PHONE

Call Toll Free 800-343-5691

NEC The end of the compromise.

ATARI Computers for people.™

PRICE EXAMPLES:

ATARI 800 with 16K memory	**$749**
ATARI 400 with 16K memory	**$329**
ATARI 810 Disk Drive	**$439**
ATARI Software 10% OFF	
NEW: Pro Football 16K (Cassette)	**$7.95**
Galactic Chase (Disc)	**$29.95**
Ghost Hunters (Disc) **$34.95** (Cass.)	**$29.95**
ATARI 410 Recorder	**$65**

- Shipping Address
- Mass. res. 5% sales tax
- Include check, money order, MasterCharge or VISA number. We ship by best means possible freight collect

WARREN'S COMPUTER SYSTEMS
4 Elm St., Braintree, MA 02184
Mass. res. (617) 843-1133
Please Dial 800-343-5691
SEND FOR CATALOG

OLYMPIC SALES COMPANY

Authorized Dealer

apple computer

NEW APPLE FAMILY SYSTEM

LARGE INVENTORY LOWER-THAN-EVER PRICES

- ★ 48K Apple II +
- ★ Disk w/controller
- ★ RF Modulator
- ★ 7 Personal & game disks

Retail: $2495.00
Call for BEST Price!

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

WE ALSO CARRY SOFTWARE!

Personal Software	Innovative
Peach Tree Software	American
Microsoft	System Plus
Micropro	and more.

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

Just some of the software in stock NOW—
Magic Window, Raster Blaster, Ultima, Home $ Minder, Tax Preparer, Pool 1.5, Typing Tutor, Olympic Decathalon, ABM, Robot War, Castle Wolfenstein, Cranston Manor, Expediter, OO-Topos, Cyborg, Gorgon, Falcons, Wizardry, Personal Filing System, PFS: Report, Flight Simulator...

Texas Instruments Home Computer

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

New keyboard

	Retail	Your Cost
10" color monitor for 99/4	650.00	319.95
32K Exp. mem. module	399.95	314.95
Extended Basic, a MUST for 32K module	100.00	75.00
Speech synthesizer	150.00	129.95
Disk memory drive	500.00	397.95
RF modulator	69.50	49.50
Telephone coupler (modem)	225.00	189.95
Printer (solid state)	400.00	319.95

We carry a large inventory of software, & accessories

HEWLETT PACKARD

HP-11C
Advanced Programmable Scientific LCD *Retail:$130.00* *Your Cost:* **$114.95**

HP-12C
Advanced Programmable Financial LCD *Retail:$150.00* *Your Cost:* **$129.95**

	Retail	Your Cost
HP-125 New Microcomputer 64K CPU/Terminal/Keyboard	3750.00	3195.00
HP-85 Microcomputer—built-in printer/monitor	3250.00	2499.00
HP-83 Microcom—built-in mon	2250.00	1799.00
HP-2631B Printer, dot mtrx	3950.00	3199.00
HP82905A 80 col printer	945.00	755.00
HP-2601A Letter quality prter		3495.00
HP-82901M Dual master	2500.00	1999.00
Call us for the lowest prices on 7 disk drives		
Graphics plotter 7225	2450.00	2089.00
Personality module—a must for 7225	750.00	636.00
Graphics Tablet 9111A	2050.00	1699.00
HP-41CV	325.00	256.00
Card Reader for 41CV	215.00	171.00
Printer for 41CV	385.00	294.00
Quad Ram	95.00	84.95
Optical Wand for 41CV	125.00	99.95
HP-41C Calculator	250.00	188.95
Memory mod. for 41C		26.95
HP-97 Progrmble printer	750.00	579.95
HP-34C Progrmble scientific	150.00	117.95
HP-38C Progrmble business R/E	150.00	119.95
HP-32E Advanced scientific	55.00	48.95
HP-37E Business management	75.00	58.95

We carry a complete inventory of Libraries, accessories & supplies.

ATARI® COMPUTERS

Model		Retail	Your Cost
800	16K	1080.00	759.95
810	Disk Drive	600.00	469.95
Atari	VisiCalc	200.00	169.95
825	Impact Printer (Centronics 737)	1000.00	779.95

SUPER SPECIAL—ATARI 400 (16K)
Retail:$595.00 Your Cost:**$339.95**
Language Cart. not included—Opt'l at $54.95

Mattel 1670 Race Horse Computer
Ret. $125.00 Y/C: $24.95

ATARI®

Programmable Color TV Game Video Console: **$137.95**
Retail $225.00

MATTEL ELECTRONICS INTELLIVISION

Retail:$325.00 *Your Cost:* **$234.95**

VIDEO TAPES

Min order: 3, mixed OK

SONY	L500	2 hr	$11.89
SONY	L750	3 hr	14.69
RCA	VK250	6 hr	13.95

TEXAS INSTRUMENTS

	Retail	You Pay
TI-59	295.00	179.95
TI-3011 LCD Stu Slide/NEW		18.95
TI-35SP LCD Sci		22.50
TI-40 LCD Sci/NEW		28.95
Bus. Anal. I		19.95
Bus. Anal. II		44.95
Bus. Card		39.95
MBA		54.95
Invest. Analyst		48.95
TI-54 Adv. LCS Sci/NEW		39.95
TI-5511 NEW		44.95
TI-57 Prog. Sci		39.95
TI-58C 480 Step. Prog.		89.95
PC-100C Print/Plot		149.95
LCD-Programmer/NEW		59.95

OHIO SCIENTIFIC

Professional Computers

	Retail	Your Cost
C8PDF—48K	3495.00	3195.00

- Dual 8" Drives
- 64 col x 32 line/color
- .7 MIPS —*FAST!*
- Many more stndard featres

Fortran & Pascal available

Many other OSI products available —at discounted prices, of course!

Main Showroom & Offices:
216 South Oxford Avenue
Los Angeles, CA 90004
WE HONOR VISA and MASTERCHARGE
TELEX: 67 34 77

ORDER DESKS open 7 Days a Week!
7:00 AM to 7:00 PM Mon thru Sat
Sunday Noon to 5:00 PM
Order Desks: (213) 739-1130

TOLL-FREE (outside Calif.)	TOLL-FREE (within Calif.)
800-421-8045	800-252-2153

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

EPSON PRINTERS

	Retail:	Your Cost
MX80	645.00	499.00
Optional Graftrax Chip 80		95.00
MX80 FT	745.00	599.00
MX100	995.00	749.00

DIABLO 630

Letter Quality daisy wheel w/tractors
Retail: $3395.00
Y/C: $2595.00

PAPER TIGER PRINTERS

460	995.00	895.00
460G graphics	1094.00	969.95
560	1295.00	1099.00
560G graphics	1394.00	1195.00
445	795.00	695.00
445G	894.00	789.00

SANYO MONITORS High resolution

13" Color (new) high quality	550.00	419.95
12" Green phosphorous	360.00	259.95
12" Black & white	340.00	239.95
15" Black & white	370.00	259.95
9" Black & white (the best seller)	235.00	169.95

AMDEK(Leedex) High Quality Monitors

100	12" B/W, 12 MHz	179.00	139.95
100-G	12" Green, 12 MHz	199.00	174.95
300-G	12" Green. 18 MHz	249.00	199.95
Color I	13" Color, NTSC comp. input,	449.00	339.95

systems.

The terminal combines state-of-the-art features such as a built-in modem with more familiar features including an alphanumeric keyboard. The tactile keyboard consists of 43 functional keys arranged in typewriter sequence and provides a 64 upper case ASCII character set. The display is a 16 character fluorescent display which is tilted for

viewing. The terminal offers a 96 character display memory, which can be scrolled 16 characters at a time, and a 16 character display memory in the transmit mode.

The Axlon HOTLINE Personal Communication Terminal is designed for ease of use. The user dials the data base, waits for the connect signal, and then connects the modular headset jack to the terminal's modular telephone receptacle. The terminal also has a receptacle for connection of an ASCII serial printer should the user require hard copy of information retrieved.

For more information contact John Vurich or Robert Sultemeier, Axlon, Inc., 170 N. Wolfe Rd., Sunnyvale, CA 94086.

Medical Package For Apple III

Monument Computer Service has released a new software applications for the Medical Profession running on the Apple III Computer. The package, called the Medical Clinic, runs under the SOS operating system and is written in Business Basic. The package is designed for the multi-practitioner medical practice.

The package manages the physician's appointment schedules, does patient recall, prepares appointment logs, and provides for patient file management. The system also has a full accounts receivable system for managing daily transactions and payments, preparing monthly client bills, and reporting aged accounts receivable. The billing element also prepares standard AMA approved claim forms.

The system will handle a virtually unlimited patient base using either mini-floppy diskettes or the latest Apple hard disk. The system is designed to improve professional cash flow with such features as a superbill, individual bill preparation and cycle billing.

The package is available for $1,495.95 complete. A demonstration manual is also available for $50.00. Additional information is available from Monument Computer Service, Village Data Center, P.O. Box 603, Joshua Tree, CA. 92252. Technical questions and dealer inquiries should be directed to (714) 365-6668. Additional written information is available from the order center at (800) 854-0561 Ext. 802 (In California call 800-432-7257.)

Business Planner

Duosoft Corporation introduces BUSINESS PLANNER, a modeling package for entrepreneurs planning to start or expand a new business.

Designed to help develop viable business plans, the program groups labor, equipment and other costs into income-related projects.

Projects are combined into a model which generates graphical projections and estimated financial statements.

By exploring alternate models, you can evaluate critical timing and resource allocation decisions.

"What If" scenarios help you plan for the future and respond appropriately to changing demands.

You can compare actual results against the model in order to focus attention on problems that may harm future performance.

BUSINESS PLANNER is now available for the Apple II and III, and will soon be released for other machines.

For further information: Duosoft Corporation, Box 1827, Champaign, Illinois 61820.

Pascal Procedures For Business

Users Pascal Procedures Exchange Register (UPPER) has announced the release of "The Most Commonly Re-created Pascal Procedures for Business Application Programmers." This booklet contains UCSD p-System Pascal source code for user-friendly, bomb proof: screen input, access methods, printed report formatting, text formatting, data type conversions, and sample shell programs. These procedures can be incorporated into library units, segments, or used as in-line code.

Price: $19.95. Available from: Users Pascal Procedures Exchange Register, 1372 East 52nd Street, Chicago, IL 60615.

Financial Modeling Software Package

Osborne/McGraw-Hill has announced plans to distribute MicroFinesse, a financial modeling software package, with initial shipments to dealers commencing January, 1982. This move marks a major thrust by the McGraw-Hill Book Company into software distribution.

Lyco Computer Marketing & Consultants

We Specialize in Quality, Knowledge, Service, and Microcomputers

717-435-5197

ATARI®

A Warner Communications Company

© 1981, ATARI, INC

February Specials

32K MEMORY BOARDS	**$149.00**
16K MEMORY BOARDS	**$75.00**

for ATARI 400 & 800 with 1 year warranty!!

ATARI 800 16K	**$659.00**
ATARI 400 16K	**$329.75**

PROGRAMMING SOFTWARE

CX4101 INVITATION TO PROG	$ 17.00
CX4106 INVITATION TO PROG. 2	$ 19.00
CX4117 INVITATION TO PROG. 3	$ 19.00
CXL4003 ASSEMBLER EDITOR	$ 42.00
CXL4002 ATARI BASIC CART	$ 42.00
CXL4015 TELELINK	$ 19.00

EDUCATIONAL SOFTWARE

CXL4001 EDUCATIONAL SYSTEM	$ 19.00
CX6001 U.S. HISTORY	$ 22.00
CX6002 U.S. GOVERNMENT	$ 22.00
CX6003 SUPERVISORY SKILLS	$ 22.00
CX6004 WORLD HISTORY	$ 22.00
CX6005 BASIC SOCIOLOGY	$ 22.00
CX6006 COUNSELING PROCED	$ 22.00
CX6007 PRINCIPLES OF ACCT.	$ 22.00
CX6008 PHYSICS	$ 22.00
CX6009 GREAT CLASSICS	$ 22.00
CX6010 BUS. COMMUNCIATION	$ 22.00
CX6011 BASIC PSYCHOLOGY	$ 22.00
CX6012 EFFECTIVE WRITING	$ 22.00
CX6014 PRINCIPLES OF ECON	$ 22.00
CX6015 SPELLING	$ 22.00
CX6016 BASIC ELECTRICITY	$ 22.00
CX6017 BASIC ALGEBRA	$ 22.00
CX4108 HANGMAN	$ 12.00
CX4112 STATES & CAPITALS	$ 12.00
CX4114 EUROPE COUNT. & CAP.	$ 12.00
CX4121 ENERGY CZAR	$ 12.00
CX4123 SCRAM	$ 12.00
CX4102 KINGDOM	$ 12.00
CX4103 Statistics	$ 17.75
CX4104 Mailing List	$ 17.75
CX4105 Black Jack	$ 17.75
CX4107 Biorhythm	$ 17.75
CX4109 Graphit	$ 17.75
CX4110 Touch Typing	$ 19.75
CX4121 Energy Czar	$ 12.75

ATARI HARDWARE

410 Cassette Recorder	$57.75
810 Disk Drive	$444.75
850 Interface	$165.75
830 Modem	$145.75
825 Printer	$625.75

ENTERTAINMENT

CXL4004 BASKETBALL	$ 29.00
CXL4005 VIDEO EASEL	$ 29.00
CXL4006 SUPER BREAKOUT	$ 29.00
CXL4009 CHESS	$ 29.00
CXL4010 3-D TIC-TAC-TOE	$ 29.00
CXL4011 STAR RAIDERS	$ 30.00
CXL4012 MISSILE COMMAND	$ 30.00
CXL4013 ASTEROIDS	$ 30.00
CXL4008 Space Invader	$ 30.75
CXL4007 Music Composer	$ 42.75
CX4118 German	$ 42.75
CX4119 French	$ 42.75
CX4120 Spanish	$ 42.75
CX4125 Italian	$ 42.75

ACCESSORIES

CX30 PADDLES	$ 17.00
CX40 JOYSTICKS	$ 17.00
CX853 16K RAM	$ 85.00

WE CARRY MANY OTHER LINES OF MICROCOMPUTERS YOU CAN CALL FOR PRICES ON:

CPU's	PRINTERS
COMMODORE	TEC
ALTOS	DIABLO
TEXAS INSTRUMENT	NEC
ZENITH	EPSON
INTERTEC DATA	ANADEX
AND OTHERS	AND OTHERS

At Lyco Computers we offer our expert services to help customers make their first computer purchase, schools establish a computer program, or evaluate multiterminal systems.

to help evaluate your needs, or if you wish to make a purchase.

CALL US AT 717-435-5197

LYCO COMPUTERS
P.O. BOX 10
COGAN STATION, PA 17728

ORDERS SHIPPED SAME DAY

NO COD CHARGES

WE PAY FREIGHT ON PRE-PAID ORDERS

Written in Pascal, MicroFinesse runs on the Apple II microcomputer and will be distributed by Osborne/McGraw-Hill in the US and Canada only. The complete menu-driven package, with documentation, will sell for $495.00, available primarily through retail computer stores.

A financial modeling, forecasting and decision-making system, MicroFinesse was developed by the P-E Consulting Group, one of England's top management firms, with over 10 years experience in financial modeling.

According to Martin McNiff, Technical Group Manager at Osborne/McGraw-Hill, MicroFinesse is more than a spreadsheet package. "It offers planning capabilities seen before only on much larger systems, such as the ability to create investment and financial alternative models, as well as pro forma statements, sales productivity or profitability forecasts. Users can define target figures and use MicroFinesse to determine what must be done to meet those goals," says McNiff. He also points to the program's color graphics, model consolidations and report-generating features.

A significant commitment to after-sale support has also been announced by Osborne/McGraw-Hill, including a dedicated toll-free telephone "hot line" which will be in place at the time initial shipments commence.

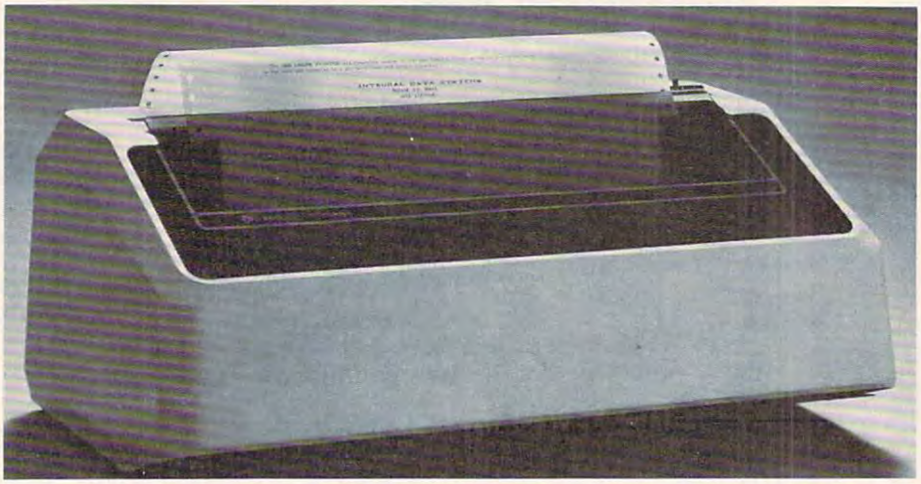
For more information, please contact: Chris Chambers, Sales and Marketing Director, Osborne/McGraw-Hill, (415) 548-2805.

Prism Printer From Integral Data Systems

A new color printer user-priced at $1,995 has been introduced by Integral Data Systems of Milford, New Hampshire. The Prism Printer™ is a low cost commercial color printer designed to compete with units costing three times as much.

The new 132-column dot matrix printer will produce eight colors using a four band ribbon which carries the process colors of cyan, magenta and yellow, as well as black.

"It's going to help define the expanding color graphics market," says Peter R. Eisenhauer, Integral Data Systems Vice President of Marketing. "There's a demand for color, primarily among business and professional users." Other immediate applications for the printer include the visual translation of scientific and medical data.



In addition to the color printer itself, Integral Data has plans to offer a number of collateral products which will facilitate the use of the Prism Printer in many key system environments. The first is expected to be an interface card for the Apple II and III which will have a graphics driver for the color printer resident on the card. Other products in the works include additional software drivers for Apple products as well as a color/graphics driver for the recently announced IBM Personal Computer.

The Prism Printer offers semi-automatic cut sheet feed, also a high-speed data mode. In the normal (correspondence) mode, the unit prints overlapping high density (24x9) matrix characters at up to 150 characters per second, bidirectionally. The high-speed data mode enables the user to select a standard density matrix and output large volumes of data at print speeds in excess of 200 cps.

Standard features include proportional spacing, enhanced (bold) text printing and standard print densities of 10, 12 or 16.7 characters per inch. The Prism Printer prints a full 132 characters per line at 10 pitch (characters per inch) with other pitches giving line lengths up to 220 columns on standard 15-inch-wide EDP paper.

Selectable features include automatic text justification, programmable horizontal and vertical tabbing, reverse paper feed, and "fine positioning" of characters of 1/120th of an inch. While the Prism Printer employs the standard ASCII upper- and lower-case 96-character set, up to four different 96-character sets can reside within the printer at the same time, for foreign language or custom character printing,

The Prism Printer is microprocessor controlled, with true "logic seeking" look-ahead capability and a high-speed slew for maximum output. It has a standard RS-232C serial interface as well as a Centronics-compatible

parallel interface. Serial transmission rates from 300 to 9,600 baud are switch selectable.

For more information on the Prism Printer and sample output, contact Integral Data Systems, Inc., Milford, New Hampshire 03055. (603) 673-9100; 800-258-1386; Telex: 953032.

New From Commodore

Valley Forge, PA, Nov. 12, 1981 – Commodore Business Machines has introduced its newest printer, the low-cost, high-speed CBM 8023P.

This latest addition to the growing line of CBM peripherals is a bidirectional, 136-column printer with both tractor and friction feed. The 8023P is dot-matrix, and prints 150 characters-per-second (CPS). It is available through Commodore dealers throughout the nation for $995.00.

The new CBM printer is designed to operate through software control, prints upper and lower case alphabetic characters, all graphic characters available with a Commodore computer, as well as user-defined characters.

The 8023P conforms to IEEE interface requirements and connects directly to a Commodore computer. It is designed to be used with the CBM floppy disk drives, and may be daisy-chained with other IEEE-488 devices.

Because the printer is an "intelligent" peripheral, it uses none of the computer's memory. In addition, the 8023P contains Random Access memory (RAM), which permits storage of formatting data.

* * * * * * *

A programmable character set and gamegraphics editor on cassette has been introduced by Commodore Business Machines, Inc., for users of its VIC 20™ personal computer.

Now available at authorized Commodore dealers throughout the nation for $14.95, the character set editor comes with a 16-page instruction manual and allows VIC users to create groups of 64, 128, or 192 programmable characters at a time and use them in BASIC programs. Each group of characters takes only one-half kilobyte (0.5K) of program space.

With the new character editor, Commodore VIC 20 users can create their own character set and easily modify letters, numbers, and graphics to include foreign language letters, mathematic and scientific symbols, or special "arcade" game graphics.

Commodore's new character set editor also allows VIC 20 users to save their newly-created character set on tape or disk for future use, and then easily insert the set in a BASIC program.

Along with the character set editor, also new from Commodore is the recently-introduced VIC 1515 low-cost dot-matrix printer. Available for $395, the VIC 1515 has a printing speed of 30 CPS, and prints any of the alphabetic, numeric, and graphic symbols common to the VIC.

From Krell Software Corp.

WAR OF THE SAMURAI is a game of combat and intrigue. Two to four players may compete in this original game that combines the strategic complexity of Go with the subtle dynamics of Chess. Detailed graphics.

Machines: Apple, PET, TRS-80, 16K, available on disk or cassette. $39.95

ALEXANDER THE GREAT is a vocabulary building game in a fantasy game context. Based on the Sword of Zedek, their best selling fantasy game. Alexander the Great introduces Aristotle as a mentor to the player. When called on, Aristotle poses vocabulary questions, and depending on the speed and accuracy of the player response, confers secret information. With Aristotle as an ally, the quest to overthrow Ra, The Master of Evil, assumes a new dimension of complexity. Players may select the level of vocabulary difficulty.

Machines: Apple, TRS-80, PET, 48K, available in two versions (K-8) & (9-College). $39.95.

ISAAC NEWTON challenges the players to assemble evidence and discern the underlying "Laws of Nature" that have produced this evidence. ISAAC NEWTON is an inductive game that allows players to intervene actively by proposing experiments to determine if new data conform to the "Laws of Nature" in question. Players may set the level of difficulty from simple to fiendishly complex.

In a classroom setting the instructor may elect to choose "Laws of Nature" in accordance with the complete instruction manual provided.

Machines: Apple, PET, TRS-80, Atari, 16K, available on disk or cassette. $24.95

FIG NEWTON – full graphics Newton. This version of Isaac Newton presents all data in graphic form. Because data is graphic rather than symbolic, this game is suitable for very young children. Players may, however, select difficulty levels challenging to the most skilled adults.

Machines: Apple, PET, TRS-80, Atari, 16K, available on disk or cassette. $24.95

ODYSSEY IN TIME adventure game adds a new dimension of excitement and complexity to TIME TRAVELER. Players must now compete with the powerful and treacherous adversary in their exacting quest for victory.

To succeed they must vanquish this adversary in combat

commodore spectacular

8032-32K 80 COL CRT	
REG. $1495	**$1095**
4032-32K 40 COL CRT	
REG. $1295	**$995**
4016-16K 40 COL CRT	
REG. $995	**$795**
64K ADD ON MEMORY	
REG. $500	**$395**
9000-134K SUPER PET	
REG. $1995	**$1795**
8050-DUAL DISK 950K	
REG. $1795	**$1345**
4040-DUAL DISK 343K	
REG. $1295	**$995**
2031-SINGLE DISK 170K	
REG. $695	**$555**
C2N-CASSETTE DRIVE	
REG. $95	**$69**
4022-80 COL PRINTER	
REG. $795	**$629**
8023P-136 COL PRINTER	
REG. $995	**$849**
8300P-40 CPS LETTER QUALITY	
REG. $2250	**$1995**
8024-MANNSMAN TALLEY	
REG. $1995	**$1595**
8024L - LETTER TALLEY	
REG. $2495	**$1995**
25 CPS-STARWRITER	
REG. $1895	**$1445**

SOFTWARE

	LIST	YOU PAY
OZZ — The Information Wizard	$395	$299
Wordcraft 80	$395	$299
IRMA — Info Retrieval & Mgmt. Aid	$495	$399
Dow Jones Portfolio Mgmt.	$149	$119
Pascal Development Pkg.	$295	$229
EBS — Receivables, Inventory	$750	$579
BPI — General Ledger	$395	$299
Word Pro 3 — 40 Column	$250	$179
Word Pro 4 — 80 Column	$375	$269
Word Pro 4 Plus	$450	$329

CBM-IEEE MODEM	
REG. $279	**$229**
VOICE SYNTHESIZER	
REG. $395	**$329**
VIC 20 ★★★★	
REG. $299	**$269**
VIC 1011 RS-232 INTER.	
REG. $49.95	**$39**
VIC 1112-IEEE INTER.	
REG. $99.95	**$79**
VIC 1540 - DISK 170K	
REG. $599	**$499**
VIC 1515 - 30 PS PRINTER	
REG. $395	**$349**
PET TO IEEE CABLE	
REG. $39.95	**$34**
IEEE TO IEEE CABLE	
REG. $49.95	**$39**

Philadelphia Computer Discount

155 E. LANCASTER, PA 19087

MAIL ORDER
800-345-1289
IN PA CALL (215) 687-8540

- PREPAID ORDERS SHIPPED FREE
- VISA & MASTERCARD ADD 3%
- COD - UPS
- PA RESIDENTS ADD 6% PA TAX

that rages across 24 time periods.

ODYSSEY IN TIME includes all the challenges of TIME TRAVELER plus 10 additional eras, including those of Alexander the Great, Emperor Asoka of India, Attila the Hun, Genghis Khan. Each game is unique and may be interrupted and saved at any point for later play.

Machines: Apple, PET, TRS-80, Atari, 32K, available on disk or cassette. $39.95

Krell Software Corp.
21 Millbrook Drive
Stony Brook, NY 11790

From Strategic Simulations

Napoleon's Campaigns is a corps-level game simulating the last campaigns of Napoleon: Leipzig and Waterloo. It is an advanced-level, board-assisted computer game.

Each campaign is displayed on a colorful 18x21 hex grid map in Hi-Res Graphics. The game employs a unique system requiring orders to be sent and received through dispatches. The reports received vary in degree of accuracy based on a variety of historical factors. The computer acts as corps commander for each corps, interpreting the orders the corps receives and often acting on its own initiative. These features simulate the frustration experienced by commanders of the Napoleonic Era.

The game includes one diskette, rule book, player aid card, 2 two-sided map boards and 100 counters for $59.95.

Southern Command is a battalion-level simulation for the Israeli counterattack to cross the Suez Canal during the October War of 1973 against Egypt.

The Sinai battleground is displayed in Hi-Res Graphics on a 28x39 hex grid map which can be viewed on one screen or on twelve screens, using scrolling. More than ten unit types including tanks, halftracks, BDM's, infantry and Egyptian SAM sites (to combat Israeli airstrikes) are used in the two player and each of the four computer-as-opponent scenarios.

Modern warfare is accurately reflected in the ability of units to reorganize after they have been attacked and in the "Delayed Move" feature, allowing units to ambush moving enemy units. Each side also has the ability to sight hidden enemy units.

Southern Command is available with diskette, rule book, map and player aid card for $39.95.

Both games require a 48K Apple II with Applesoft in ROM and one disk drive.

From Strategic Simulations, 465 Fairchild Dr., Suite 108, Mountain View, CA 94043.

From Automated Simulations

Automated Simulations, Inc. has released a new MIND TOY, Ricochet, an original abstract strategy game designed exclusively for the home computer.

Ricochet is a game of subtle strategy combined with fast action and arcade-style graphics. The game can be played against any of four different computer opponents, or against another human.

The player maneuvers blocks to set up a shot at his opponent's goal and to protect his goal from attack. Each player has two launchers he can fire. His shots ricochet off the blocks, earning him points each time a block is hit, plus he gets bonus points for hitting his opponent's goal.

Before he can claim victory, the player must win two out of three (or three out of five) games. A match victory also boosts his personal Ricochet Player Rating, which measures his mastery of the game against other players.

Ricochet is available on cassette for the Atari 400/800 (16K with BASIC ROM cartridge) and TRS-80 (16K, Level II), or on disk for the Atari 400/800 (32K), TRS-80 (TRSDOS 32K) and APPLE (48K with Applesoft in ROM). $19.95

From Automated Simulations, P.O. Box 4247, Mountain View, CA 94040.

From Synergistic Software

Odyssey: The Compleat Adventure, is now available for the Apple II computer in the Applesoft BASIC language. This adventure game is expanded into three separate but interlocking programs. The programs have colorful high resolution animation as well as sound effects. Many different paths to the goal exist that will not trap the player or force repetition. Being a role playing game, player action determines alignment, charisma, wisdom, experience, etc. These features affect the outcome of friendship and battle encounterd during play.

The object of this game is to save a realm from an evil ruler. Starting alone on a large island you seek out gold, soldiers to join you, and useful tools while gaining experience. If you are successful and clever you can not only walk but also ride, fly, and sail.

With dozens of high resolution pictures and animation effects different each time you play, each game is unique. Careful planning and strategy are necessary to successfully complete this adventure. Requires 48K Apple II or Apple II Plus. Available in Integer or Applesoft for $30.00 from Synergistic Software, 5221 120th Avenue SE, Bellevue, WA 98006. (206)226-3216