OSI

AARDVARK NOW MEANS BUSINESS!

OSI

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 capabilities. 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 (OS65D 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

NEW-NEW—NEW 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

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

ACCOUNTS RECEIVABLE — This program will handle up to 420 open accounts. It will age accounts, print invoices (including payment reminders) and give account totals. It can add automatic interest charges and warnings on late accounts, and can automatically provide and calculate volume discounts.

24K and OS65D required, dual disks recommended. Specify system.
Accounts Receivable. $99.95

* * * SPECIAL DEAL — NO LESS! * * *

A complete business package for OSI small systems — (C1, C2, C4 or C8). Includes MAXI-PROS, GENERAL LEDGER, INVENTORY, PAYROLL AND ACCOUNTS RECEIVABLE — ALL THE PROGRAMS THE SMALL BUSINESS MAN NEEDS. $299.95

P.S. We're so confident of the quality of these programs that the documentation contains the programmer's home phone number!

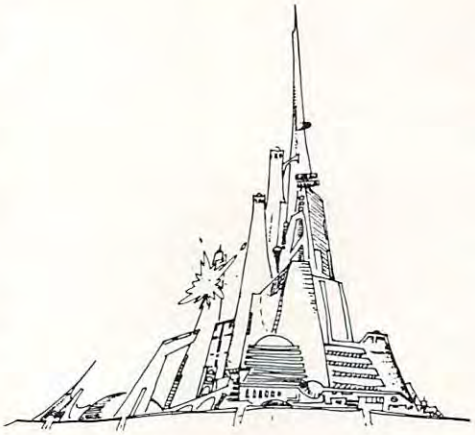
SUPERDISK II

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 (5 1/4") $34.95 (8").

AND FUN, TOO!



BOOKKEEPING THE EASY WAY — WITH BUSINESS I

Our business package 1 is a set of programs designed for the small businessman who does not have and does not need a full time accountant on his payroll.

This package is built around a **GENERAL LEDGER** program which records all transactions and which provides monthly, quarterly, annual, and year-to-date PROFIT AND LOSS statements. GENERAL LEDGER also provides for cash account balancing, provides a BALANCE SHEET and has modules for DEPRECIATION and LOAN ACCOUNT computation.
GENERAL LEDGER (and MODULES) $129.95.

PAYROLL is designed to interface with the GENERAL LEDGER. It will handle annual records on 30 employees with as many as 6 deductions per employee.
PAYROLL - $49.95.

INVENTORY is also designed to interface with the general ledger. This one will provide instant information on suppliers, initial cost and current value of your inventory. It also keeps track of the order points and date of last shipment.
INVENTORY - $59.95.

GAMES FOR ALL SYSTEMS

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

NEW — NEW — NEW

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.

NIGHT RIDER - You've seen similar games in the arcades. You see a winding twisting road ahead as you try to make time and stay on the road. NIGHT RIDER uses machine code to generate excellent high speed graphics - by the same author as MINOS.
NIGHT RIDER — $12.95 cassette only

THIEF - Another machine code goody for the C1P cassette only. You must use mobile cannon to protect the valuable jewels in the middle of the screen from increasingly nasty and trigger happy thiefs. Fast action and fun for one or two players. THIEF $13.95 on C1 cassette only!

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

Please specify system on all orders

This is only a partial listing of what we have to offer. We now offer over 100 programs, data sheets, ROMS, and boards for OSI systems. Our $1.00 catalog lists it all and contains free program listings and programming hints to boot.

VISA

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

master charge

OSI

OSI

page). The C1P video memory has 4 pages (D000-D3FF) and the C2P has 8 (D000-D7FF).

The cursor stays on the "home" line; its address varies by only 24 or 64 (dec.). Because we know it will always be on the same page, the cursor's location can be specified by only one byte (loc. 0200). I call this byte the cursor offset. The locationn of the cursor is found by adding the cursor offset to D300 (C1P) or D700 (C2P).

For those not familiar with machine language, I suggest you think of the A, X, and Y registers as variables. When a value is loaded into the X register, think x = value.

On Our Way At Last!

We start with the character to be displayed in the A register. This character may come from any one of several routines such as LOAD, LIST, or FTB. At BF2D, the A register is loaded into location 0202. The A, X, and Y registers are saved on the stack. At BF35, the contents of 0202 are put back into the A register. This seemingly meaningless back-and-forth shift is done because the X and Y registers must be transferred to the A register so that they can be pushed onto the stack. At BF38, the character is checked to see if it is a null (00). If so, the routine branches to BF6D where the Y, X, and A registers are pulled from the stack and restored. Then, at BF72, the VR returns to the routine that called it.

If the character is not a null, location 0206 is examined to see if it is greater than zero. If it is greater than zero, the contents of 0206 are used as a counter for the TV delay loop. This timing loop slows the VR to keep it from printing too fast for slower peripherals. If 0206 contains zero, the timing loop is bypassed.

At BF47, the character is checked to see if it is an LF(0A). If the character is other than an LF, it falls through to BF48. If the character is an LF, we go to BF76. For the time being, let's bypass BF76 and see what happens if the character is not an LF.

At BF48, the character is tested to see if it's a CR(0D). If the character is a CR, the routine falls through to BF4F. Again, let's defer exploration of this route and branch to BF55. This is the route all non-control characters travel.

Stalking The Non-Control Character

At BF55, the character is stored in location 0201. We JSR (jump to subroutine) to BFC2, where the contents of 0201 are printed D300 (C1P) or D700 (C2P) plus the cursor offset. The cursor offset is stored in 0200, which was initialized with the contents of FFE0 when the BREAK key was pressed at system start-up. The contents of FFE0 are 65 (C1P) or 40 (C2P). This means "home" position in the C1P is D365 and D740 in the C2P. After the character is printed, we RTS to BF5B and increment the cursor offset (loc. 0200).

At BF65, the current cursor offset is compared to the maximum cursor offset. If the end of the video line has been reached, an automatic CR/LF is done (JMP BF73). Otherwise, the routine JSRs to BFDE. At BFDE the character at D300 (C1P) or D700(C2P), plus the cursor offset, is stored in 0201. Remember that location 0200 was incremented at BF5B, so the character stored is the one in front of the current character. As far as I can tell, this character is never reused, except when a CR is done. At BFEF, the cursor character (5F) is printed and an RTS to BF6D is done.

At BF6D, the A, X, and Y registers are pulled from the stack and restored. Then BF72 does an RTS back to the routine that called the VR.

Let's go back and see the path a CR character follows. The CR starts at BF4F with a JSR to BFD5. BFD5 does a JSR to BFC2, which prints the character in 0201, the character "underneath" the cursor. This character is invariably a "space" (20). At BFD4 we RTS to BFD8, where the character in the "home" position is stored in 0201. At BFEF, the cursor character (5F) is printed at the "home" position. Now we RTS to BF52, which JMPs to BF6D. The A, X, and Y registers are restored and we RTS to the VR calling routine.

Spoor Of The Wily LF

Let's review the status of the TV display at the end of the CR. The cursor character has moved from its former position at the end of the home line to the home position. The character that formerly occupied this position is now stored in 0201.

With this in mind, we track the line feed character through the VR. The LF is usually done immediately after a CR. We left the LF at BF76, which JSRs to BFC2 and prints the contents of 0201 at the home position. This restores the first character of the line and erases the cursor. At BFD4, we RTS to BF79, where the cursor offset is ANDed with the hex number E0. This has the effect of rounding the offset to the start of the video line.

The rounded-off number is stored in 0202. Next, a scroll-one-byte routine is copied from BASIC ROM to RAM at 0207-020E. At BF8C, the X register is loaded with D3(C1P) or D7(C2P). The X register will be used later to determine whether or not the routine is scrolling the last page of video memory. Hex 20 is stored in the A register and the line width is put into the Y register. If the line width is greater than 20, which indicates a 2K memory, the A register is doubled (40). At BF99, the A register is used to set the 0207-E subroutine for a 20 (32 dec.) or a 40 (64 dec.) character line length. The Y register is zeroed in preparation for use as an offset counter for the 0207 subroutine.

At BF9E, the actual scroll is started with a JSR to 0207, which gets one byte from video memory and stores it in the next line above. The Y register is incremented and we RTS to BFA1 and check to see if the current page has been completely scrolled.

If the page is not done, we branch to BF9E to scroll another byte. When the page is done, the 0207 subroutine is set to scroll the next page. A check is made to see if the 0207 subroutine is set to the last page of video memory. If the sub is not set to the last page, we go back to BF9E to scroll another page. If we are on the last page, we scroll down to the home line, using the Y register and location 0202 to tell when to stop scrolling. At BFB6, the home line is cleared by storing "space" characters in its memory locations. We JSR to BFDE, which prints the cursor in the home position. Finally, we RTS to BF6D, pull the A, X, and Y registers from the stack and, at BF72, the VR returns to the routine that called it. Our journey is finally over, and I hope it has been an informative one.

Video Routine (BF2D)

Figure 1.

```
  JSR          - GOSUB
  RTS          - RETURN
BRANCH, JUMP   - GOTO
INCREMENT      - Add one
/0200/         - Contents of loc. 0200
AND            - Logical function
```

All numbers are in hexadecimal.

```
BF2D Put /A reg./ (char.) in 0202
BF30 Save A,X,Y registers on stack
BF35 Put /0202/ (char.) in A reg.
BF38 If char. is null, branch to BF6D.
BF3A Load Y reg. with /0206/ (TV delay)
BF3D If Y is zero, branch to BF47.
BF3F TV delay loop
BF47 If char. is a LF, branch to BF76.
BF4B If char is not CR, branch to BF55.
BF4F JSR to BFD5.

 BFD5 JSR BFC2

  BFC2 Load X with /0200/
       (cursor offset)
  BFC5 Load A with  /0201/
       (char to print)
  BFC8 Load Y with /FFE2/
       (video mem size)
  BFCB If Y is not zero, go to BFD1.
  BFCD Store A is D300+/X/ (C1P)
  BFD0 RTS
  BFD1 Store a in D700+/X/ (C2P)
  BFD4 RTS
 BFD8 Load A with /FFED/
      (cursor "home" offset)
 BFDB Put /A/in 0200 (cursor offset)
 BFDE Put /0200/ is X.
 BFE1 Put char at D300-/X/ in A.
 BFE4 Put /FFE2/ (video mem size) in Y.
 BFE7 If Y is equal to zero, (1K video
      mem.) goto BFEC.
 BFE9 Load A with char. at D700+/X/.
 BFEC Put A in 0201
      (temporary char. storage)
 BFEF Put cursor char. (5F) in A.
 BFF1 Branch always to BFC8.
 BFC8 Load Y with /FFE2/
      (video mem size)
 BFCB If Y is not zero, go to BFD1.
 BFCD Store A is D300+/X/ (C1P)
 BFD0 RTS
 BFD1 Store a in D700+/X/ (C2P)
 BFD4 RTS

BF52 JMP BF6D
BF55 Put char. in 0201
     (temporary char. storage)
BF58 JSR BFC2

 BFC2 (See BFC2 subroutine above)
 BFD4 RTS

BF5B Increment /0200/ (cursor offset).
BF5E Put /FFE1/ (chars/line-1) in A.
BF62 Add /FFE0/ (cursor "home" offset)
     to A.
BF65 If A is greater than /0200/
     (cursor offset) JMP BF73.
BF73 JSR BFD8

 BFD8 (See BFD8 subroutine above)
      Put char. in "home" position into
      0201 and print cursor in
      its place.
 BFD4 RTS

BF76 JSR BFC2

 BFC2 (See BFC2 above)/
      Print char. from 0201 at home
      position.
 BFD4 RTS

BF79 Put /FFE0/ (cursor "home" offset)
     in A.
BF7C AND A with number E0 and put
     result in 0202.
BF81 Transfer scroll subroutine from
     BFF3-A to RAM. (0207-B)
BF8C Load X with /BFFE/ (D3) or
      /BFFC/ (D7).
BF8F Put 20 (line length) in A.
BF91 Put /FFE1/ (chars.line-1) in Y.
BF94 If Y greater than 20, then /A/.
BF99 Use A to set line length in
     scroll subroutine (0207-E).
BF9C Zero Y register (byte counter).
BF9E JSR 0207

 0207 Load one byte from video memory
 020A Store byte one line above
      previous location
 020D Increment Y.
 020E RTS

BFA1 If page is not done, loop to BF9E.
BFA3 Increment high byte of video
     addresses in scroll subroutine.
BFA9 If high byte not equal to D3 (C1P)
     or D7 (C2P) then branch to BF9E.
BFB1 Scroll last page down to
     cursor line.
BFB6 Put "spaces" in home line
     (erase line).
BFC0 Branch always to BF6A.
BF6A JSR BFDE (See BFDE sub above.)
     print cursor at home position,
     store char. in 0201.
BF6D Pull A,X,Y from stack.
BF72 RTS (Return from calling routine.) ©
```

Odds And Ends

String Array Bug

J. Horemans

If you use string arrays for example A$(N), then be aware of OSI's program bug that can wipe out your program if the FRE function is called by the machine when leftover strings fill your RAM. Avoiding the problem is simple. Merely choose the DIMensions of the string array with the formula N=3*+3. For example DIMA$(44) is O.K., but DIMA$(45) can cause a program to crash. Also call the FRE function after doing a string array manipulation. Write yourself a program to find (and then make a permanent list of) the values that circumvent this problem. Here is a suitable example.

```
10 for I=1 to 20 (or any other values e.g. 21 to 40)
20 N=3*I+2
30?N
40 next I
```

You should get a list like:

2, 5, 8, 11,14, ... 44, 47, 50, ...

The values of N printed on the screen are the ones to use. Choose the one closest to the size array you need. It is also necessary to put a line like this after string array manipulations:

```
100 X=FRE(0)
```

For Those Inclined To Experiment

Here is a program to demonstrate the string array bug in OSI.

```
 5 A$="B"
10 FOR I=1 TO 100
20 B$(1)=B$(1)+A$
30 next
40 B$(1)=""
50 X=FRE(0)
60 A=A+1:?"DONE"A;X
70 GOTO 10
```

Run the program as is. Nothing happens for a few moments, then the screen starts to flicker. This is the external symptom of the string bug. Recover by pressing the break key and doing a warm start by pressing W. Now add line one:

```
1 DIMB$(2) (or use any number generated by 3*N+2)
```

The computer should now hum through this little program, telling you it is happily doing its job and, in the process, cleaning up the leftover string in line 50. Remember, it *is necessary to have a line like line 50* to call this function. Try running the program without line 50. If the DIM statement is o.k. the screen won't flicker, but my 8K machine never gets past DONE 1.

If you are making a graphic display, and the o.k. message keeps coming up to mess it up, here are two ways to defeat it.

1. In immediate mode (no line number):

```
POKE 3, 96
```

This will turn off the message altogether, but you have to press RETURN to get the cursor back to the left of the screen.

2. Just before your subroutines, put this line:

```
3990 FOR I=1 to 10000: NEXT:END
```

Your program will now wait for the time specified by counting from 1 to 10000 (or whatever number you put in) before flashing the o.k. message and ruining your display. If your subroutines aren't at the end or you haven't any yet, just use any convenient high number for the line.

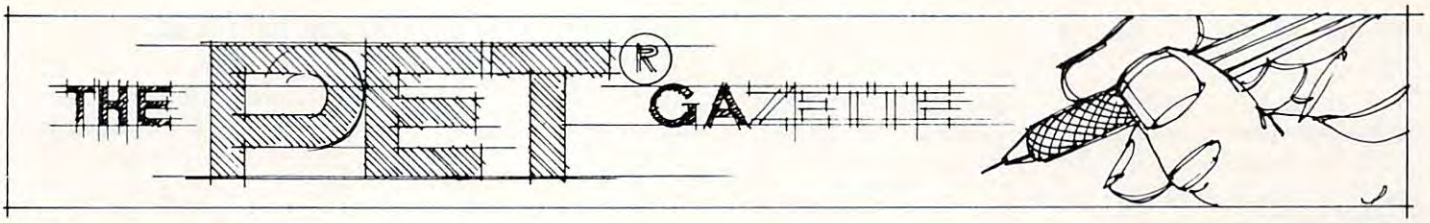
Those fast screen clears are great, but those pokes to locations 11 and 12 can be hard to keep in order if several ML calls are needed. Here is a fast full screen clear that does NOT use the ML call via USR(X). The method is described in *The First Book of OSI* by Williams and Dorner. I have adapted it to the C1P/Superboard screen. It does take more memory than some machine screen clears, but many of them need DATA lines to POKE in too, and that can cause confusion with your DATA statements.

```
5000 A=PEEK(129): B=PEEK(130):POKE129,0:
   POKE130,212
5010 S$="                ": rem 31 BLANK SPACES
5020 FOR I=1 to 32:S$=S$+" ":NEXT:POKE 129,A:
   POKE 130,B:RETURN
```

To call this screen clear insert a line like this:

```
100 GOSUB 5000
```

©



The Unwedge — Tape Append And Renumber

David A. Hook
Ontario, Canada

Lately in **COMPUTE!** there has been a great emphasis on the subject of appending. Both Harvey Herman (#8 pg. 96, #10 pg. 164) and Roy Busdiecker (#10 pg. 132) have presented their techniques.

These examples all require the use of a Basic program, which must be deleted later. Reference is made to the need to correctly arrange the line numbers. However, the programmer is required to find his own solution to this particular obstacle.

Wouldn't a utility combining Append with Renumber be helpful? The UNWEDGE attempts to satisfy these needs:

1. Written in machine language for speed of operation.

2. Self-relocating. The Basic program loads the code into high memory and protects itself from intrusion by Basic.

3. Works with Basic 2.0 (upgrade ROM) or Basic 4.0. The Basic loader automatically takes the proper code for the machine in use.

4. Reads and allows Appending of programs recorded on original (1.0) upgrade, and 4.0 machines.

5. The UNWEDGE attaches itself to the operating system (CHRGOT routine). Its commands may be used in the direct mode, with a minimum of user effort.

6. Compatible with the popular DOS support programs. While the Append function only works for tape append, the UNWEDGE doesn't interfere with the Wedge's function.

7. The same 'SYS' command is used to activate and to cancel the routines.

8. Consumes only 771 bytes of user memory.

Bill Seiler deserves the credit for nearly all the coding. 'PET Renumber 3.0' for original ROM (1.0) appeared in Pet User Notes (Vol I, #7 Nov./Dec. 1978). I have upgraded this for 2.0 and 4.0 ROM and provided for user input of line numbers.

Seiler was also responsible for the 'Append Wedge' for original ROM, which was printed in Commodore Canada's Transactor (Vol. 2, #3 July 1979).

The above PET User Notes issue included "M7171" by Jim Russo and Henry Chow. This is a high-monitor with merge capability for original ROM.

Features of both of these were cannibalized and converted for use with the recent ROM releases. Additions were made to provide messages and to allow loading of tape recorded on older machines.

To Use The Program (First Time):

1. Copy the Basic program listing. Watch for the "L", "H" and "-" in the DATA statements.

2. Save the Basic Program. If you try to run it first, an error may require a reset of the PET. You know what that means!

3. RUN the program. In about 20 seconds the screen will clear and give you the important information. Copy down the SYS number shown — you'll need it to activate/cancel UNWEDGE.

4. (OPTIONAL)

(If you're not familiar with the machine language monitor, the next set of "funny numbers" might confuse you. Don't let it, they are there to ease the next step.)

Prepare a tape to save the machine language code by itself. Move the cursor up to the line that begins:

.S "AP/REN...(etc.)

Hit the return key, and follow instructions (PRESS PLAY...). When the cursor returns, type

X RETURN

which gets you back to Basic.

You will note that the file name contains both the version (2 or 4) saved as well as the correct SYS number to use. This will be helpful when you come to use the machine code again.

This copy will load faster than the Basic program and doesn't require RUN after loading. SYS number will activate/cancel. The restriction is that it always sits in the same spot, and could overwrite something already in that spot in high-memory. Use the relocating Basic version if you're not sure.

For Tape #2 or Disk:

Change the "01" to "02" if you're saving to Tape #2. Insert "0:" or "1:" (Drive #) after the quote and ahead of the "AP/REN", and change

"01" to "08" if you're saving the object code on disk.

5. Activate the routine with the given SYS number. Since this is reversible, you may cancel UNWEDGE using the same number. (Don't try to save the machine language as in step 4 above, when the UNWEDGE is active).

To Use The Program (Subsequently):

Either: Load the Basic program, then Steps 3 and 5.
Or: Load the machine code, then Step 5.

To Renumber A Basic Program In Memory

The Append routine requires each segment (on tape) to have higher line numbers than the previous one (in memory). Thus, Renumber will often be called upon to prepare the various segments — the benefit of combining these utilities.

1. Clear the screen and move the cursor about 3/4 of the way down the screen.
2. Type the '‹' (less-than key) in the first column of a line.

Examples of options available:

‹R	'RETURN'
‹R 1000	'RETURN'
‹R 250,5	'RETURN'

The first example will renumber with starting line of 100 and step size of 10, which are the default settings.

The second one selects a starting line of 1000, and increment 10.

The final selection results in a starting line of 250, with a step size of 5.

The maximum step size allowed is 255. Be sure that the last line number (after Renumbering) will not exceed 63999.

3. When 'RETURN' is pressed, the upper part of the screen will show a variety of characters. (It is being used to store the line numbers). The message 'RENUMBERING' will be displayed.
4. When the cursor returns (1-10 seconds) the process is complete. All 'GOTO', 'THEN', 'GOSUB' and 'RUN' destinations have been updated. Any references to non-existent lines will be numbered '65535' (illegal line number) to flag the error. You'll have to correct these before proceeding.
5. Renumber will handle up to 500 lines, which should cover most programs.

To Append One Basic Program To Another

1. Place the program to be appended in Tape #1. Its first line number must be higher than the last line of the program in memory. Use the Renumber feature to prepare this segment beforehand.
2. Type the '‹' (less-than key) in the first column of a screen line.

Examples:

‹APPEND	'RETURN'
‹A	'RETURN'
‹A "PROG"	'RETURN'

The first two examples will Append the next program found on tape to the program in memory. Note that only the first letter "A" is necessary, though the whole word may be entered.

By specifying a file name (in quotes), the program 'PROG' will be Appended. Others found on the tape will be bypassed.

3. Only Basic programs may be Appended and Appended-to. If the in-memory program is not all Basic, the routine aborts giving 'NOT ALL BASIC PROGRAM' message.
4. If the specified program is not all Basic, a simple LOAD, not Append is executed. The same message is printed, but without the error condition. Be careful here, since the memory pointers may have been changed.
5. If the combined program will not fit into memory, an 'OUT OF MEMORY ERROR' will be shown. The Append will not take place.
6. The routine handles one of the differences in SAVE locations between Basic 1.0 and 2.0/4.0.

The other difference may be handles in one of two ways: Since Renumber (with subsequent SAVE) will often precede Append of a routine, the other discrepancy will be covered. As an extra precaution, simply reSAVE any segment before attempting to append it.
7. The message 'APPENDING' will be displayed as the proper file is found and loaded. When the cursor returns, the job is complete.
8. Repeat step 2 for further Appends, using Renumber as necessary.

I hope that this utility will find a home in your machine. Loading UNWEDGE into memory after power-up will keep it available for use.

As with most of my work, little would be possible without the published references by Jim Butterfield. Additionally, my thanks to Jim for the advice and suggestions on many subjects.

```
0 REM APPEND & RENUMBER BASIC PROGRAMS
1 REM        SELF-RELOCATING
2 REM FOR BASIC 2.0 AND BASIC 4.0
3 REM
4 REM DAVID A. HOOK, 58 STEEL STREET
5 REM    BARRIE, ONTARIO, CANADA
6 REM   L4M 2E9     (705) 726-8126
```

DR. DALEY'S SOFTWARE

Software with a difference.

THE MAIL LIST

A Proven Performer

Proven Performance with The Mail List is now in use by many satisfied customers.

It has been nearly two years since the first version of The Mail List was introduced. Only the BEST software will stand this test of time.

The Mail List has been specially designed to be used by the novice computer operator. All operations in the system are menu driven with built in user protection. This insures that you can have hassle free and error free operation.

Why waste time with other inferior mailing lists? Compare these features:

1. **User defined data structures.** You are the best judge of how your files should be organized. DR. DALEY'S mailing list package is unique in this feature. With The Mail List you can divide each record to suit your needs.
2. **User defined label format.** You can print from one to eight labels across and up to 10 lines per label.
3. **Interface to WordPro 3 or 4.**
4. **Fast and easy input and editing.**
5. **Easy to use 'wild card' sorting.** This will allow searches through the file using up to 3 fields.
6. **Multiple disk files.** Maximum capacity is 80 disks per file.

This powerful package comes on diskette with nearly 100 pages of documentation. It is packaged in an attractive binder.

When ordering please specify the version you are ordering. It is available for the following systems:

Version 4.4 $159.95
(Any computer with Commodore BASIC 4.0 and 32K memory with the 4040 (or upgraded 2040) disk drive.)

Version 4.8 $179.95
(Commodore 8032 with the 8050 disk drive.)

Call or write for details of our other software.

NOTE OUR NEW ADDRESS

DR. DALEY'S SOFTWARE
Water Street
Darby, MT 59829
Phone: (406) 821-3924
(Hours: 10 a.m. to 6 p.m. Mountain Time)

MasterCard VISA

KILO CORPORATION
313 N. First Street
Ann Arbor, MI 48103
Telephone: (313) 668-1566

DIRECT PAGE REGISTER FOR THE 6502

Create pseudo "zero-pages" at user specified memory locations * Access by poking from Basic or directly with machine code * For 6502's running at 1 MHz or less * Inquire for faster frequencies * EACH $155.00

CBM RUN TIME LINKING LOADER

Allows : * Linked loading of Basic programs anywhere in memory * Run time linking of Basic program segments * Saving Basic Machine Code loads during Basic execution - Creates M.C. virtual memory * EACH $27.95, with Screen Form, BOTH $45.95

SCREEN FORM CBM

Create invoice and other forms by using the cursor to layout CRT Input/Display fields * Avoid Input statement problems * Avoid programming Get commands * Formatted printing * Programmed controlled status line similar to WordPro's * EACH $27.95

CBM KEYBOARD EXPANSION FOR BASIC 4.0

16 keys allow: Inst/Del Lines, Erase Begin/End, Scroll Up/Down, Set Scroll Top/Bottom, Set Text/Graphics Mode * Plus numeric pad Return/Minus, Inst, Del, Slow/Stop-Scroll keys and joystick cursor control * $89.95

A CBM/PET SOFT ROM (8 KBYTES!)

8K CMOS RAM with write protect allows extra machine code or data storage * soft loading of ROM images to avoid ROM conflicts * EACH $190.00, with battery back-up $225.00

STD-BUS INTERFACE AND MOTHER BOARD

Expand with STD-BUS board level products. Used in industry, the variety of relatively low cost cards available make this bus a wise choice for development or laboratory systems. Interface $135.00, Mother Board and Rack - 6 slots $125.00

SEND FOR STD-BUS PRODUCTS CATALOG!

Specify all system hardware when ordering. CBM/PET and WORDPRO are trademarks.

```
 8 REM      ALL RIGHTS RESERVED
 9 REM PERMISSION TO COPY FOR NON-COMMERCIAL PURPOSES
10 QV=PEEK(50003):IF QV=1 OR QV=160 THEN12
11 PRINT"■WON'T WORK WITH YOUR ROM":END
12 EA=PEEK(52)+256*PEEK(53)-1:B=770:SA=EA-B:J%=SA/256:J=SA-256*J%
13 POKE EA-1,J%:POKE 53,J%:POKE 49,J%
14 POKE EA-2,J:POKE 52,J:POKE 48,J
15 J%=(SA+31)/256:J=SA+31-256*J%
16 POKE EA-3,J%:POKE EA-4,J
17 PRINT"■■■■■■■■WORKING...(20 SECONDS)
18 FOR I=SA TO EA-5:READ A$:A=VAL(A$)
19 IFASC(A$)=72 THEN J=I+VAL(MID$(A$,2)):A=J/256:GOTO21
20 IFASC(A$)=76 THEN J=I+VAL(MID$(A$,2)):A=J-256*INT(J/256)
21 POKE I,A:NEXT
22 IFQV=1THEN25
23 PRINT"■■■CORRECTING FOR BASIC 4.0..."
24 FORI=1TO28:READA,D:POKESA+A,D:NEXT
25 PRINT"■ACTIVATE OR CANCEL ■APPEND/RENUM"
26 A$=MID$(STR$(SA),2):QV$=MID$(STR$(4+2*(QV=1)),2)
27 PRINTTAB(18)"■--WITH: ■'SYS("A$")'
28 PRINT"■SAVE NOW (WITH MONITOR) ON TAPE#1:
29 A$="AP/REN"+QV$+"-"+A$
30 PRINT"■.S "CHR$(34)A$CHR$(34)",01";
31 X=SA/4096:GOSUB33:X=EA/4096:GOSUB33:PRINT:PRINT
32 SYS1024
33 PRINT",";:FORJ=1TO4:X%=X:X=(X-X%)*16:IFX%>9THENX%=X%+7
34 PRINTCHR$(X%+48);:NEXTJ:RETURN
35 DATA 173, L767, H766, 133, 52, 173
36 DATA L763, H762, 133, 53, 162, 3
37 DATA 181, 120, 72, 189, L748, H747
38 DATA 149, 120, 104, 157, L742, H741
39 DATA 202, 208, 241, 76, 121, 197
40 DATA 96, 201, 60, 208, 8, 72
41 DATA 165, 119, 201, 0, 240, 8
42 DATA 104, 201, 58, 176, 239, 76
43 DATA 125, 0, 32, 112, 0, 201
44 DATA 65, 240, 7, 201, 82, 208
45 DATA 237, 76, L282, H281, 162, 1
46 DATA 134, 212, 202, 134, 209, 134
47 DATA 157, 169, 2, 133, 219, 32
48 DATA 112, 0, 170, 240, 23, 201
49 DATA 34, 208, 246, 166, 119, 232
50 DATA 134, 218, 32, 112, 0, 170
51 DATA 240, 8, 201, 34, 240, 4
52 DATA 230, 209, 208, 242, 32, 86
53 DATA 246, 32, 18, 248, 32, 10
54 DATA 244, 165, 209, 240, 8, 32
55 DATA 148, 244, 208, 8, 76, 110
56 DATA 245, 32, 166, 245, 240, 248
57 DATA 224, 1, 208, 235, 165, 150
58 DATA 41, 16, 208, 127, 162, 24
59 DATA 173, 124, 2, 201, 4, 240
60 DATA 7, 162, 0, 32, L128, H127
61 DATA 240, 108, 32, L123, H122, 56
62 DATA 165, 42, 233, 2, 133, 42
63 DATA 165, 43, 233, 0, 133, 43
64 DATA 160, 0, 177, 42, 240, 24
65 DATA 32, L91, H90, 32, 110, 242
66 DATA 169, 13, 32, 210, 255, 169
67 DATA 63, 32, 210, 255, 162, 1
68 DATA 32, L83, H82, 76, 119, 195
69 DATA 32, L70, H69, 177, 42, 208
70 DATA 3, 32, L63, H62, 173, 125
71 DATA 2, 56, 237, 123, 2, 170
72 DATA 173, 126, 2, 237, 124, 2
73 DATA 168, 165, 42, 56, 233, 2
74 DATA 141, 123, 2, 165, 43, 233
75 DATA 0, 141, 124, 2, 138, 24
76 DATA 109, 123, 2, 141, 125, 2
77 DATA 152, 109, 124, 2, 141, 126
78 DATA 2, 197, 53, 144, 3, 76
79 DATA 85, 195, 32, 185, 243, 76
80 DATA 221, 243, 32, L2, H1, 230
81 DATA 42, 208, 2, 230, 43, 96
82 DATA 189, L11, H10, 240, 6, 32
83 DATA 210, 255, 232, 208, 245, 96
84 DATA 13, 78, 79, 84, 32, 65
85 DATA 76, 76, 32, 66, 65, 83
86 DATA 73, 67, 32, 80, 82, 79
87 DATA 71, 82, 65, 77, 32, 0
88 DATA 13, 65, 80, 80, 69, 78
89 DATA 68, 73, 78, 71, 32, 0
90 DATA 13, 82, 69, 78, 85, 77
91 DATA 66, 69, 82, 73, 78, 71
92 DATA 13, 0, 32, 112, 0, 240
93 DATA 33, 176, 249, 32, 115, 200
94 DATA 72, 166, 17, 164, 18, 134
95 DATA 62, 132, 63, 104, 240, 24
96 DATA 32, 112, 0, 240, 19, 176
97 DATA 249, 32, 115, 200, 166, 17
```

```
98 DATA 134, 66, 208, 12, 169, 100
99 DATA 133, 62, 169, 0, 133, 63
100 DATA 169, 10, 133, 66, 162, 36
101 DATA 32,L-115,H-116,169,254, 133
102 DATA 33, 169, 127, 133, 34, 165
103 DATA 40, 133, 31, 165, 41, 133
104 DATA 32, 32, L292, H291, 160, 3
105 DATA 177, 31, 145, 33, 185, 92
106 DATA 0, 145, 31, 136, 192, 1
107 DATA 208, 242, 177, 31, 240, 16
108 DATA 32, L301, H300, 170, 136, 177
109 DATA 31, 133, 31, 134, 32, 32
110 DATA L267,H266,208,220,169,255
111 DATA 200, 145, 33, 200, 145, 33
112 DATA 165, 40, 133, 119, 165, 41
113 DATA 133, 120, 208, 3, 32, L277
114 DATA H276, 32, L274, H273, 208, 3
115 DATA 76, 57, 196, 32, L266, H265
116 DATA 32, L263, H262, 32, L260, H25
117 DATA 170, 240, 233, 162, 4, 221
118 DATA L262, H261, 240, 5, 202, 208
119 DATA 248, 240, 238, 165, 119, 72
120 DATA 165, 120, 72, 32, 112, 0
121 DATA 176, 230, 32, 115, 200, 32
122 DATA L51, H50, 104, 133, 120, 104
123 DATA 133, 119, 160, 0, 162, 0
124 DATA 189, 1, 1, 240, 15, 72
125 DATA 32, 112, 0, 144, 3, 32
126 DATA L82, H81, 104, 145, 119, 232
127 DATA 208, 236, 32, 112, 0, 176
128 DATA 8, 32, L102, H101, 32, 118
129 DATA 0, 144, 248, 201, 44, 240
130 DATA 192, 208, 175, 32, L134, H133
131 DATA 169, 0, 133, 33, 169, 128
132 DATA 133, 34, 160, 1, 177, 33
133 DATA 197, 18, 240, 21, 201, 255
134 DATA 208, 24, 133, 95, 133, 94
135 DATA 165, 94, 133, 96, 162, 144
136 DATA 56, 32, 85, 219, 76, 233
137 DATA 220, 136, 177, 33, 197, 17
138 DATA 240, 236, 32, L96, H95, 32
139 DATA L116,H115,208,212,32,L62
140 DATA H61, 160, 0, 177, 31, 200
141 DATA 145, 31, 32, L90, H89, 208
142 DATA 8, 230, 42, 208, 2, 230
143 DATA 43, 136, 96, 164, 31, 208
144 DATA 2, 198, 32, 198, 31, 76
145 DATA L-29,H-30,32,L28,H27,160
146 DATA 1, 177, 33, 136, 145, 33
147 DATA 32, L56, H55, 240, 5, 32
148 DATA L65, H64, 208, 239, 164, 42
149 DATA 208, 2, 198, 43, 198, 42
```



```
150 DATA 96, 165, 42, 133, 31, 165
151 DATA 43, 133, 32, 165, 119, 133
152 DATA 33, 165, 120, 133, 34, 96
153 DATA 165, 62, 133, 94, 165, 63
154 DATA 133, 95, 96, 165, 94, 24
155 DATA 101, 66, 133, 94, 144, 2
156 DATA 230, 95, 96, 165, 31, 197
157 DATA 33, 208, 4, 165, 32, 197
158 DATA 34, 96, 32, L2, H1, 230
159 DATA 33, 208, 2, 230, 34, 96
160 DATA 160, 0, 230, 119, 208, 2
161 DATA 230, 120, 177, 119, 96, 137
162 DATA 138, 141, 167, 76
```

```
163 DATA 28,240,29,181,107,149,110,87,113,73,120,211,125,173
164 DATA128,229,184,162,202,237,203,179,264,205,265,179,267,248
165 DATA270,28,271,244,352,246,353,184,374,246,375,184,481,173
166 DATA482,180,519,246,520,184,608,127,609,205,611,147,612,207
READY.
```

STP-488 A Smart Terminal Program For An IEEE-488 Modem

Earl Wuchter
N. Catasauqua, PA

When the time comes that you decide to hook up to a commercial sharing network, or take advantage of some of the hundreds of free systems in operation across the country, you will have to make a hardware/software selection. The options facing you are as confusing as they are numerous. The program given here may help you decide.

The most straightforward way to go is with an IEEE modem, either the Commodore 8010 or the STAR 488. My personal choice was the STAR, but I think that the 8010 is identical. The STAR 488 has all the features a modem should have. It has both originate and answer modes and a half/full duplex switch. There is also a test (analog echo) mode. In addition, there are LEDs to indicate carrier detect, xmit, recv, and test.

You must select a terminal program that is compatible with the hardware. The PET does not use true ASCII, so the program must do some conversions if you want to send or receive lower case. Some IEEE to RS-232 interface devices do this conversion for you. STP-488 will probably not work with that type of interface.

It is possible to communicate with an IEEE modem in BASIC, but if you try to write a terminal program, you will most likely be disappointed with the results. The simplest BASIC "dumb" terminal program will be hard

```
1                       ; SMART TERMINAL FOR PET AND IEEE MODEM
2                       ;
3                       ; EARL WUCHTER
4                       ; 1610 WASHINGTON ST.
5                       ; CATASAUQUA, PA  18032
6                       ; JUNE 81
7                       ;
8                       ;-- ROM ROUTINES USED (VER 3.0 & 4.0)
9                       ; $FFC6  SET INPUT DEVICE. X=FILE NO.
10                      ; $FFC9  SET OUTPUT DEVICE. X=FILE NO.
11                      ; $FFCC  RESTORE DEFAULT I/O DEVICES.
12                      ; $FFD2  PRINT (A)  ASCII
13                      ; $FFE4  GET ONE FROM KYBD BUFF
14                      ; $F2E2  CLOSE FILE A (4.0)
15                      ; $F2AE  CLOSE FILE A (3.0)
16                      ; $BB1D  PRINT A STRING (4.0)
17                      ; $CA1C  PRINT A STRING (3.0)
18                      ;-- SYSTEM VARIABLES (VER 3.0 & 4.0)
19                      ; $35    PTR (HI) TO TOP OF MEMORY
20                      ; $96    ST (I/O STATUS)
21                      ; $9B    STOP AND RVS FLAG
22                      ; $9E    KEYBOARD BUFFER CHAR COUNT
23                      ; $A7    ENABLE CURSOR FLASH
24                      ; $C4    ADDR OF CURRENT SCREEN LINE
25                      ; $C6    CURSOR POSITION ON LINE
26                      ; $CD    QUOTE FLAG
27                      ; $D1    FILENAME LENGTH FOR OPEN FILE
28                      ; $D2    FILE NO. FOR OPEN
29                      ; $D3    SECONDARY ADDR FOR OPEN
30                      ; $D4    DEVICE NO. FOR OPEN
31                      ;-- ZERO PAGE USED FOR PROGRAM WORK SPACE
32      00 11           BPT     = $11    BUFFER POINTER
33      00 40           CTRL    = $40    CTRL FLAG
34      00 41           CCNT    = $41    CHARACTER COUNT
35      00 42           LCNT    = $42    LINE COUNT
36      00 43           ASAVE   = $43    RAW CHARACTER
37      00 44           VER     = $44    ROM VERSION INDICATOR
38      00 45           MODE    = $45    SAVES ORIGINAL UC/LC STATUS
39                      ;-- CBM/ASCII STUFF
40      00 12           RVS     = $12    RVS USED IN PLACE OF CTRL KEY
41      00 92           OFF     = $92
42      00 0D           CR      = $0D
43                      ;
44                      ; NOTE: ASSEMBLER QUIRK:
45                      ; > RETURNS LOW BYTE OF ARGUMENT
46                      ; < RETURNS HIGH BYTE
47                      ;
48                              *=   $0400
49  0400 00 0B 04               BYT  $00,$0B,$04   BASIC LINE 1 SYS1037
50  0403 01 00 9E               BYT  $01,$00,$9E,'1037',$00
51  0406 31 30 33
52  0409 37 00
53  040B 00 00                  BYT  $00,$00
54                      ;
55  040D AD 4C E8               LDA  59468   SAVE ORIGINAL MODE
56  0410 85 45                  STA  MODE
57  0412 A9 0E                  LDA  #14     SELECT LOWER CASE MODE
58  0414 8D 4C E8               STA  59468   POKE 59468,14
59  0417 A9 05                  LDA  #05     SET UP TO OPEN A FILE
60  0419 85 D2                  STA  $D2     FILENO=5
61  041B 85 D4                  STA  $D4     DEVICE=5 (MODEM)
62  041D A9 00                  LDA  #00
63  041F 85 D1                  STA  $D1     NO FILE NAME
64  0421 85 D3                  STA  $D3     SECONDARY ADDR
65  0423 85 9E                  STA  $9E     CLEAR KYBD BUFFER COUNT
66  0425 20 61 05               JSR  RESET   SET BUFFER POINTER
67  0428 91 11                  STA  (BPT),Y INSERT INITIAL EOF
68  042A AD E5 FF               LDA  $FFE5   TEST FOR 3.0 OR 4.0 ROM
69  042D 29 04                  AND  #04
70  042F 85 44                  STA  VER     0=3.0  4=4.0
```

80 × 25

On The Built-in Display

PET/CBM™ 2000/3000/4000 Series

$275.00 *

Select either . . . . . **80** x **25** or **40** x **25** . . . . . display format

From the keyboard or program

Features the same memory map and subroutines from the Basic ROMs [specify which version of Basic when ordering] * *, for standard programming. You also gain the use of **1023** extra memory locations in the 40 column mode, or **47** extra locations in the 80 column mode. [These locations are available only to machine language programs, or peek & poke from Basic].

Displays the full, original character set, including graphics characters in either mode.

All utility software, firmware, like Toolkit™, Dos Support [Wedge], Extra-mon, etc., is compatible in both modes of operation.

The complete enhancement consists of: 1 dual 24-pin socket [one socket for the 40 column screen editor, and one for the 80 column screen editor], and a circuit board that replaces the existing screen RAM. Each circuit board is registered to the original owner. There is also an 80 column reference ROM that plugs in one of the expansion sockets [specify the address when ordering]. An option board is available [$25.00] that allows the ROM to be used with any other 2K ROM, in any of the expansion sockets.

Available from your local dealer or:

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

* Plus appropriate installation charges. This requires some circuit modification. [available from the factory for $75.00 plus shipping]

* * If power-on message = ### COMMODORE BASIC ### you have 3.0 Basic. [Available only for Basic 3.0 & Basic 4.0 at the present].
PET™ & CBM are trademarks of Commodore Business Machines.
We will ship via Master Charge, VISA, C.O.D., or pre-paid.
Toolkit™ is a trademark of Palo Alto IC's, Inc.
Installation may void your Commodore 90 day warr.
The Execom™ board is guaranteed for 1 year.

pressed to keep up with a steady stream of incoming characters without losing some. It may have to ignore the keyboard as long as characters are coming in, which means you can't interrupt the other system. BASIC is certainly not fast enough to do the communication and also store text in memory.

On the other hand, a machine language program can be made as elaborate as you like. The biggest problem with a machine language program is writing it. PET I/O routines are very good, but not easy to trace. STP-488 does the equivalent of BASIC file handling commands by calling the appropriate routines in ROM. This technique was described by Raymond Diedrichs in the April 81 issue of **COMPUTE!**.

STP-488 begins with the familiar SYS 1037. If you are not up on this trick, there are two things to know: You may not change the BASIC line, and you cannot copy the program without using the (machine language) monitor. You can list it, but you won't see much.

The program was designed to be as powerful and easy to use as possible without becoming too large to be keyed in manually. In order to achieve this goal, some frills such as a help option had to be omitted.

I have tested the program on a variety of PETs. It works with Upgrade and 4.0 ROM, with graphic and business keyboards, in 40 and 80 column machines. The function of the shift keys varies slightly depending on the type of keyboard used.

Program Features

1. STP-488 Loads and runs like BASIC.
2. It works with Upgrade and 4.0 ROMS.
3. Monitor style operation.
4. Upper/lower case sent and received.
5. CAPS LOCK (modified).
6. CTRL characters may be sent.
7. Automatic storage of received text.

```
 71  0431 F0 06        BEQ  INIT2
 72  0433 20 63 F5     JSR  $F563    4.0 OPEN 5,5,0
 73  0436 4C 3C 04     JMP  QUERY
 74  0439 20 24 F5 INIT2 JSR $F524   3.0 OPEN
 75                    ;
 76  043C A9 14 QUERY  LDA  #>PROMPT
 77  043E A0 06        LDY  #<PROMPT
 78  0440 20 09 05     JSR  PRMSG    PRINT PROMPT LINE
 79  0443 20 E4 FF GETANS JSR $FFE4  GET ONE CHAR
 80  0446 29 7F        AND  #$7F     ACCEPT UC OR LC
 81  0448 C9 41        CMP  #'A'     WAIT FOR AN
 82  044A 90 F7        BCC  GETANS   - ALPHA CHAR
 83  044C A2 06        LDX  #06      2*( N.COMMANDS-1)
 84  044E DD 15 06 CHECK CMP CMDS,X  CMDS EMBEDDED IN PROMPT STRING
 85  0451 F0 07        BEQ  OKCMD    FOUND A MATCH
 86  0453 CA           DEX           TO SLASH DELIMITER
 87  0454 CA           DEX           TO NEXT CMD CHAR
 88  0455 10 F7        BPL  CHECK    TRY AGAIN
 89  0457 4C 43 04     JMP  GETANS   WAIT FOR A VALID REPLY
 90  045A A9 04 OKCMD  LDA  #<(QUERY-1)  PUSH ADDR
 91  045C 48           PHA                - FOR EASY
 92  045D A9 3B        LDA  #>(QUERY-1)  - RTS TO
 93  045F 48           PHA                - MONITOR
 94  0460 BD 21 06     LDA  PGMA+1,X  PUSH ADDR OF
 95  0463 48           PHA              - ROUTINE FOR
 96  0464 BD 20 06     LDA  PGMA,X    - JUMP VIA RTS
 97  0467 48           PHA
 98  0468 60           RTS           JUMP TO ROUTINE
 99                    ;
100  0469 68 EXIT      PLA           PULL RETURN ADDR
101  046A 68           PLA           - TO MONITOR
102  046B A5 45        LDA  MODE     RETURN SCREEN TO
103  046D 8D 4C E8     STA  59468    - ORIGINAL MODE
104  0470 A9 05        LDA  #05      CLOSE 5
105  0472 A6 44        LDX  VER
106  0474 F0 06        BEQ  EXIT3
107  0476 20 E2 F2     JSR  $F2E2    4.0 CLOSE
108  0479 4C CC FF     JMP  $FFCC    RESTORE DEFAULT I/O
109  047C 20 AE F2 EXIT3 JSR $F2AE   3.0 CLOSE
110  047F 4C CC FF     JMP  $FFCC    RESTORE DEFAULT I/O
111                    ;
112  0482 A9 2A ONLINE LDA  #>MSG1   PRINT ONLINE
113  0484 A0 06        LDY  #<MSG1
114  0486 20 09 05     JSR  PRMSG
115  0489 A9 00        LDA  #00
116  048B 85 40        STA  CTRL     RESET CTRL FLAG
117  048D 20 7C 05 MAIN  JSR  RECV   RECIEVE AND ECHO
118  0490 F0 03        BEQ  INKEY    WHEN NOTHING
119  0492 20 49 05     JSR  STORE    CHAR TO BUFFER
120  0495 20 13 05 INKEY JSR  TSTOP  CHANCE TO EXIT
121  0498 20 C6 05     JSR  KYBD     TRY THE KEYBOARD
122  049B F0 F0        BEQ  MAIN     WHEN NOTHING
123  049D 20 BB 05     JSR  XMIT     SEND IT OUT
124  04A0 4C 8D 04     JMP  MAIN     LOOP
125                    ;
126  04A3 A9 0D FORWRD LDA  #CR      NO MSG NECESSARY
127  04A5 20 D2 FF     JSR  $FFD2    PRINT CR
128  04A8 A2 13 PAGE   LDX  #19      20 LINES PER 'PAGE'
129  04AA 86 42        STX  LCNT     SET THE COUNTER
130  04AC A2 27 LOOP1  LDX  #$27     SET CHAR COUNTER
131  04AE 86 41        STX  CCNT     - FOR LONG LINES
132  04B0 20 24 05 LOOP  JSR  TSTOP2 STOP KEY TEST
133  04B3 20 4F 05     JSR  FETCH    RETRIEVE NEXT CHAR
134  04B6 F0 39        BEQ  EOFMSG   END OF TEXT
135  04B8 20 6C 05     JSR  PRINT    DISPLAY IT
136  04BB C9 0D        CMP  #CR      END OF LINE ?
137  04BD F0 04        BEQ  EOL      YES
138  04BF C6 41        DEC  CCNT     COUNT CHARACTER
139  04C1 10 ED        BPL  LOOP     NOT END OF LINE
140  04C3 C6 42 EOL    DEC  LCNT     COUNT THE LINE
141  04C5 10 E5        BPL  LOOP1    WHEN NOT END OF PAGE
```

NEW - LOW COST FLOPPY DISK FOR PET*!

PETDISK II

WHY PEDISK?

FAST — The *Pedisk* system loads directly to main memory from the disk and loads — saves information at least 3 times faster than any other disk available.

SIMPLE — A simple command syntax makes its BASIC commands easy to use. At the same time, its sophisticated indexed, sequential, or relative file handling makes it powerful.

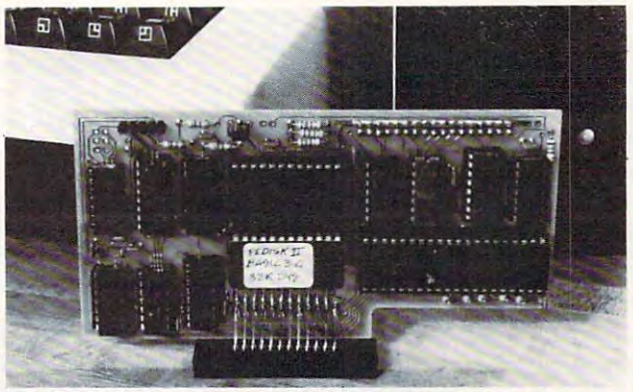
RELIABLE — The ultra-simple circuit design with its LSI disk controller chip provides maximum timing margins for error free operation.

5¼" system — 1 drive, double density (143K)	$595.00
5¼" system — 1 drive, quad density (286K)	$795.00
8" system — 1 drive, IBM 3740 business (295K)	$1,295.00
2 drives, double density (286K)	$945.00
2 drives, quad density (572K)	$1,195.00
2 drives, IBM 3740 business (590K)	$1,895.00

Introducing **PEDISK II,** a low cost high performance floppy disk system consisting of: 1) *PEDISK II* controller, 2) flat disk cable, 3) Disk Drive Assembly, and 4) PDOS II software. The *PEDISK II* is a full function peripheral that provides high speed program storage and a sophisticated file handling package.

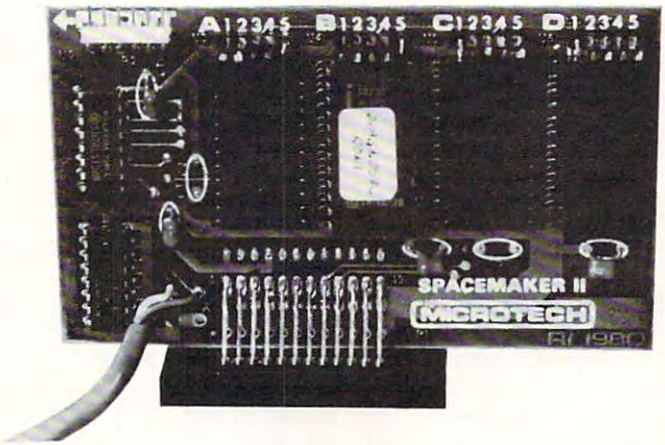
The small *PEDISK II* controller board mounts inside any 2000, 4000, or 8000 series machine. Standard systems are available with one or two disk drives. An optional third drive can also be added. The *PEDISK II* System can be operated simultaneously with any Commodore disk system for data exchange purposes.

The PDOS II software links BASIC by adding a new repertoire of disk commands. !OPEN, !CLOSE, !INPUT, and !PRINT provide the basis of the powerful file handling package. !LOAD, !SAVE and !RUN allow complete disk control. PDOS II also offers a full DOS-mode of operation for all disk diagnostic and utility functions. Diskette format, backup, diagnostic, and reorganization capabilities are provided.



PEDISK II CONTROLLER BOARD

NEW - SPACEMAKER II switch between one of FOUR ROMS



SPACEMAKER II is the new ROM switch from Microtech. It allows either manual or software controlled switching of up to four ROMs in a single ROM expansion socket. The switching is accomplished with a side-mounted slide switch or via ROMDRIVER, an accessory board which allows software controlled switching and keyboard controlled switching. ROM I/O is a special software package available on disk to implement full keyboard control of the ROMs. In addition, it adjusts for memory differences found in various utility ROMs.

SPACEMAKER II	$39.00
ROMDRIVER	$39.00
ROM I/O Commodore or Pedisk	$ 9.95

SEE YOUR LOCAL DEALER FOR ALL cgrs MICROTECH PRODUCTS

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

*PET IS A TRADEMARK OF COMMODORE

8. Allows creation of discrete files.

9. Easy review of files with page control.

In general, the program operates as follows: A prompt line "o/r/f/x/?" is displayed and you make your selection by touching the proper key. You may use upper or lower case commands. There will not be a flashing cursor at this point. The options are Online, Rewind, Forward, and Exit. The STOP key gets you out of the online and forward modes and returns you to the prompt line. Exit takes you back to the PET READY mode, so there are really only three program functions.

In the online mode, a flashing cursor will appear. All characters received (including the echoes of those sent) will be stored in a buffer. This buffer is circular, and extends from address $0700 to the upper limit of memory. During program initialization an EOF (end-of-file) mark is put into the first buffer location. When the STOP key is used to go offline, another EOF is inserted and its address is printed for reference.

The Rewind function moves the buffer pointer backwards to the previous EOF and again prints the address.

The Forward function re-prints the stored information from the current location up to the next EOF. Forward will print twenty screen lines and pause until you hit any key. The STOP key can be used to get back to the prompt mode. It will not insert an EOF when used in Forward, but will leave the pointer at the current location.

There is no fast forward function that will jump you to the next EOF without printing, but if you know the address of the EOF you want to reach, you can get there quickly with one or more rewind commands.

File Control

Going in and out of the online mode does not break the line. You stop sending and receiving,

```
142  04C7 20 E4 FF   KWAIT   JSR  $FFE4   WAIT FOR A KEY
143  04CA F0 FB              BEQ  KWAIT   - BEFORE DOING
144  04CC D0 DA              BNE  PAGE    - NEXT PAGE
145                  ;
146  04CE A9 34      REWIND  LDA  #>MSG2  PRINT REWIND
147  04D0 A0 06              LDY  #<MSG2
148  04D2 20 09 05           JSR  PRMSG
149  04D5 A2 00              LDX  #00     FOR INDEXED INDIRECT
150  04D7 A0 06              LDY  #<TOP   HIGH PAGE OF THIS PGM
151  04D9 C6 11      REW2    DEC  BPT     DECREMENT POINTER LO
152  04DB A5 11              LDA  BPT     TEST FOR PAGE CHANGE
153  04DD C9 FF              CMP  #$FF
154  04DF D0 0C              BNE  REW3    OK, NO PAGE CHANGE
155  04E1 C6 12              DEC  BPT+1   DECREMENT PAGE
156  04E3 C4 12              CPY  BPT+1   LOWER THAN LIMIT ?
157  04E5 D0 06              BNE  REW3    NO, NO WRAP
158  04E7 A5 35              LDA  $35     HIMEM PAGE
159  04E9 85 12              STA  BPT+1
160  04EB C6 12              DEC  BPT+1   PTR= HIMEM-1
161  04ED A1 11      REW3    LDA  (BPT,X)  LOOK FOR EOF
162  04EF D0 E8              BNE  REW2     NO, KEEP TRYING
163  04F1 A9 3E      EOFMSG  LDA  #>MSG5  PRINT EOF
164  04F3 A0 06              LDY  #<MSG5
165  04F5 20 09 05           JSR  PRMSG
166  04F8 A5 12              LDA  BPT+1   PRINT EOF ADDR
167  04FA 20 33 05           JSR  WROB
168  04FD A5 11              LDA  BPT
169  04FF 20 33 05           JSR  WROB
170  0502 A9 44              LDA  #>MSG5B PRINT OFF,CRLF,
171  0504 A0 06              LDY  #<MSG5B
172  0506 4C 09 05           JMP  PRMSG
173                  ;
174  0509 A6 44      PRMSG   LDX  VER
175  050B F0 03              BEQ  PRM3
176  050D 4C 1D BB           JMP  $BB1D   4.0 PRINT STRING
177  0510 4C 1C CA   PRM3    JMP  $CA1C   3.0
178                  ;
179  0513 A6 9B      TSTOP   LDX  $9B     FROM ONLINE
180  0515 E0 EF              CPX  #$EF    9B IS STOP/RVS FLAG
181  0517 D0 19              BNE  STOP4   WHEN NOT STOP
182  0519 A9 00              LDA  #00     INSERT EOF
183  051B 20 49 05           JSR  STORE   - THEN PRINT
184  051E 20 F1 04           JSR  EOFMSG  - EOF ADDR
185  0521 4C 2A 05           JMP  STOP3
186  0524 A2 EF      TSTOP2  LDX  #$EF    FROM REPLAY
187  0526 E4 9B              CPX  $9B
188  0528 D0 08              BNE  STOP4
189  052A A9 00      STOP3   LDA  #00
190  052C 85 9E              STA  $9E     CLEAR KYBD BUFFER
191  052E E6 A7              INC  $A7     DISABLE CURSOR FLASH
192  0530 68                 PLA          RETURN TO MONITOR
193  0531 68                 PLA
194  0532 60         STOP4   RTS
195                  ;
196  0533 48         WROB    PHA          CONVERT HEX TO ASCII
197  0534 4A                 LSRA         SHIFT HIGH
198  0535 4A                 LSRA         - NYBBLE TO
199  0536 4A                 LSRA         - LOW END
200  0537 4A                 LSRA
201  0538 20 3E 05           JSR  HEXASC  - THEN CONVERT AND PRINT
202  053B 68                 PLA          - ONE HEX DIGIT
203  053C 29 0F              AND  #$0F    DO THE LOW NYBBLE
204  053E 09 30      HEXASC  ORA  #$30    ADD THE ASCII BITS
205  0540 C9 3A              CMP  #$3A    GREATER THAN 9 ?
206  0542 90 02              BCC  HEXASC2  NO
207  0544 69 86              ADC  #$86    YES, ADD 7 & UC
208  0546 4C D2 FF   HEXASC2 JMP  $FFD2   PRINT ONE HEX DIGIT
209                  ;
210  0549 20 55 05   STORE   JSR  PTRUP   PUT CHAR INTO
211  054C 91 11              STA  (BPT),Y  BUFFER
212  054E 60                 RTS
```

but this is transparent to the remote system. Because of this, and the fact that the STOP key inserts an EOF, it is possible to isolate particular sections of text for use later.

For example, suppose you want to send the command "TYPE GOOD.NEWS" and isolate the returned file. Type the command, but do not send the RETURN that ends it. Next, hit STOP to insert an EOF and give you the prompt line. You could now rewind if desired, but it is not necessary. Next, go online again and send the RETURN. When the entire file has been received and the other system has sent its ready prompt, hit STOP and immediately go online again. Your file is now bracketed by EOF marks and can be rewound and replayed at will.

Isolated files can be saved on tape by using the monitor. This program, or a similar one, will be needed to read them.

The Keyboard

The program will put your PET into the lower case mode. On exit, the original mode will be restored.

The function of the shift keys will be modified. They will function more like the CAPS LOCK key found on some terminals. On graphic keyboards, they will affect only the alpha characters and the "high specials" (left bracket, backslash, right bracket, up arrow, and left arrow). These last characters, when shifted, represent respectively those characters having ASCII codes 7B through 7F. Because the PET does not have symbols corresponding to these codes, they will be displayed as PET graphics. If a shifted left arrow (ASCII del) is sent, the echo will be ignored.

With a business keyboard, the action of the shift keys is slightly different. For keys with one symbol, the action will be as described above. For keys with two symbols, shift will behave normally.

```
213                         ;
214  054F 20 55 05  FETCH   JSR  PTRUP    FETCH NEXT CHAR
215  0552 B1 11             LDA  (BPT),Y  - FROM BUFFER
216  0554 60                RTS
217                         ;
218  0555 E6 11     PTRUP   INC  BPT      INCREMENT PTR
219  0557 D0 10             BNE  RET2     WHEN NO PAGE CHANGE
220  0559 E6 12             INC  BPT+1    INCR PAGE
221  055B A4 12             LDY  BPT+1    TEST PAGE VS
222  055D C4 35             CPY  $35      - HIMEM (HIGH)
223  055F D0 08             BNE  RET2     NO WRAP
224  0561 A0 07     RESET   LDY  #<TOP+1  RESET BUFFER PTR TO
225  0563 84 12             STY  BPT+1    - START OF BUFFER
226  0565 A0 00             LDY  #00      - WHICH BEGINS ON PAGE
227  0567 84 11             STY  BPT      - FOLLOWING THIS PGM
228  0569 A0 00     RET2    LDY  #00      FOR STORE & FETCH
229  056B 60                RTS
230                         ;
231  056C 48        PRINT   PHA           BEFORE PRINTING, ERASE
232  056D A9 20             LDA  #$20     - FLASHING CURSOR
233  056F A4 C6             LDY  $C6      CURSOR COLUMN
234  0571 91 C4             STA  ($C4),Y  C4 IS ADDR OF LINE START
235  0573 68                PLA           RESTORE ORIGINAL CHAR
236  0574 20 D2 FF          JSR  $FFD2    PRINT IT
237  0577 A2 00             LDX  #00      RESET THE
238  0579 86 CD             STX  $CD      - QUOTE FLAG
239  057B 60                RTS
240                         ;
241                         ;
242  057C A2 05     RECV    LDX  #05      INPUT FROM FILE 5
243  057E 20 C6 FF          JSR  $FFC6    DEVICE = MODEM
244  0581 20 E4 FF          JSR  $FFE4    GET#5
245  0584 A6 96             LDX  $96      TEST ST (STATUS)
246  0586 D0 2F             BNE  NULL     NOT OK
247  0588 29 7F             AND  #$7F     MASK PARITY BIT
248  058A C9 0D             CMP  #CR      CR IS THE ONLY
249  058C F0 1E             BEQ  ECHO     - SPECIAL CHARACTER
250  058E C9 20             CMP  #$20     - ACCEPTED. REJECT
251  0590 90 25             BCC  NULL     - CONTROL CHARS
252  0592 C9 7F             CMP  #$7F     REJECT DEL
253  0594 B0 21             BCS  NULL     IF SENT, NO ECHO
254  0596 C9 60             CMP  #$60     IS IT LOWERCASE ?
255  0598 90 08             BCC  UCIN     NO, GOTO UC CONVERSION
256  059A 29 DF             AND  #$DF     MASK ASCII LC BIT
257  059C C9 5B             CMP  #'Z'+1   LEFT BRACKET
258  059E B0 0A             BCS  ORB7     SHIFT HIGH SPECIAL
259  05A0 90 0A             BCC  ECHO     LC BECOMES SHIFTED
260  05A2 C9 41     UCIN    CMP  #'A'     IS IT LESS THAN 'A'
261  05A4 90 06             BCC  ECHO     YES, NO CHANGE
262  05A6 C9 5B             CMP  #'Z'+1
263  05A8 B0 02             BCS  ECHO     YES, DON'T SHIFT IT
264  05AA 09 80     ORB7    ORA  #$80     UC ALPHA BECOMES SHIFTED
265  05AC 20 6C 05  ECHO    JSR  PRINT    DISPLAY INCOMING CHAR
266  05AF 48        FIX     PHA
267  05B0 20 CC FF          JSR  $FFCC    RESTORE DEFAULT I/O
268  05B3 68                PLA
269  05B4 09 00             ORA  #00      TO SET FLAGS
270  05B6 60                RTS           RETURN CHAR IN (A)
271  05B7 A9 00     NULL    LDA  #00      FOR TESTS
272  05B9 F0 F4             BEQ  FIX
273                         ;
274  05BB A2 05     XMIT    LDX  #05      FILENO.
275  05BD 20 C9 FF          JSR  $FFC9    SET OUTPUT DEV =MODEM
276  05C0 20 D2 FF          JSR  $FFD2    PRINT#5
277  05C3 4C CC FF          JMP  $FFCC    RESTORE I/O AND RTS
278                         ;
279  05C6 46 A7     KYBD    LSR  $A7      ENABLE CURSOR FOR GET
280  05C8 20 E4 FF          JSR  $FFE4    FROM KYBD BUFFER
281  05CB F0 3E             BEQ  RETRN2   NOTHING THERE
282  05CD 85 43             STA  ASAVE    RAW CHARACTER
```

Control Characters

Control (CTRL) characters can be sent, but will be ignored when they are received. CTRL characters serve no purpose on the PET system except to make the cursor do funny things. The only exception is carriage return. Line feed (CTRL J) may be sent. It is not needed for CRT printing.

To send a CTRL character, first key RVS (shifted or unshifted). It will print on the screen as an inverse check mark, but will not be sent. The next key, however, will be sent as the CTRL equivalent. This is much simpler than it sounds. To send CTRL C, key RVS followed by C. Once RVS has been keyed, it cannot be cancelled. If you hit it accidentally, follow it with an @. This will send a null (00).

Echoes

Some host computers, upon receiving a character, immediately send it back to the terminal. A system that does this is called a "full duplex" system. Those that do not echo each character are called "half duplex" systems.

Hardware terminals normally have a half/full duplex switch. When this switch is set to the half duplex mode, the terminal itself will echo each character keyed back to its printer or CRT. Terminal programs often have a program option to simulate the function of the half/full switch. This option will cause the program to print each character that it sends out.

Many modems have a similar switch. When this switch is in the half duplex position, the modem will echo each character from the terminal back to the terminal.

When you are communicating with a full duplex system you must set all of your switches to full duplex, or you will receive more than one echo for each keyed character. For a half duplex system, one switch must be set to half duplex.

Because the IEEE modem has such a switch, this program does not need one. If you are connecting to a remote system for the first time, and you are not sure what type of system it is, set the modem switch to full duplex. If each character you send appears twice, flip the modem switch to half duplex.

Test the program thoroughly before attempting to go online with any system. Then, before using a commercial network, get some experience by calling a

```
283  05CF 29 7F        AND  #$7F    MASK CBM SHIFT BIT
284  05D1 C9 12        CMP  #RVS    CTRL KEY ?
285  05D3 D0 11        BNE  KYBD4   WHEN NOT
286  05D5 85 40        STA  CTRL    SET CTRL FLAG =$12
287  05D7 20 D2 FF     JSR  $FFD2   PRINT RVS
288  05DA A9 BA        LDA  #$BA    CHECK MARK
289  05DC 20 D2 FF     JSR  $FFD2
290  05DF A9 92        LDA  #0FF    TURN RVS OFF
291  05E1 20 D2 FF     JSR  $FFD2   - FOR ECHO
292  05E4 D0 14        BNE  NULL2   BRANCH ALWAYS
293  05E6 A4 40   KYBD4 LDY  CTRL    CTRL FLAG SET ?
294  05E8 C0 12        CPY  #RVS
295  05EA D0 06        BNE  TSTCR   NO,
296  05EC C6 40        DEC  CTRL    YES, UNSET IT
297  05EE 29 1F        AND  #$1F    PREPARE CTRL CHAR
298  05F0 D0 19        BNE  RETRN2  AND SEND IT BACK
299  05F2 C9 0D   TSTCR CMP  #CR
300  05F4 F0 15        BEQ  RETRN2
301  05F6 C9 20        CMP  #$20    LESS THAN SPACE ? (CURSOR CTRL)
302  05F8 B0 03        BCS  TESTA   NO,
303  05FA A9 00   NULL2 LDA  #00     YES, IGNORE IT
304  05FC 60           RTS
305  05FD C9 41   TESTA CMP  #'A'    SPECIAL OR NUMERIC ?
306  05FF 90 0A        BCC  RETRN2  YES, RETURN UNSHIFTED
307  0601 A6 43        LDX  ASAVE
308  0603 30 09        BMI  SHFTED  FOR UC ALPHA AND HIGH SPEC.
309  0605 C9 5B        CMP  #'Z'+1  LEFT BRACKET
310  0607 B0 02        BCS  RETRN2  YES, DONT SHIFT HIGH SPECIALS
311  0609 09 20   ORB5  ORA  #$20    INSERT LC BIT FOR UNSHIFTED
312  060B 09 00   RETRN2 ORA  #00    RETURN WITH CHAR IN (A)
313  060D 60           RTS          AND FLAGS SET
314  060E C9 5B   SHFTED CMP  #'Z'+1 IF HIGH SPECIAL,
315  0610 B0 F7        BCS  ORB5    SEND SHIFTED
316  0612 90 F7        BCC  RETRN2  SEND UC ALPHA WITHOUT SHIFT
317                    ;
318  0614 0D      PROMPT BYT  CR
319  0615 4F 2F 52 CMDS   BYT  'O/R/F/X/? ',00
320  0618 2F 46 2F
321  061B 58 2F 3F
322  061E 20 00
323                    ;
324  0620 81 04   PGMA  WORD ONLINE-1
325  0622 CD 04         WORD REWIND-1
326  0624 A2 04         WORD FORWRD-1
327  0626 68 04         WORD EXIT-1
328  0628 46 06         WORD TOP-1   FOR A PATCH
329                    ;
330  062A 12 4F 4E MSG1   BYT  RVS,'ONLINE',OFF,CR,00
331  062D 4C 49 4E
332  0630 45 92 0D
333  0633 00
334  0634 12 52 45 MSG2   BYT  RVS,'REWIND',OFF,' ',00
335  0637 57 49 4E
336  063A 44 92 20
337  063D 00
338  063E 12 45 4F MSG5   BYT  RVS,'EOF ',00
339  0641 46 20 00
340  0644 92 0D 00 MSG5B  BYT  OFF,CR,00
341                    ;
342  0647 EA       TOP    NOP          COMPUTED PGM END ADDR
343  0648                END
 TOTAL ASSEMBLER ERRORS =   0
 TOTAL WARNINGS         =   0
```

CBBS (Computer Bulletin Board System). These free systems are set up by clubs (or by very special individuals). Check with your local computer club for details. You may have so much fun that you will forget about the commercial networks.

For Hackers And Bit Twiddlers

Being one of you myself, I have provided the means to add another program function. The ? in the prompt can be changed to any character you want. The search count byte at 044D can be increased to 8. The address table for the routines contains a dummy entry. What more do you need? Don't forget to change the value of TOP in the Rewind and Reset routines.

STP-488 Smart Terminal Program

If you would like to have the program on tape send me a blank tape, a stamped, self-addressed mailer, and $3.00. Sorry — no disk. ©

4.0 Garbage Collection: A Small Bug

Jim Butterfield
Toronto, Canada

One of the nicest things about 4.0 Basic is that garbage collection problems are completely eliminated. Well, not quite ...

The following problem is a rare bug. A number of things must come together within your processor before it has a chance of happening:

— You must be almost completely out of memory space. By this, I don't mean that the processor has gotten to the point where it needs to do a garbage collection; rather, the processor must have very little space left even after the collection. The processor must be on the edge of saying OUT OF MEMORY.

— You must be in the process of concatenating at least three strings. "Concatenating" is a good word to use when you want to impress your friends (pronounce it with emphasis on the cat). It means: joining together. A three-way concatenation might be coded as: M$ = A$ = B$ = C$ or PRINT J$ = " " = M$. The = sign does the concatenation; it joins the strings together.

Seeing The Bug

Now we know the rules, let's inspect the bug. Remember, it only happens on 4.0 Basic machines, so users with earlier ROMs won't be able to join in the fun.

Try the following program:

```
100 X=FRE(0)/5-10
110 DIM A(X)
120 Z$=Z$+"X"+"="
130 PRINT Z$:GOTO 120
```

The first two lines guarantee that we will have very little working memory, and that the program will very quickly stop with an OUT OF MEMORY message. The calculation of line 100 is a great leveller: whether you have 8, 16 or 32K, you'll run out of memory very promptly.

The last two lines build a string which continually increases in size. Z$ progressively becomes: X=, then X=X=, then X=X=X= and so on. But as you will see when you run the program, something goes badly amiss just before the OUT OF MEMORY halt. Variable Z$ starts to pick up completely wrong values.

Why It Happens

If the computer was performing a two-way concatenation (e.g. Z$ = Z$ + "X"), it would join the two strings together and store the result. No problem. But with three or more strings to put together, PET must save an intermediate result — and that's where it gets into trouble.

The intermediate result is held in memory, of course. But since it's not permanent, the garbage collection procedure pays no attention to it. Most strings are moved, but not the intermediate one. To make the situation even more complex: the computer, ignoring the temporary string, loses track of how much space is really available in which to stage the concatenation. If it had all the facts, it would decide that it didn't have enough space for the job, and would report OUT OF MEMORY immediately. Instead, it tries the job — even though there isn't enough memory — and ends up botching it.

What To Do About It

There are several easy ways out of this problem.

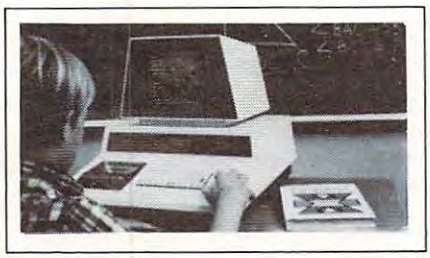
It's likely that Commodore will make a corrected ROM available in the near future to clean up the problem permanently. In the meantime, however, you can get around the potential dangers with a little coding.

One way is to make sure that you'll never run the processor out of memory. If there's any doubt, you could sprinkle a few memory tests of your own into your program. For example, a line like IF FRE(0)<768 THEN PRINT "OUT OF MEMORY" : STOP would guarantee that you would always have space for three jumbo-sized strings in memory — or would stop if you didn't.

Another way is to avoid multi-way concatenations. So long as you join only two strings together at a time, you'll be safe. For example, try changing line 120 in the test program above to: Z$ = Z$ + "X" : Z$ = Z$ + "+". The problem will go away. ©

Skyles Electric Works

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



A. With the Regent.

Q. What is the Regent?

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

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

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

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

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

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

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

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

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

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

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

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

Q. How expensive are these classroom miracles?

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

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

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

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

Using The Monitor On The Pet

Eric Brandon

Supposedly, everyone knows how to use the monitor in the PET computer, right? A lot of people seem confused to me, however, and many authors of articles seem to feel they have to explain how to SAVE their programs using the monitor, how to enter their programs using the monitor, and so on.

There are basically two versions of the monitor, one is for the Original ROM PET, and must be LOADed from tape. The monitors in the upgrade ROM and BASIC 4.0 ROM are identical to the user, and can be entered with an SYS 4 or SYS 1024 from BASIC. Where there are differences in usage, I will point them out.

One difference I should point out right here, however, is that, with the Original ROM monitor, it is difficult to program in BASIC and use the monitor at the same time, since BASIC programs reside in the same area of memory as the monitor. One way to overcome this problem is presented in **COMPUTE!** #4 ("Relocate PET Monitor Almost Anywhere") and **COMPUTE!** #5 ("After the Monitor's Moved"), both by Roy Busdiecker.

When you first enter the monitor, what you see is a letter (either "B" or "C"), followed by an asterisk. A "B" indicates you have entered it via a machine language BRK instruction such as when you use SYS 1024. A "C" indicates you have Called the monitor directly. The only time you really need to Call the monitor is when you want to enter it on the BASIC 4.0 ROM without canceling a CMD instruction. This is done with a SYS 54386. A further explanation of this can be found in **COMPUTE!** #11 ("Working With BASIC 4.0") by Jim Butterfield.

The next thing you see is a list of the 6502's registers. This list can also be called up by the "R" command, and looks like this:

```
     PC   IRQ  SR  AC  XR  YR  SP
.;   0401 E62E 32  04  5E  00  FE
```

The hexadecimal numbers are in left to right order: the Program Counter, the Interrupt ReQuest vector, the Status Register, the ACcumulator, the X Register, the Y Register, and the Stack Pointer. You can move the cursor over any of these values, and change them to whatever you wish by simply overstriking the old numbers. The Original ROM monitor does not supply you with the value of the IRQ vector; this can be found at $0219 and $021A.

One of the commands available to you at this point is "M". This command allows you to see the value of memory locations in hexadecimal format. For example, to view the bytes from $033A to $0400, you would type:

```
M 033A 0400
```

You can stop such a listing by hitting the STOP key. You may also use the cursor to overstrike the hex numbers with any values you wish. You must hit RETURN over each line, however, to enter the changes into memory.

Another command available is "G". This means Go, and is used to execute machine language programs. If you wished to execute a program beginning at $033A, for example, you would type in:

```
G 033A
```

This is equivalent to SYS 826. If you enter just "G", you will execute instructions beginning where the Program Counter is pointing.

The "L" command will LOAD bytes from tape or disk. The format is:

```
L "NAMEPROG",DN
```

where DN is the device number (01 for tape, 08 for disk) in hex. For the original ROM, the format is:

```
L DN,NAMEPROG
```

Note that no quotation marks are necessary. An advantage of LOADing from the monitor is that certain pointers are conserved. If, for instance, you LOADed (from BASIC) a machine language program into the second cassette buffer, and then typed in a line of BASIC, your machine would hang up since your variable pointer would be below $0400, the beginning of your BASIC program! Using the monitor, however, you would not have any of these problems. If you must LOAD machine language from BASIC, type "NEW:CLR" after LOADing.

The "S" command is used to SAVE memory to tape or disk. The format is:

```
S "NAMEPROG",DN,BADD,EADD
```

where DN is the device number, BADD is the beginning address in hex, and EADD is the ending address *plus one* in hex. For instance, to SAVE a program called TEST that resides from $033A to $03C0 onto drive 0 of the disk, you would type in:

```
S "0:TEST",08,033A,03C1
```

The format for the Original ROM monitor is different. To SAVE the same program to tape you would type in:

```
S 01,TEST,033A,03C1
```

As with LOAD, no quotation marks are necessary.

The last command is "X" which means eXit. It takes you out of the monitor and returns you to BASIC. This command should be taught to every computer science student, since they somehow manage to end up in the monitor despite the fact

COMMU-NICATE!

McTerm

Telecommunications For The Commodore

*Your Commodore computer becomes a very intelligent terminal. Operates with a modem over ordinary telephone lines.

*Let's you use the power and memory of distant mainframe computers. Write programs on your micro, run them on the big ones.

*Access giant data bases all over the world.

*Tune into Micronet or the "Source" for up to the minute news, sports, weather, stock market reports, electronic mail airline reservations, games, etc.

*Automatically record all or any part of your transmissions onto disc or printer.

*Save time and money when you store your data on disk and send it in batch. No more slow on-line data entry. Transfers programs, sequential data, or WordPro files.

*Sends and receives on the PET much faster than anything else on the market -- up to 1200 baud.

*Customize McTerm to meet your needs: have a number of stores that need to communicate? Link them with McTerm.

*Auto dial/auto answer available as option.

******* Only $195 complete *******

Features:
*Baud selection from 75 to 1200
*Full and half duplex, local echo
*Supports odd, even, and mark parity
*CRC, error checking for transfers between PETS
*Works with 3.0 or 4.0 Basic, 1.0 or 2.0 DOS, 40 or 80 col. computers, 2040, 4040, or 8050 disk drives.
*Works with most RS-232 modems

- -

YES! SEND ME THE FOLLOWING:
☐ **Complete McTerm Communications package with program, ROM, cable and manual. I've enclosed $195.00**
Basic Version? ☐ 4.0 ☐ Other
If other, which keyboard?
☐ business ☐ graphics
☐ **Send more information on McTerm.**

Name _____

Address _____

_____ **Zip** _____

Phone ( ) _____

MADISON COMPUTER
Dealer Inquiries Welcome

1825 Monroe Street
Madison, WI 53711
(608)255-5552

- -

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 rewritten 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

that they don't know how to leave BASIC.

You may have also heard of something called an extended monitor. This is a program that adds useful commands such as Disassemble, and Fill Memory, to the resident monitor. Some of the more popular ones are Supermon, Extramon, or if you have the Original ROM, Newmon. They are quite useful, and should be in every PET owner's program library. ©

Odds And Ends

Louis C. Ray
Los Angeles, CA

In issue 6 of **COMPUTE!** Jim Butterfield described the various ROM sets that are available for the CBM/PET computers and the Commodore disk systems. On the strength of that article I proceeded to acquire the BASIC 3.0 ROMs for my model 2001-8 PET and a 2040 disk with DOS 2.1.

I wanted to use the "relative file mode" of the DOS 2.1 program in the 2040 disk drive. However, the manuals from Commodore do not indicate how the relative positioning feature can be called from a computer without BASIC 4.0. Unfortunately, my Commodore computer is an old model with static RAMs that cannot be easily upgraded to use the BASIC 4.0 ROMs. Phone calls to several Commodore district offices yielded no help, although most of Commodore folks were friendly and sympathetic.

After some experimentation, I have discovered the syntax necessary for the RECORD command of BASIC 4.0 to be simulated by BASIC 3.0. It is as follows:

PRINT#15,"P"ch/rcl/rch/b

where **ch** is the channel open to the relative file (in binary)
rcl is the low order half of the record number (in binary)
rch is the high order half of the record number (in binary)
b is the byte pointer (in binary) If not given, a 1 is assumed for the byte pointer.

The following short program shows how I get the "relative files" with my PET.

```
10 OPEN15,8,15:LF=10
20 OPENLF,8,LF,"0:TEST,L80,W"
22 INPUT"RECORD NUMBER    ":RN%
23 IFRN%>254ORRN%<0THEN22
24 PRINT#15,"P";CHR$(LF);CHR$(RN%);CHR$(0);
25 GOSUB400:IFEN=50THEN600
26 INPUT"READ OR WRITE  W";C$
27 IFLEFT$(C$,1)="R"GOTO500
28 IFLEFT$(C$,1)="A"THEN50
29 GOTO600
50 CLOSE3
60 STOP
400 INPUT#15,EN,EM$,ET,ES
405 IFEN=0THENRETURN
410 PRINTEN;EM$;ET;ES:RETURN
420 CLOSE3:CLOSE15:STOP
500 INPUT#LF,A$
510 PRINTA$:GOTO22
600 PRINT"TYPE LINE":INPUT"    ";A$
  610 PRINT#LF,A$;CHR$(13);
620 GOTO22
```

©

PETTERM™

Terminal Program

Intelligent Communication

Uploads and Downloads

In Machine Language

Use your PET as an intelligent terminal. Access timeshare systems and networks such as the Source or Telenet. This is an interrupt driven system with buffered input/output. Unlike dumb terminals, the PET can do other tasks instead of having to wait for incoming data. This flexibility allows many advanced features.

These are complete assembled hardware and software packages. All include line editing/resend, auto-repeat, shift lock, output to IEEE printer and much more. Delivered on cassette or disk. Also, available for Commodore 8010 modem or Livermore Star 488 modem.

PETTERM I All features above ..... **$ 80.00**
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

A Programming Revolution

FORTH Interactive high level compiler and operating system 5–10 times faster than PET BASIC. High level block structured language. This is a true fig-FORTH implementation for 16 and 32K PETs.

fig-FORTH. with editor, assembler **$50.00**
with floating point and strings **$60.00**
with floating point. strings. macro-assembler .. **$70.00**
cassette and disk interface available.

DEALERS INQUIRE

F S S

software for small computers

1903 Rio Grande
Austin, Texas
78705

1-512-477-2207

P.O. Box 8403
Austin, Texas
78712

Source: TCE995

2040 Disk Program Listing

David M. Conley
Santee, CA

This is a program for the PET and the 2040 disk drive, that will list, in alphabetical order, every file and program on all your disks. This is done in a two-step process.

The first step is to read the directory off of each disk, one by one, and put this data onto a disk, as a sequential file. The disk being read goes in drive 0, and the file is put on the disk in drive 1. The PET checks the name you give each disk against a list of legal names, and will refuse a bad name. When it writes a file, the PET first scratches any *old* file with the same name. This makes updates automatic.

When you are done inputing/updating the disk directory files, Input an up arrow for the disk name, and the PET will start sorting the names.

This sorting routine and a print-out routine are in the second part of the program. It reads each file in drive 1, using the list of names in the DATA statements, and, on the first pass, gets all the programs that start with any non-alphabetical character. It then sorts these programs, and prints them out in order. Then it goes through the files again, this time getting all the ones that start with "A", sorts and prints them. It does this all the way to "Z", and then stops. The end result of all this is a listing of ALL the programs and files you have on ALL your disks.

In this format, this program will show Program Name, Type, Disk name, and Size, and is set up to use the 2022 printer.

For large disk libraries, this program is SLOW! To speed it up, you can have the PET look for several letters at once, which will require fewer passes through the files to get all the data. This uses up lots of memory though, so don't try to do too much at once, or you'll run out of memory. Putting too many names into the PET will also

```
- ONE AT A TIME    + THREE AT A TIME
-------------------------------------
- 720 REM ** THIS SORTS FOR 1 LETTER AT A TIME
+ 720 REM ** THIS SORTS FOR 3 LETTERS AT A TIME

- 840 PRINT"□":FOR I=1 TO 27:E$=MID$(A$,I,1)
+ 840 PRINT"□":FOR I=1 TO 27 STEP 3:E$=MID$(A$,I,1)

+ 842 G$=MID$(A$,(I+1),1)

+ 844 H$=MID$(A$,(I+2),1)

- 850 PRINT"■■■LOOKING FOR "E$"'S
+ 850 PRINT"■■■LOOKING FOR "E$", "G$" & "H$"'S

- 980 IF E$<>"*" THEN 1000
+ 980 IF E$="*" THEN 990

+ 982 IF G$="*" THEN 990

+ 984 IF H$="*" THEN 990

+ 986 GOTO 1000

- 1000 IF MID$(F$,2,1)=E$ THEN C$(J)=F$:J=J+1:PRINT"*";
+ 1000 IF MID$(F$,2,1)=E$ THEN C$(J)=F$:J=J+1:PRINT"*";:GOTO 1010

+ 1002 IF MID$(F$,2,1)=G$ THEN C$(J)=F$:J=J+1:PRINT"*";:GOTO 1010

+ 1004 IF MID$(F$,2,1)=H$ THEN C$(J)=F$:J=J+1:PRINT"*";:GOTO 1010
-------------------------------------
- ONE AT A TIME    + THREE AT A TIME
```

drastically increase the time required to sort these names.

Shown below are the changes necessary to look for three letters at a time.

If you have a different printer, insert its routine in the appropriate spot. Do the same thing for the sort routine, if you have a better one. It's probably better to have a disk reserved for this program and data files only. It'll save the 2040 a little time when it looks for a file.

```
100 REM ** BY D. M. CONLEY 3/81
110     ¬
120 REM * THIS PROGRAM READS AND STORES ¬
      ¬THE DIRECTORY FROM A DISK AS A SEQ
130 REM * FILE ON THE DISK IN DRIVE #1. ¬
      ¬ THE DISK BEING READ GOES IN ¬
      ¬DRIVE #0.
140     ¬
150 N=152:DIM D$(N):OPEN15,8,15
160 PRINT"ñ↓↓↓↓TO START PRINT OUT,
      ¬ INPUT '^'
170 INPUT"↓↓↓↓↓DISK NO.   ?←←←";B$
180 IF B$="?" THEN PRINT"↑↑↑↑↑↑↑":
      ¬GOTO 170
190 IF B$="^" THEN CLR:GOTO 740
200 READ E$
210 IF E$="END" THEN PRINT"↓↓↓THAT'S ¬
      ¬NOT A GOOD DISK NAME: TRY AGAIN!
220 IF E$="END" THEN RESTORE:GOTO 170
230 IF E$<>B$ THEN 200
240 RESTORE
250     ¬
260 REM **  READ DIRECTORY
270     ¬
280 PRINT#15,"I0":REM ** INITIALIZE ¬
      ¬DRIVE 0
290 PRINT#15,"M-E"CHR$(212)CHR$(237):
      ¬REM ** DIRECTORY LOAD PATCH
300 PRINT "ñ↓     rDIRECTORY FROM DRIVE ¬
      ¬0î↓"
310 B=30:C$="$0:*":OPEN 2,8,0,C$
320 GOSUB 1410:GET#2,A$:GET#2,A$:I$=""
330 IF ST<>0 GOTO 520
340 FOR A=1 TO B :GET#2,A$:IF A$="" ¬
      ¬THEN A$=CHR$(0)
350 I$=I$+A$:NEXT:G=ASC(MID$(I$,3,1)):
      ¬G=G+ASC(MID$(I$,4,1))*256
360 IF B=30 THEN 510
370 FOR I=6 TO 27:IF MID$(I$,I,1)<>CHR$(
      ¬34)THEN NEXT:GOTO 510
380 D$=MID$(I$,5,27)
390 IF LEFT$(D$,1)=" " THEN J=LEN(D$):
      ¬D$=RIGHT$(D$,(J-1)):GOTO 390
400 D$=D$+B$+STR$(G):K=K+1
410 FA=0:F=0:FOR X=1 TO LEN(D$):
      ¬X$=MID$(D$,X,1)
420 IF X$=CHR$(34) AND F=0 THEN F=1:
      ¬GOTO 480
430 IF X$=CHR$(34) THEN F=0:GOTO 480
440 IF F THEN 480
450 IF X$<>" " THEN FA=0:GOTO 480
460 IF X$=" " AND FA=0 THEN X$="*":FA=1
470 IF X$=" " THEN NEXT X
480 X1$=X1$+X$:NEXT X
490 D$=X1$:X1$=""
500 D$(K)=D$:PRINTD$
510 D$="":I$="":B=32:IF A$<>"S" THEN 330
520 PRINT:C$="READ DIRECTORY":GOSUB 1410
```

```
530 C$="CLOSE 2": CLOSE 2:GOSUB 1410:
    ¬PRINT"ϟϟ"
540 REM  **  PRINT#15,"I1":GOSUB 1420   ¬
    ¬ (THIS IS NOT NORMALLY NEEDED"
550 OPEN 1,8,15
560 E$="S1: "+B$+" ":REM ** SCRATCH OLD ¬
    ¬FILE
570 PRINT#1,E$
580 CLOSE 1
590    ¬
600 E$="@1: "+B$+" ,S,W"
610 OPEN 2,8,2,E$
620 FOR X=1 TO K
630 PRINT#2,D$(X)CHR$(13);:REM *** PUT ¬
    ¬FILE ON DISK
640 NEXT X
650 PRINT#2,"*"CHR$(13);
660 CLOSE 2:GOSUB 1410:K=0:GOTO 160
670    ¬
680    ¬
690     ¬
700 REM  ***  THIS HALF OF THE PROGRAM ¬
    ¬PRINTS OUT THE DIRECTORYS
710     ¬
720 REM ** THIS SORTS FOR 1 LETTER AT A ¬
    ¬TIME
730    ¬
740 A$="*ABCDEFGHIJKLMNOPQRSTUVWXYZ":
    ¬N=1000:DIM C$(N)
750     ¬
760 REM 'N' IS THE MAX NO. OF PROGRAMS ¬
    ¬THAT START WITH THE SAME LETTER ¬
    ¬IN YOUR
770 REM DISKS. BE SURE 'N' IS LARGE ¬
    ¬ENOUGH....BETTER TOO BIG THAN TOO ¬
    ¬SMALL!
780     ¬
790 OPEN 15,8,15:PRINT#15,"I1"
800 GOSUB 1410:J=1
810 OPEN4,4:PRINT#4,"ĥ"
820 PRINT#4,"          PROGRAM NAME     ¬
    ¬ TYPE       DISK  SIZE
830 PRINT#4,"          CCCCCCCCCCCCCCCCC
    ¬CCCCCCCCCCCCCCCCCCCCCCCCC":CLOSE4
840 PRINT"ĥ":FOR I=1 TO 27:E$=MID$(A$,I,
    ¬1)
850 PRINT"ϟϟϟLOOKING FOR "E$"'S
860 PRINT"                    ('*' = ¬
    ¬FILE FOUND)
870 READ B$
880 PRINT:PRINT"SEARCHING DISK #"B$;
890 IF B$="END" THEN RESTORE:GOTO 1030
900 D$="1: "+B$+" ,S,R"
910 OPEN 2,8,2,D$:GOSUB 1410
920 GET#2,F1$:IF F1$=CHR$(13) THEN 940
930 F$=F$+F1$:GOTO 920
940 IF ST THEN CLOSE 2:GOTO 870
950 T=LEN(F$):IF LEFT$(F$,1)="*" THEN ¬
    ¬F$=RIGHT$(F$,(T-1)):GOTO 950
960 REM ** THE ABOVE LINE ELEMINATES   ¬
    ¬A OBSCURE PROBLEM THAT CAUSES A ¬
    ¬"*"
970 REM ** TO BE THE 1ST CHARACTER IN  ¬
    ¬F$.  I DON'T KNOW WHY IT DOES IT!
980 IF E$<>"*" THEN 1000
990 T=ASC(MID$(F$,2,1)):IF T<65 OR T>90 ¬
    ¬THEN C$(J)=F$:J=J+1:PRINT"*";:
    ¬GOTO 1010
1000 IF MID$(F$,2,1)=E$ THEN C$(J)=F$:
    ¬J=J+1:PRINT"*";
1010 F$="":GOTO 920
1020    ¬
1030 REM  **  START SORT
1040 TP=1:LOWER(1)=1:UPPER(1)=J
1050 IF TP<=0 THEN 1240
1060 LB=LOWER(TP):UB=UPPER(TP):TP=TP-1
1070 IF UB<=LB THEN 1050
1080 L=LB:K=UB:TEMP$=C$(L)
1090 IF K<1 THEN 1120
1100 IF TEMP$>=C$(K) THEN 1120
1110 K=K-1:GOTO 1090
1120 IF K<=L THEN C$(L)=TEMP$:GOTO 1190
1130 C$(L)=C$(K):L=L+1
1140 IF L>K THEN 1170
1150 IF C$(L)>=TEMP$ THEN 1170
1160 L=L+1:GOTO 1140
1170 IF K>L THEN C$(K)=C$(L):K=K-1:
    ¬GOTO 1100
1180 C$(K)=TEMP$:L=K
1190 TP=TP+1
1200 IF L-LB<UB-L THEN LOWER(TP)=L+1:
    ¬UPPER(TP)=UB:UB=L-1:GOTO 1070
1210 LOWER(TP)=LB:UPPER(TP)=L-1:LB=L+1
1220 GOTO 1070
1230    ¬
1240 L=0:OPEN4,4:FOR K=1 TO J
1250 X1$=C$(K):IF X1$="" THEN 1310
1260 D$="":F=0:FOR X=1 TO LEN(X1$)
1270 X$=MID$(X1$,X,1):IF X$="*" THEN ¬
    ¬F=F+1:GOTO 1340
1280 D$=D$+X$
1290 NEXT X
1300 IF C$(K)<>"" THEN PRINT#4,"        ¬
    ¬  ";D$:C$(K)="":L=L+1
1310 NEXT K:PRINT:PRINT"FOUND"L"ITEMS":
    ¬CLOSE4
1320 J=1:NEXT I
1330 END
1340 IF F=1 AND LEN(D$)<20 THEN D$=D$+" ¬
    ¬":GOTO 1340
1350 IF F=2 AND LEN(D$)<32 THEN D$=D$+" ¬
    ¬":GOTO 1350
1360 IF F=3 AND LEN(D$)<38 THEN D$=D$+" ¬
    ¬":GOTO 1360
1370 GOTO 1290
1380    ¬
1390 REM ** INPUT FROM ERROR CHANNEL
1400    ¬
1410 INPUT#15,EN,EM$,ET$,ES$:IF EN=0 ¬
    ¬THEN RETURN
1420 IFEN=1THEN RETURN
1430 PRINT"î ERROR #"EN"ɾ"EM$;:IF EN<30 ¬
    ¬THEN PRINT" ON "ET$"."ES$;
1440 PRINT"ϟ":END
1450    ¬
1460       BELOW IS THE LIST OF ACCEPTABL
    ¬E NAMES FOR DISKS.  THESE ARE ¬
    ¬MINE.
1470       PUT YOUR OWN IN PLACE OF ¬
    ¬THESE.
1480       "END" MUST ALWAYS BE THE LAST ¬
    ¬NAME.
1490    ¬
1500 DATA 00,1A,1B,2A,2B,3A,3B,4A,4B,5A,
    ¬5B,6A,6B,7A,7B,8A,8B,9A,9B
1510 DATA 10A,10B,11A,11B,12A,12B,13A,
    ¬13B,14A,14B,15A,15B,16A,16B,17A,
    ¬17B
1520 DATA 18A,18B,19A,19B,20A,20B,21A,
    ¬21B,22A,22B,23A,23B,24A,24B,25A,
    ¬25B
1530 DATA 26A,26B,27A,27B,28A,28B,29A,
    ¬29B,30A,30B
1540 DATA END
```

If you can't get the complete print-out in one sitting, you can easily continue later from where you had to stop. Say, for example, you just got the print-out for the "E" 's, and you had to stop. To start up again from where you stopped, change the FOR... TO... statement in line 840 to read FOR I=7 TO 27. This will cause the PET to start looking for the 7th letter in A$, which is an "F".

The Data is stored as C$(1) to C$(K) in the sort and print routine. After it prints, a letter sort could be saved as a tape or disk file if you wanted, using a routine similar to the one in lines 610 to 660.

Beware of the dreaded Garbage Collection Routine! Because of all the string manipulation, after this program runs for a while, Garbage Collection can eat up a LOT of time. If you have a solution to this problem (No, I don't want to buy a set of 4.0 ROMs), let me know.

This should make keeping track of all those hundreds of programs a little easier, and make it easy to spot unnecessary duplicates that use up valuable disk space. ©

All About LOADing Pet Cassettes

Louis F. Sander
Pittsburgh, PA

This article, based on long experience and on material from many sources, summarizes the major points about PET's LOAD command as it applies to cassette tapes. Disk LOADing is rather different and is not covered here. If you don't yet have a disk drive, or if you have any questions about LOADing from tape, read on.

The Pet Cassette

To understand LOADing, it's helpful to understand what is recorded on the tape itself. It's extremely useful for you to *hear* the recorded material, and you can easily do so by connecting an audio amplifier between the tape drive connector's READ and GND pins. Your CB2 amplifier will do nicely, if you have one. Here is what you'll find on every PET tape:

1. Ten seconds of high-pitched leader tone. This tone prepares PET for LOADing the program, but mainly it was put there to insure that the tape's non-magnetic leader passed by before anything important was recorded. Only a second or so of the tone is required to initiate a LOAD, so you can often save some time by winding your tape past most of it before LOADing. Your audio amplifier or an external recorder will make this step a cinch.

2. A three-second, 192-byte tape header, consisting of one byte to differentiate program and data tapes, four bytes telling PET the starting and ending addresses of the LOAD, and 187 bytes containing the program name plus trailing spaces as needed. (Although the program name can be up to 187 bytes long, most are much shorter because only the first 16 characters print on the screen when FOUND, and because 77 characters at most can be entered as a name in immediate mode.) All this data is recorded twice in the header and if you listen carefully you can hear the tiny gap between the two copies.

3. Two more seconds of leader tone, followed by ...

4. The program itself, repeated twice, with an audible gap between the copies. If the program is in BASIC, each keyword is recorded as a one-byte abbreviation, or "token," to save space in memory and on the tape. On this part of the tape, each 1K of program takes about 18 seconds of playing time, so a 1K program takes about 33 seconds to LOAD, including leader, header, etc.

Many Pet owners like to use C-10 cassettes, recording only one or two programs per side, to save unproductive SEARCHING time. But longer cassettes can hold any number of programs, and PET's ability to LOAD them selectively by name, in immediate or program mode, allows very large and complex programs to be cut into pieces which even the smallest PET can digest.

The LOAD Command

When executed, LOAD transfers a program from tape to PET's memory. A LOAD command can be executed in immediate mode by typing it in on the keyboard or by pressing the shifted RUN/STOP key. LOADing can be done in program mode by executing a line containing a LOAD command, or by POKEing the keyboard buffer. There are important differences in the way PET handles these four methods of execution, and they will be described in detail later. Whatever method is used to execute the LOAD, if the tape is not positioned at the start of a programm when the LOAD is attempted, PET will ignore what it reads until it finds the next program header on the tape.

The format for LOAD is:

LOAD ("program name") (,device number)

The items in parentheses are optional. The devices number can be either 1 or 2, indicating a LOAD from TAPE #1 or TAPE #2. It can also be a numeric variable or expression whose value, after going through the INT function, is either of those numbers. If it is omitted, TAPE #1 is automatically selected. If used, the program name can be enclosed in quotes, or can be specified as a string variable. If no program name is specified, PET will LOAD the next program it finds on the selected tape.

If a program name *has* been specified, PET will search for that name before LOADing. The search is on a character-by-character basis, and a match is made whenever each character in "program name" is found in the corresponding position in the name of the FOUND program. This means that LOAD "CAT" will load CAT, CATNIP, CATAPULT, or any other program whose name begins with CAT. Programs such as TOMCAT will *not* cause a match, because the letters in CAT will not be found in the right positions. Programs like CA won't match either, because there is nothing to match with the T.

This method of searching can be useful when you enter

LOAD "STAR"

because you're not sure if your program is called STARWARS or STARFIGHTER, but it can some-

times cause PET to load an unwanted program, such a STARVATION.

The following examples show LOAD's valid formats and the resulting screen dialogue:

PET LOAD Command Formats

To LOAD From TAPE #1	To LOAD From TAPE #2*	Action
LOAD	LOAD "",2	LOAD the next program found on the tape.
LOAD "WIMP"	LOAD "WIMP",2	Search for the program names WIMP, and LOAD it when found.
LOAD A$	LOAD A$,2	Search for the program with the same name as A$, and LOAD it when found.
LOAD A$, J-3	LOAD A$, J-3	Search for A$ on TAPE #(J-3) and LOAD it when found. 1<(J-3)<3

*An alternate way of specifying TAPE #1 is to use this format with a 1 rep

LOAD Command Screen Dialogue

LOAD "WIMP"	This is your command. All else is PET's response.
PRESS PLAY ON TAPE #1	Shown only if no tape controls are down when LOAD is executed.
OK	Shown when you press a tape control.
SEARCHING FOR WIMP	The tape has started to move.
FOUND SPACEWARS	PET has read the header of a program named SPACEWARS.
FOUND WIM	Not a match with "WIMP," since fourth character is missing.
FOUND	Found an unnamed program. No match.
FOUND SPACEWIMP	Not a match, since first character doesn't match.
FOUND WIMPOLESTREET	A match, since all letters searched for are in the right place.
LOADING	WIMPOLESTREET is being loaded into memory.
READY.	The LOAD is complete.

LOADing In Immediate Mode

Pressing the shifted RUN/STOP key in immediate mode LOADs and automatically RUNs the next program on TAPE #1; this is probably the most common method of LOADing PET programs.

LOAD can also be executed in immediate mode by typing the command onto the screen and pressing RETURN. Any of the previously illustrated command formats can be used in this way. In every case, the word LOAD can be abbreviated by typing an L and a shifted O, (even if the shifted O puts a graphics character on the screen). Try this trick if you haven't used it already — it's a good one. Programs loaded by typing in LOAD or 'L shifted O' do *not* automatically begin executing when the LOAD is complete. You must type in RUN (or 'R shifted U'), then press RETURN to start them.

You can find the names of all programs on a tape by entering LOAD "XYZ", where XYZ is any combination of letters *not* starting the name of a program on the tape. If you do this, PET will search the tape from beginning to end, showing you the first 16 characters of the name of each program it finds along the way.

When a program is LOADed in immediate mode, all the appropriate prompts and messages appear on the screen, and the new program completely replaces the old one in memory. On completion of the LOAD, PET performs a CLR, which initializes all variables except TI and TI$, destroys all arrays previously set up, and closes all logical files.

LOADing Under Program Control

There are two ways of having one PET program LOAD another. The first is to have the original program POKE a 131 into the keyboard buffer, then terminate. This is the equivalent of stopping the first program and pressing shifted RUN/STOP in immediate mode. In an Original ROM PET this might be done by:

```
9150 POKE 527,131 : POKE 525,1 : END
```

The Upgrade ROM version of this would be:

```
9150 POKE 623,131 : POKE 158,1 : END
```

When either of these lines es executed, PET will immediately LOAD and RUN the next program it finds on TAPE #1. All the appropriate screen messages will appear, the new program will completely replace the old one in memory, and a CLR will be performed at the end of the LOAD, just as though shifted RUN/STOP had been pressed.

The second method of LOADIng under program control is to use the LOAD command, in any of its previously mentioned formats, in a program line. For example:

```
140 LOAD "WIMP" –or– 150 LOAD A$,2
```

This method has some properties which are at once useful and troublesome:

1. LOAD in a program line stops execution of the current program, LOADs the specified program from wherever it is found on the tape, and begins executing it at once.

2. If PLAY is already pressed, no messages appear on the screen to disturb its appearance. If PLAY has *not* been pressed, the PRESS PLAY message, but no others, will appear on the screen.

3. No CLR is performed after the LOAD, and the start of variables pointer is not reset, so the values of all numeric variables in the new program will be the same as those in the old one. This process is sometimes called "passing parameters" from one program to another. As an example of how it works, if variable KM had a value of 4986 in the old program, and the first line of the new one is 10 PRINT KM, the number 4986 will be printed on the screen. Without the passing parameter feature, of course, KM would have an initial value of 0 in

the new program, and the 0 would be printed on the screen.

4. String variables will be passed to the new program, but only if they have been "operated on" in the old one. The line:

```
150 A$="KATHLEEN"
```

will *not* get KATHLEEN into the new program unless it has been followed by something like:

```
155 A$=A$+""
```

5. User-defined functions will not be reliably passed from the old program to the new one.

6. One program or a whole series of programs can be loaded in this way, but none of them can be longer than the program most recently loaded by one of the other methods, (that is from immediate mode or by POKEing the keyboard buffer). This is an absolute requirement, caused by the fact that the start of variables pointer is not reset by the program mode LOAD. Programs violating this restriction may LOAD all right, but they will not RUN properly, if they RUN at all. You can set the pointer yourself to get around the restriction, if that's your cup of tea.

The above set of properties allows the creative programmer to write very large programs of appreciable complexity, and to break them into subprograms, each of which will fit into PET's memory. The first program can fill the screen with graphics or whatever, make some calculations and, based on their result, ask PET to LOAD any one of many other programs from either tape drive. Because the screen messages are suppressed, the PET operator will not see the LOAD taking place. Nor will he have to take action to RUN the new program. And, because numeric and string parameters can be passed from one program to the next, the second program can work on input, or results, from the first, and on ad infinitum. The only price for this flexibility is the length restriction mentioned above, and the fact that strings need to be operated on in order to be passed from program to program.

Machine Language

The material in this article applies in full only when you are LOADing programs written in BASIC. PET handles machine language programs (MLP) somewhat differently, and a discussion of the differences is beyond our scope at this time. Suffice it to say that the LOAD formats and LOAD execution methods we've described here will load MLPs into PET's memory, but afterwards things begin to differ. MLPs are never automatically RUN by a LOAD command. And the status of memory pointers and other programs afterwards is likely to be quite different from what is seen with BASIC.

The machine language programmer is well-

equipped to delve into these matters on his own and, if he pays careful attention to PET's memory pointers, he's likely to be able to get LOAD to do anything he wants.

LOAD Errors And Other Problems

During a LOAD, PET compares the two copies of the program on the tape. If, for some reason, it finds them different, it tries to decide which copy is correct, and it is often able to do so. But sometimes PET can't correct the errors, and in these cases you'll get a ?LOAD ERROR message. Other times, PET will find and print the program name, but will not LOAD the program. Or it may miss the program name altogether and merrily keep SEARCHING to the end of the tape.

When any of these things happen, you should rewind your tape and try again. If things don't work properly the second time, you should take corrective action. Clean and demagnetize your tape head, as described in **COMPUTE!** #10 ("Getting the Most From Your PET Cassette Deck," page 42) and elsewhere in the PET literature. If this doesn't correct the problem, try LOADing the program several times. If you are at last successful, make sure the program runs properly, SAVE it on a fresh tape, and consider discarding the first copy. Usually the combination of cleaning/demagnetizing/new program copy will cure any persistent LOADing difficulties. If not, you, and perhaps your service dealer, will have to look further.

Along these lines, it's impossible to over emphasize the value of *listening* to your LOADs, since most error situations can be easily heard on the tape. Without trying, I have developed the ability to *hear* most bad loads, and you can do it, too. The ear is an amazingly sensitive organ, even to digital input. Advanced PET troubleshooters can also make use of the STATUS word, and of PEEKing tape error locations.

A word on head alignment is in order here. If you consistently have trouble LOADing tapes created on other PETs, it is likely that your tape deck's read/write head, or the one on the other PET, is out of alignment. **COMPUTE!** #8, contains some good instructions on correcting this problem. ("Detecting Loading Problems and Correcting Alignment on your PET," page 114). Incidentally, head misalignment is *very* easy to detect by ear. ©

SM-KIT

for Commodore Computers

The SM-KIT is a collection of machine language firmware programming and test aids for BASIC programmers. SM-KIT is a 4K ROM (twice the normal capacity) which you simply insert in a single ROM socket on any BASIC 4 CBM/PET–either 80 column or 40 column. Includes both programming aids and disk handling commands.

ERROR DETECTION: the SM-KIT automatically indicates the erroneous line and statement for any BASIC program error.
LINE NUMBERING: the SM-KIT automatically numbers BASIC statements until you turn the function off.
SCREEN OUTPUT: the commands FIND, DUMP, TRACE and DIRECTORY display on the CRT while you hold the RETURN key (display pauses when the key is released). Continuous output is selected with shift-lock.
OUTPUT CONTROL to DISK or PRINTER: in addition to displaying on the CRT, you can direct output to either disk or printer.
HARDCOPY: allows screen displays to be either printed or stored on disk.
FIND: searches all or any part of a program for text or command strings or variable names. Either exact search or wild card search supported.
RENUMBER: the SM-KIT can renumber all or any part of a program. The selective renumbering allows you to move blocks of code within your program.
VARIABLE DUMP: displays the contents of floating point, integer, and string variables (both simple and array). Can display all variables or any selected variables.
TRACE: SM-KIT can trace program execution either continuously or step by step starting with any line number. Selected program variables can be displayed while tracing.
DISK COMMANDS: as in DOS Support (Universal Wedge), the "shorthand" versions of disk commands may be used for displaying disk directory, initializing, copying, scratching files, load and run, etc.
LOAD: SM-KIT can load all or part of BASIC or machine language programs. It can append to a program in memory, overwrite any part of a program, load starting with any absolute memory location, and load without changing variable pointers.
MERGE: allows merging all or any part of a program on disk with a program in memory.
SAVE and VERIFY: SM-KIT provides one step program save and verification. It also allows you to save any part of a program, or any address range.

A Programming Productivity Tool

ONLY $40

A 4K ROM with both programming and disk handling aids.

Developed by (and available in Europe from) SM Softwareverbund-Microcomputer GmbH, Scherbaumstrasse 29, 8000 Munchen 83, Germany

252 Bethlehem Pike
Colmar, PA 18915
215-822-7727
A B Computers

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

NEW PRODUCT

SWARM-100

JUST PLUG IT IN

• No soldering • No messy wires

SOFTWARE SELECTABLE

1. Software select one of two operating systems. (BASIC 2.0 / BASIC 4.0)
2. Software select utility ROMs at conflicting addresses.

$125.00 (U.S.)
($150.00 Canadian)
Add $3.00 shipping to all points outside Canada.

Master Charge and VISA accepted.

For 24 Pin ROM Machines Only.

BATTERIES INCLUDED

Village by the Grange
71 McCaul Street
Toronto, Ontario
Canada M5T 2X1
(416) 596-1405

Graph Plotting Routine

Claud Cleeton
Bellevue, WA

A routine for printing graphs is given in the accompanying listing. I use it in my Stock Market Series of programs written for PET. It will run on all PET's whether old or new ROM's and the 80 column CBM. The program is in BASIC and is oriented in the normal manner rather than rotated 90 degrees as in some other graphical programs. Several equations may be individually plotted and superimposed, a very useful feature for some applications. For example in one of my stock option programs a plot of normal values, as a function of the underlying stock price, for various strategies, allows comparison of different options in a class (such as different times to expiration). Again, in a program for the analysis of a time series of prices, three equations are found representing the long-term trend, the major cycle, and the next shorter cycle. The three curves are plotted superimposed, to compare the effect each has on the projected sum. The listing gives the plotting routine, including drawing and labeling of the coordinates and, as an illustration, a sine and cosine wave are plotted.

In the listing, "clr" is the clear screen character and "home" is cursor home. Lines 50-90 determine whether the machine is one of 40 columns or the 80 column CBM. If using a 40 column machine only, PV, which is used as a multiplier to match the 25 by 80 screen, may be deleted. Screen memory locations start at 32768 (the upper left corner). The 40 column machines end at 33767 (the lower right position) and the 80 column machine ends at 34767. Line 9000 asks for a PRINT RATIO, R, which is multiplied by the ordinate values of the curve being plotted in order to adjust to the display area. The units of ordinate values are given in terms of M, the reciprocal of R.

The horizontal coordinate is drawn by lines 9010-9060 by POKING minus signs into a horizontal line starting at 32768 + 920*PV + 4, or 33692 for the 40 column machine. However, every fifth position is marked with a plus sign by line 9050. This location is two lines up from the bottom of the screen and five positions to the right where the vertical axis will start to allow for ordinate values to be printed to its left. Line 9065 homes the cursor and 9070 moves the cursor down to the first line below the line just drawn for printing of the coordinate values by line 9100.

Lines 9200-9250 draws the vertical coordinate as a series of colons marked with a plus sign at every fifth position. Lines 9300-9305 POKES in the ordinate values to the left of the axis. Lines 9300-9320 prints -10M, lines 9325-9335 prints -5M, line 9340 is zero and 9345-9350 prints 5M. Line 9510 prints the values of a sine function as an * and 9520 the cosine function as a +.

```
50 PRINT "ĥ"
60 PV=PEEK(50003)
70 IF PV=160 THEN 90
80 PV=1:GOTO 9000
90 PV=2
9000 INPUT "PRINT RATIO R=";R:M=1/R
9005 PRINT,,"M=";M
9010 FOR I=4 TO 35*PV STEP PV
9020 FOR J=PV TO 4*PV STEP PV
9030 POKE 32768+920*PV+I+J,45
9040 NEXT J
9050 POKE 32768+920*PV+I,43
9060 NEXT I
9065 PRINT "h"
9070 PRINT "¥¥¥¥¥¥¥¥¥¥¥¥¥¥¥¥¥¥¥¥¥¥¥¥"
9100 PRINT TAB(3);0;TAB(12*PV);10;TAB(22
     ¬*PV);20;TAB(32*PV);30;"X";
9200 FOR I=0 TO 20 STEP 5
9210 FOR J=1 TO 4
9220 POKE 32772+920*PV-40*(I+J)*PV,58
9230 NEXT J
9240 POKE 32772+920*PV-40*PV*I,43
9250 NEXT I
9300 POKE 32768+920*PV,45
9305 POKE 32769+920*PV,49
9310 POKE 32770+920*PV,48
9320 POKE 32771+920*PV,13
9325 POKE 32769+720*PV,45
9330 POKE 32770+720*PV,53
9335 POKE 32771+720*PV,13
9340 POKE 32771+520*PV,48
9345 POKE 32770+320*PV,53
9350 POKE 32771+320*PV,13
9500 FOR X=0 TO 35
9510 POKE 32772+520*PV-INT((10*SIN(X*2* ¬
     ¬π/36))*R+.5)*40*PV+X*PV,42
9520 POKE 32772+520*PV-INT((10*COS(X*2* ¬
     ¬π/36))*R+.5)*40*PV+X*PV,42
9530 NEXT X                                    ©
```

Linelist

G. H. Watson
Physics Department
University of Delaware
Newark, DE

When examining a PET BASIC program for the first time it is certainly convenient to have a listing produced by a printer. No printer means that you must list the program on the screen. When the screen is full, each new line which is listed will scroll an earlier line off the top of the screen. At most, you may view 24 lines of your program. This would be acceptable if only you could stop the listing, examine the contents, and then continue. The listing process may be slowed by depressing the OFF/RVS key and stopped by hitting the RUN/STOP key, but unfortunately no provision was made for an indefinite pause and then continue.

If you have suffered from typing LIST line#–line# over and over or you have tired of trying to strike the STOP key at just the right instant, then LINELIST may be the answer to your problems. LINELIST is a BASIC "loader" for a fairly short machine language program (140 bytes). By loading and running LINELIST, a program will be created at the top of memory — usually just below screen memory and where ASCII strings are stored. With an appropriate SYS command, a listing will be created just as though LIST were entered. The important distinction is that the listing may be halted by holding down the SHIFT key. When the shift key is released, the listing continues. Depressing SHIFT/LOCK frees both hands to copy the program onto paper. Further, one additional line is listed each time another key is struck; I use INST/DEL to avoid having some of the keystrokes displayed when the listing is completed.

How does this program work? Most of the LINELIST machine code is identical to the LIST routine stored in ROM. "Why not just use a JSR instruction to connect to the BASIC LIST routine?" you may ask. This indeed was my original intention; unfortunately no corresponding RTS exists to facilitate return from the subroutine call. Luckily, the part of the LIST routine we are interested in is relocatable — the absolute address of the code is not referenced. So the routine can be "downloaded" into RAM where the necessary changes may be made.

First the loader program reads the top of memory pointer at $34, $35 and moves the pointer down to make room for 140 bytes. Then eight bytes are loaded which are responsible for setting up the pointer which starts the listing process at the start of BASIC. The pointer at $28, $29 (start of BASIC) is transferred to $5C, $5D (block transfer pointer #2):

```
A5 28   LDA  *TXTTAB
85 5C   STA  *LOWTR
A5 29   LDA  *TXTTAB+1
85 5D   STA  *LOWTR+1
/TXTTAB=0028      /LOWT=005C
```

Through PEEK and POKE statements, the required part of the LIST routine is downloaded — $C5DC-$C657 for BASIC 2.0 (upgrade ROM) and $B657-$B6DD for BASIC 4.0.

In order to provide the pause feature, the listing routine must be diverted after each line is printed. The patch is made by replacing the stop key test (JSR ISCNTC – 20 E1 FF) present in the transplanted ROM routine with a JSR to the pause subroutine. The loader program keeps track of where the new subroutine resides and replaces $FFE1 with the correct address (which depends on the top of memory pointer). The PAUSE subroutine is short:

```
20 E1 FF   PAUSE  JSR  ISCNTC
A5 98      WAIT   LDA  *SHFKEY
D0 FC             BNE  WAIT
60                RTS
/ISCNTC=FFE1      /SHFKEY=98
```

If SHFKEY contains a one, the SHIFT key is depressed; not depressed if zero. Finally, the loader program displays the SYS command which will produce the listing.

With just four bytes (A5 98 D0 FC) any machine language program may have a pause feature. This may also be added to a BASIC program with the following statement: WAIT 152,1,1. As an example, the previous statement placed as line 505 in DISK DISASSEMBLER will allow you to control the output with the SHIFT key.

I hope you will find this program useful. With a small amount of programming (16 bytes along with changing 2 bytes), it is possible to get a dramatic effect. Perhaps you will find a different application which requires moving a ROM routine into RAM. Please share it with us if you do.

```
100 REM LINELIST/2.0  G.H.WATSON 4/81
110 :
120 REM READ TOP OF MEMORY
130 M=PEEK(52)+256*PEEK(53)
140 :
150 REM SET NEW TOP OF MEMORY
160 M=M-140:X=M:GOSUB400
170 POKE 52,L:POKE 53,H
180 :
190 REM BEGIN AT START OF BASIC
200 FOR J=0TO7:READ A:POKE M+J,A:NEXT
210 :
220 REM DOWNLOAD ROM ROUTINE FOR LIST
230 FOR J=0TO123:A=PEEK(50652+J)
```

```
240 POKE M+J+8,A:NEXT
250 :
260 REM PATCH LINELIST TO ROUTINE
270 X=M+J+8:GOSUB400
280 POKE M+23,L:POKE M+24,H
290 FOR J=0TO7:READ A:POKE X+J,A:NEXT
300 :
310 PRINT"LINELIST:  SYS";M:END
320 :
400 H=INT(X/256):L=X-H*256:RETURN
410 :
500 DATA 165,40,133,92,165,41,133,93
510 DATA 32,225,255,165,152,208,252,96
```

For BASIC 4.0 change the following lines to:

```
160 M=M-151:X=M:GOSUB400

230 FOR J=0TO134:A=PEEK(46679+J)
```


Power On/Error Indicator For CBM Disks

Jim Butterfield
Toronto, Canada

This is a simple, inexpensive little board that you mount in your Commodore 2040 or 4040 disk unit. It does two things:

— The red error light on the disk unit becomes two-colour: red for error, green for power on;
— When an error occurs, the light turns to a flickering red, and there is a low-level audible alarm.

The first effect — the red/green light — is similar to that of the 8050 disk unit. It's very handy for reminding you that the power is on.

The second item — flickering light and audible alarm — is very useful, particularly to beginning disk users. Instead of having to remember to look at the disk unit to see if everything is going well, the unit will let you know that there's a problem.

The disk will continue to sound the alarm until the error condition is cleared, either by doing another operation or by interrogating the error status. I know one user who can't stand to have his disk yipping at him, and has disabled the sounder unit. He's an exception: the sound is quite low level, and most of us find that it's useful to have problems called to our attention.

Installation

Putting the Indicator into your 2040 is not difficult,

and requires no permanent changes to your disk unit. You can always put everything back exactly the way it was. The board mounts onto one of the disk boards by means of a screw which replaces an existing screw. You'll need to remove the old LED light — a matter of loosening the locking ring — and replace it with the new two-colour unit.

It can be done without technical expertise, but it's probably better to have someone on hand with a little experience. My disk is quite an early unit, and I found two small problems: the mounting screw wouldn't fit into its matching nut until I loosened the whole disk circuit board and positioned the nut properly; and the mounting screw didn't get a satisfactory ground from the board it was mounted on ... I had to run in a separate ground wire. Neither of these are serious problems, and I understand that they are unlikely to occur on more recent models of 2040 and 4040.

I had a little trouble with the locking ring, but that was my usual clumsiness. I dropped it on the floor and spent half an hour finding it in my shag carpet.

It's simple, inexpensive, and quite worthwhile. — Power-On/Error Indicator, available from **Canadian Micro Distributors Ltd.**, 365 Main St., Milton, Ontario Canada L9T 1P7. Price: $29.95. I understand that U.S. orders will be shipped from CMD's USA warehouse, so that purchasers won't need to worry about customs and duty. ©

SOFTWARE HARDWARE FOR THE VIC 20®

HARDWARE FOR THE VIC 20®

- 3 K Memory Expansion with addressable and switchable ROM slots
- Joysticks
- Light Pen
- Game Paddles
- RS232 Interface
- Home Controller Interface
- Add on Memory Boards

SOFTWARE FOR THE VIC 20®

- Games
- Educational
- Personal
- Professional
- Business

AND MUCH, MUCH MORE!!!!

DEALER INQUIRIES INVITED

Clip Coupon for **FREE CATALOG**
Catalog available in Spanish (Version Disponible en Español)

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 _____

United Microware Industries Incorporated
3431 H. Pomona Blvd., Pomona, California 91768
(714) 594-1351

VIC 20 is a Registered Trademark of Commodore Business Machines

THE SBC GAZETTE

An Efficient A/D Interface

Richard Olivo

We use AIM-65's for the acquisition and storage of transient analog data. Once digitized and stored, the data are played back to oscilloscopes or to chart recorders for permanent copies. The interface requirements are fairly demanding in terms of speed (a sample is taken every 100 microseconds) and sensitivity (the input signal is a bipolar voltage of about 50 millivolts). I have designed an efficient, general purpose analog/digital interface which connects to the user 6522 versatile interface adapter on the AIM, and could be used without modification with other 6522-based parallel ports.

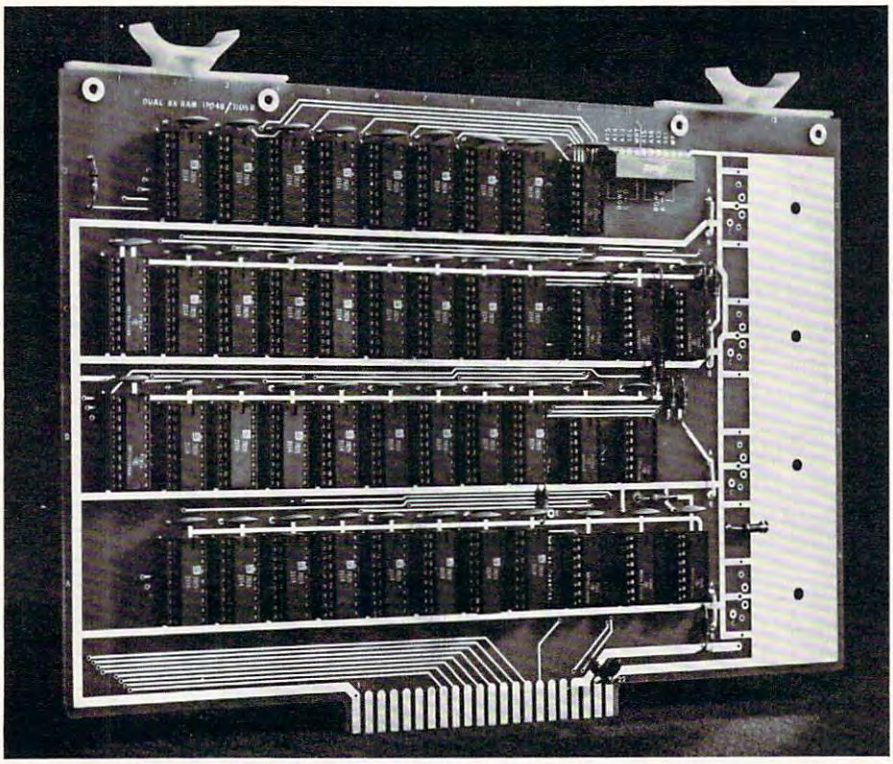
It is designed around two relatively inexpensive, high-performance integrated circuits from Analog Devices, Inc. (Box 280, Norwood, MA 02062). I chose these devices for their ease of use (they require no external parts), their speed, and their moderate cost. Analog-to-digital conversion is performed by the AD570 ($29), an 8-bit device with an input range of +5 to -5 volts and a 25-microsecond conversion time. Digital-to-analog output is provided by the 8-bit AD558 ($11), which has an output range of 0 to 10 volts and which requires about 2 microseconds to produce a full-scale change in the output voltage. The data lines of these two devices are connected to the 6522's ports A and B, with control lines CA1 and CA2 also used. Control lines CB1 and CB2 are free; we use them to receive trigger pulses (CB1) and to output sync pulses (CB2). I added circuitry to protect CB1 and CB2 against inadvertent application of non-TTL voltages, but this protection could be omitted. The interface also provides preamplification of the input with variable gain from 1 to 1000 (in 1,2,5 steps), a zero-offset adjustment, and a choice between direct- or capacitor-coupling for the input. The output connector is switchable between the preamplifier's output and the output of the digital-to-analog converter. The full interface circuitry is shown in Figure 1.

Although programs for data collection can become lengthy, the actual routines for analog input and output are brief. Output through the digital-to-analog converter is extremely simple, since the device is connected in its transparent (unlatched) mode. In this mode, a byte written to Port B appears immediately as a steady analog voltage at the converter. The AD570 input converter, on the other hand, requires control lines to start conversion and to detect that data are ready. In my circuit, control line CA2 initiates conversion by supplying a TTL pulse, which must last at least two microseconds. Conversion begins on the falling phase of the pulse, which also blanks the AD570's data lines. After 25 microseconds, the end of conversion and the appearance of new data are signalled by the $\overline{DR}$ line (pin 17 of the AD570, connected to control line CA1); this line goes low and remains low until the next conversion is initiated. Thus, a routine to acquire a byte of analog data must accomplish four tasks: it must initiate conversion by making CA2 go high and then low, it must update a storage vector to be used to place the new data in memory, it must examine CA1 to check if the new data are ready, and finally it must read Port A and store the data in memory. If the storage buffer is more than one page of memory (as in my system), the minimum practical sampling interval approaches 100 microseconds, even though the converter itself requires only 25 microseconds. The remainder of the time is taken up by the software.

In my program, data sampling is interrupt driven. Timer T1 on the AIM's 6522 versatile interface adapter is set to generate interrupts at 100-microsecond intervals and, at each interrupt, an interrupt service routine is executed to acquire one byte of data. Prior to enabling interrupts, however, a series of initialization steps must be carried out. The data-direction registers are set to make Port A an input and Port B an output; the auxiliary control register is set to place timer T1 in free-running mode with the output to PB7 disabled; the peripheral control register makes CA1 interrupt on a negative transition (used to indicate that data are ready); T1 is loaded with the two-byte value of the sampling interval, expressed in microseconds; the address of the interrupt service routine is placed in the AIM's IRQV2 (or whichever address the 6502 jumps through when an interrupt occurs); and the interrupt enable register is set up to allow timer T1 to interrupt. Details of how to use interrupts may be found in the AIM User's Guide, the 6500 Programming Manual, or in Leventhal's *6502 Assembly Language Programming* (Osborne/McGraw-Hill). In addition, if data are to be saved in memory, a storage index and a storage vector on page zero must be given their initial values prior to the first interrupt.

Once initialization is finished and the interrupt has been enabled, the microprocessor will jump to

VAK-2 8K STATIC RAM BOARD
VAK-4 16K STATIC RAM BOARD



We also carry:
SYM-1 $229.00
AIM-65 w/1K $389.00
AIM-65 w/4K $439.00
We also do custom hardware and software for the 6502 microprocessor

Special Price for Compute! readers on pre-paid orders from Sept 1 until Oct 30

PRICE:
VAK-4 ~~$325.00~~ $295.00
VAK-2 $239.00
Add $4.50 for UPS
Shipment in continental U.S.
All others call or write for shipping charges

The VAK-2/4 was specifically designed for use with the KIM-1, SYM-1 and the AIM 65 Microcomputer Systems. The VAK-4, 16K Ram Board, consists of two (2) separate 8K blocks. Each block has it's own address, write protect and block enable switches.

The VAK-2, 8K Ram Board, is identical to the VAK-4 with sockets for all 16K of Ram, but it has only one of the 8K blocks populated with IC's. Therefore, the VAK-2 is user expandable to a full 16K with the purchase of the VAK-3 Expansion Kit.

Both the VAK-2 and the VAK-4 Boards are made with 1st quality, Industry Standard 450 *nsec.* 2114 RAM Chips. They plug directly into the VAK-1 Motherboard, or with addition of voltage regulators plug into the KIM-4* Motherboard.

SPECIFICATIONS:

- Completely assembled, tested and burned-in.
- All IC's are in sockets
- Fully buffered address and data bus
- Standard KIM-4* Bus (both electrical Pin-out and card size)
- Designed for use with a regulated Power Supply such as our VAK-EPS, but has provisions for adding regulators for use with an unregulated Power Supply.
- Each 8K Block Address is independent and switch selectable.
- Separate write-protect switch for each 8K block.
- Board size: 10 in. Wide x 7 in. High (including card-edge)
- Power requirements: VAK-2—5V.DC @ 1.2 AMPS.
  Power requirements: VAK-4—5V.DC @ 2.4 AMPS.

**KIM-4 is a product of MOS Technology/C.B.M.*

We have moved to a new, larger facility. Please make note of our new address.

RNB ENTERPRISES
INCORPORATED

4030 N. 27th Avenue, Suite D
Phoenix, AZ 85017
(602) 265-7564

master charge VISA

the interrupt service routine every time T1 times out. The background task while interrupts are occurring is to scan the keyboard for single-character commands that disable the interrupts and cause jumps to playback or other data-collection modes. The interrupt service routine, shown in Program 1, first saves registers A and Y on the stack, and then executes the four input tasks that were described earlier. It assumes that T1 is the only possible source of interrupts, since it does not check the interrupt flag register for T1's flag, as would be done if there were other sources of interrupts. The program stores data in a multi-page ring buffer accessed through indirect indexed addressing. The storage vector is updated between the start of conversion and first checking for data ready, since this is time that would otherwise be wasted while waiting for conversion to finish. Part of updating the vector involves checking to see if the end of the buffer has been reached; if it has, the vector is reset to point to the beginning of the buffer. When the data are ready, the program reads Port A, stores the data in memory, and echoes the data to the output converter through Port B.

Source listing of the interrupt service routine. The routine assumes that timer T1 is the only source of interrupts.

```
INTSV1  PHA             ;PROTECT REGISTERS A & Y.
        TYA
        PHA
        LDA PCR         ;PULSE CA2 TO START A/D:
        ORA #%00001110
        STA PCR         ;FORCE CA2 HIGH
        AND #%11111101
        STA PCR         ;FORCE CA2 LOW
        LDY YVALST      ;GET STORAGE INDEX
        INY             ;UPDATE IT, &
        STY YVALST      ;SAVE INDEX VALUE
        BNE INTSV2      ;BRANCH IF Y NOT 0
        INC STVEC+1     ;ELSE UPDATE STVEC PAGE.
        LDA STVEC+1     ;CHECK NEW PAGE:
        CMP #MAXPAG     ;IS IT END OF BUFFER?
        BNE INTSV2      ;IF NOT END, CHECK A/D
        LDA #MINPAG     ;ELSE WRAP TO START
        STA STVEC+1     ;OF BUFFER.
INTSV2  LDA IFR         ;IS A/D DATA READY?
        AND #%00000010  ;ISOLATE CA1 FLAG
        BEQ INTSV2      ;LOOP IF CA1 STILL CLEAR,
        LDA PORTA       ;ELSE GET A/D DATA,
        STA (STVEC),Y   ;PUT IT IN MEMORY,
        STA PORTB       ;& ECHO IT TO D/A.
        LDA T1L         ;CLEAR T1 FLAG IN IFR
        PLA             ;RESTORE REGISTERS Y & A,
        TAY
        PLA
        RTI             ;AND RETURN ...
```

Program 1.

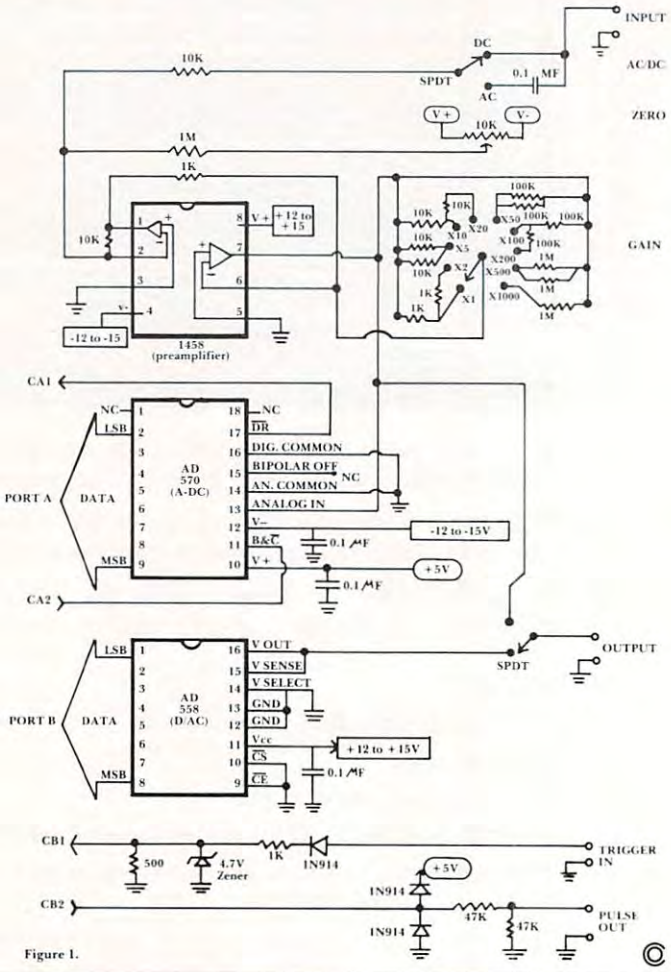


Figure 1.

A BRILLIANT FUTURE FOR YOUR AIM-65 WITH THE BANKER MEMORY ™

Your 36K of free address space is the AIM's most valuable and limited resource. With today's large capacity RAM boards, ROM boards, disk systems, video boards, and other expansion accessories it is easy to deplete this resource before the application requirement is satisfied. MTU has solved this problem.

THE BANKER MEMORY contains 32K of RAM, 4 PROM sockets for 2716/2732/2332, a PROM programmer, 40 bits of parallel I/O, and 4 timers from two 6522 I/O chips. Addressing is extremely flexible with the RAM independently addressable in 4K blocks, PROM's independently addressable, and I/O addressable anywhere on a 64 byte boundary (even in AIM's I/O area at AXXX by adding a single jumper to the AIM).

This may sound familiar, but read on! Unlike other AIM compatible memory boards, THE BANKER MEMORY has on-board bank-switching logic! The four 8K blocks of RAM plus the 4 PROM sockets make up 8 **resources**, each associated with a bit in an Enable Register. Through this Enable Register resources may be turned on and off under software control. When a resource is off, its address space is freed for other uses. You can even put BANKER resources at the same address and switch among them for virtually unlimited RAM and PROM expansion! You can even have multiple page zero's and stacks! Do you need 160K byte of memory? It only takes 5 of THE BANKER MEMORY boards and you end up with 5 page zeros and stacks to boot!

There's more! The BANKER MEMORY also incorporates 18 bit addressing which allows for the 256K address spaces of the future. RAM, PROM, and I/O each has its own full 18 bit address decoder which allows these resources to be in different 64K banks. This board and other MTU products, such as our 320 by 200 dot VISIBLE MEMORY and Floppy Disk Controller with 16K DMA RAM, can turn your AIM into a truly powerful 6502 computer that far surpasses the packaged systems in functional performance.

INTRODUCTORY SPECIAL K-1032-1 32K BANKER MEMORY FULLY ASSEMBLED AND TESTED $395.00 ($450.00 as of March 1, 1980) or the K-1032-2 16K RAM only with bank switching and 18 bit address bus only $295.00

Isn't it time you took a closer look at MTU — we offer you power now with an eye to the future.

WRITE OR CALL TODAY FOR OUR 48 PAGE FALL 1980 6502 CATALOG

International requests include $1.00

VISA and MASTERCARD accepted

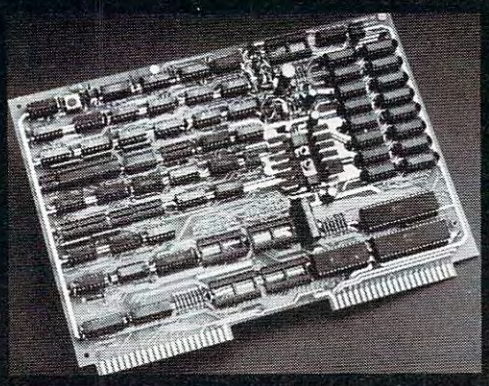


Photo credit:
SUPERNOVA CRAB NEBULA:
Palomar Observatory,
California Institute of Technology

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

New Products

Commodore Business Machines To Sponsor National Careers For The Disabled Symposium

King of Prussia, PA, Aug. 6, 1981 — The first National Careers for the Disabled Symposium, offering first-hand information to career-oriented disabled persons, will be sponsored by Commodore Business Machines, Inc., in association with Careers for Disabled, Inc., and be held at the Convention Center in Baltimore, Md., December 4-6, 1981.

The symposium will reach out to the many thousands of people who, because of their disabilities, and in some cases lack of skills, have been kept out of the mainstream of the work force and society.

Each workshop will include a special lecture on "how to" obtain training and then market new or existing skills in areas such as computer technology, starting your own business, continuing your education, sales, government and unions, finance, printing and the graphic arts, clerical, travel and leisure, food services, communications, and repair trades.

In announcing that Commodore has agreed to sponsor the symposium, James Finke, president and chief operating officer of the company, said, "Today, business leaders must assume leadership rolls in helping assimilate millions of handicapped people into the work force. It is our belief that their symposium will be an important first step in bringing about this reality.

Additional information on and reservation forms for the first National Careers for the Disabled Symposium are available by contacting Careers for the Disabled, 261 Madison Avenue, Suite 1102, New York, NY 10016.

Atari Video Computer System™ Service Network Formed

Sunnyvale, CA — Atari, Inc., announced the formation of a nationwide independent service network to provide convenient warranty service for the Atari Video Computer System (VCS™). The network, which will include 500 service locations by the end of 1981, will be composed of independent electronics retail and repair centers, trained and authorized by Atari to service the Atari VCS. In addition, the centers will become retail outlets for Atari VCS hand controllers: joysticks, paddles and keyboards.

"With the number of Atari VCS owners growing daily, we need more service centers in more places across the U.S.," said Michael J. Moone, president of Atari's Consumer Electronics Division. "By recruiting and training independent shops to service our product, Atari can better serve the more than 2.5 million Atari VCS owners in this country."

Previously, all VCS service had been performed by Atari's own regional service centers located in Somerset, New Jersey and Sunnyvale, California. These centers will continue to operate as support for the independent service network. They will be joined by two more Atari regional Centers in Chicago and Dallas later this year.

The Atari Video Computer System is a programmable home video entertainment unit that utilizes over 40 interchangeable game and educational cartridges. The VCS can be played on any home TV set.

Atari Memory Price Drop

Mosaic has announced that they are reducing the prices on two major memory products for the ATARI Computers.

The 8K to 16K RAM Expansion Kit will retail for $49.95, down from $79.00. The Mosaic 32K RAM will retail for $179.95, down from $219.95.

MOSAIC ELECTRONICS
P.O. Box 748
Oregon City, OR 97045

Family Oriented Software Developed

New York, NY — Apple Computer Inc. and the Children's Television Workshop, creators of "Sesame Street," have combined talents to design, develop and distribute 20 software programs for use on Apple personal computers.

The programs are an outgrowth of 50 educational games developed by the Children's Television Workshop (CTW), and were tested over the past year at the Sesame Place Computer Gallery, a component of the newly created "playground of the future" in Langhorne, Penn.

The Children's Computer workshop, a division of CTW and directed by Joyce Hakansson, decided to adapt the 50 games used at Sesame Place into 20 marketable programs to allow public access to these informal educational programs. Apple will publish the programs, then distribute them through its more than 1,000 U.S. computer retail dealers under the Special Delivery Software label.

"CTW used Apple computers at Sesame Place because of their 16-color graphics and sound capabilities. The new programs rely on these capabilities," said Hakansson.

"Developing the software programs is a natural extension of our interest in blending entertainment and education on television. This is our first step into electronic publishing, a field which we believe will make a significant impact on informal education," said CTW President Joan Ganz Cooney.

The games, entertaining in their use of sound, color and animation, are primarily aimed at the three-to-13-year-old age group. They are designed to acquaint children with computers.

Program concepts include tests of motor skills, reading, vocabulary, math, logic and problem solving and creative and artistic challenges.

"The programs have been thoroughly researched and their appeal has been proven through regular use by visitors at Sesame Place," said Mike Kane, marketing

commodore

EQUIPMENT:

CBM 4016 CPU (40 Col. Screen, 16K RAM)	839
CBM 4032 CPU (40 Col. Screen, 32K RAM)	1069
CBM 8032 CPU (80 Col. Screen, 32K RAM)	1249
CBM 4040 Dual Disk Drive (170K per 5¼ Diskette)	1049
CBM 8050 Dual Disk Drive (½ Meg. per 5¼ Diskette)	1449
CBM 4022 Tractor Feed Printer	659
CBM 8010 Phone Modem	239
CBM C2N Cassette Deck	85
CBM CPU/IEEE Cable	36
CBM IEEE/IEEE Cable	45
Tally 8024-7 (7x7 Matrix) Printer	1597
Tally 8024-9 (7x9 Matrix) Printer	1697

SOFTWARE:

Wordcraft 80 Wordprocessor	329
Wordpro 4 + Wordprocessor	329
OZZ Data Base	329
Tax Preparation System	419
Irma Information Retrieval	419
Dow Jones Portfolio	129
Pascal Development Package	249
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

COMING SOON:

CBM 8096
CBM Micro-Mainframe
Upgrade kit will be available to convert existing 8032's to 8096 or Micro MF
CBM 8250 Dual Disk Drive (1 Meg per 5¼ Diskette)
CBM 8061/8062 8 inch Disk Drives
CBM 2031 Single Disk Drive (170K per 5¼ Diskette)
Additional VIC Peripherals and Software

All items Insured
COD or Credit Card
Prepaid Orders Shipped Free
In stock items shipped within 48 hours

MASTERCHARGE OR VISA ACCEPTED

MART

P.O. Box 77286
Atlanta, Ga. 30357

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

Special Introductory Offer

VIC-20

$270.00
List $299.95

Price Includes Shipping

manager for Apple's Personal Computer Division.

"Apple is deeply involved in the education market and we insist that educational software for the home be of the same high quality as programs designed for classrooms," Kane added.

CompuMart Announces New Summer/Fall Catalog

CompuMart Corp. has just published its new summer/fall microcomputer catalog.

The 48-page catalog features pricing and technical information for microcomputers and microcomputer systems from Apple, Commodore, Hewlett-Packard, Rockwell International, Texas Instruments, and other manufacturers. Books, accessories, and peripherals (including 13 printers from 5 manufacturers) are listed.

To obtain your free copy of this catalog, write to CompuMart Corp., P.O. Box 568, Dept. 004, Cambridge, MA 02139.

BRAIN BOX Courseware Kit™ For Pet/Apple

BRAIN BOX — The Computer Tutor announces the release of 200 educational programs on 30 titles. These are available on floppy disks and cassettes for Apple and PET microcomputers.

BRAIN BOX programs are created by a staff of professional teachers and programmers. Each title is accompanied by extensive informative documentation, including a complete Teacher Guide designed to integrate courseware into the curriculum. The BRAIN BOX COURSEWARE KIT™, accompanying each title, permanently and conveniently houses the disk or cassettes and contains a glossary of commonly used computer terms, an Overview of Microcomputer hardware and functions, a Primer for BRAIN BOX programs and a Pronunciation Guide to the BRAIN BOX adaptation of the phonetic alphabet. Each lesson combines text with innovative, animated graphics, plus sound effects on Apple.

BRAIN BOX has a unique multiple-copy plan, designed to fully utilize a school's computer complex at discount prices. For each title package ordered at the regular price, up to 5 additional copies of the cassette or diskette may be purchased for 60% off the regular price.

A demonstration disk with a variety of lessons from the BRAIN BOX catalog is available for $30, which will be credited against the purchase price of any 2 or more titles.

For your free catalog, write BRAIN BOX, 601 West 26 Street, New York, NY 10001 or call 212/989-3573.

New Axlon* Memory System For Atari* 800 Announced

Sunnyvale, CA — Axlon Incorporated of Sunnyvale announced the release of its 128K memory system for the Atari 800, making it a powerful personal computer.

According to John Vurich, Axlon's president, the system, called the RAMDISK™ Memory System, comes with software that makes the new system function like a disk device. The system can also be programmed as bank selectable RAM memory.

The RAMDISK Memory System, when utilized as an additional disk device in conjunction with an Atari 810* Disk Drive, is compatible with existing software written for the Atari 800 system. Function for function, the RAMDISK system is up to 20 times faster than the Atari 810.

The RAMDISK Memory System can also be utilized as bank selectable RAM memory. The system is organized into

we beat the price...

800™ $779

ATARI® Computers for people.™

815 DUAL DISK DRIVE

$1199

WITH ONE YEAR EXTENDED WARRANTY

ATARI SOFTWARE

CX401 General Accounting	$399.00
CX402 Accounts Receivable	$399.00
CX403 Inventory Control	$399.00
CX404 Word Processor	$119.00
CX405 PILOT	$68.00
CX413 MICROSOFT BASIC	$68.00
CX4101 Invitation to Programming 1	$17.00
CX4102 Kingdom	$13.00
CX4103 Statistics	$17.00
CX4104 Mailing List	$17.00
CX4105 Blackjack	$13.00
CX4106 Invitation to Programming 2	$20.00
CX4107 Biorhythm	$13.00
CX4108 Hangman	$13.00
CX4109 Graph It	$17.00
CX4110 Touch Typing	$20.00
CX4111 Space Invaders	$17.00
CX4112 States & Capitals	$13.00
CX4114 European Countries & Capitals	$13.00
CX4115 Mortgage & Loan Analysis	$13.00
CX4116 Personal Fitness Prog	$59.00
CX4117 Invitation to Programming 3	$20.00
CX4118 Conversational French	$45.00
CX4119 Conversational German	$45.00
CX4120 Conversational Spanish	$45.00
CX4121 Energy Czar	$13.00
CX4125 Conversational Italian	$45.00

CX6001 U.S. History	$23.00
CX6002 U.S. Government	$23.00
CX6003 Supervisory Skills	$23.00
CX6004 World History	$23.00
CX6005 Basic Sociology	$23.00
CX6606 Counseling Proced	$23.00
CX6607 Principal of Act	$23.00
CX6608 Physics	$23.00
CX6609 Great Classics	$23.00
CX6010 Business Comm	$23.00
CX6011 Basic Psychology	$23.00
CX6012 Effective Writing	$23.00
CX6014 Principals of Econ	$23.00
CX6015 Spelling	$23.00
CX6016 Basic Electricity	$23.00
CX6017 Basic Algebra	$23.00
CX8106 Bond Analysis	$20.00
CX8107 Stock Analysis	$20.00
CX8108 Stock Charting	$20.00
CXL4001 Education System Master	$21.00
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-Tow	$24.00

CXL4011 Star Raiders	**$32.00**
CXL4012 Missle Command	**$32.00**
CXL4013 Asteroids	**$32.00**
CXL4015 TeleLink	$20.00
Compuhome	$74.95
Visicalc	$149.00
Letter Perfect (Word Processor)	$119.00
Source	$89.00

Atari® Peripherals:

400 16K	$329.00
410 Recorder	$59.00
810 Disk Drive	$449.00
815 Disk Drive	$1199.00
822 Printer	$359.00
825 Printer	$629.00
830 Modem	$159.00
850 Interface	$139.00

Atari® Accessories

New DOS 2 System	$21.00
CX70 Light Pen	$64.00
CX30 Paddle	$18.00
CX40 Joy Stick	$18.00
CX853 16K RAM	$89.00
Microtek 16K RAM	$75.00
Microtek 32K RAM	$169.00

commodore

CBM 8032 $1099

4016	$799.00
4032	$999.99
8096	$1795.00
CBM4022 Printer	$629.00
Tally 8024	$1699.00
CBM C2N Cassette Drive	$69.00
CBM4040 Dual Disk Drive	$999.00
CBM8050 Dual Disk Drive	$1349.00

VIC 20 $259

Vic-TV Modual	$19.00
Vic Cassette	$69.00
Vic Disk Drive	$ Call
Vic 6 Pack program	$44.00

CBM Software

WordPro3 Plus	$199.00
WordPro4 Plus	$299.00
Commodore Tax Package	$399.00
Visicalc	$149.00
EBS Accts. Rec./Inventory Interactive Syst.	$595.00
BPI General Ledger	$329.00
OZZ Information System	$329.00
Dow Jones Portfolio	$129.00
Pascal	$239.00
Legal Time Accounting	$449.00
World Craft 80	$289.00
Word Check	$180.00
Create-A-Base	$180.00
Power	$89.00
Socket-2-Me	$20.00
Jinsam	Call

Disks

CX8100 Blank Disk (5)	$22.00
Sycom Blank Disk (10)	$29.00
Maxell Blank Disk (10)	$36.00
Maxell Blank Disk (10)	$46.00

Printers

Epson MX-70 Epson MX-80 Epson MX-80 FT Diablo 630	Call for Prices
Nec 5530	$2495.00
TEC 1500 Starwriter 25cps	$1495.00
TEC 1500 Starwriter 45cps	$1995.00

No Risk, No Deposit On Phone Orders, COD or Credit Card, Shipped Same Day You Call *

* on all in stock units

IN PA. CALL 717-327-9575 **(800) 233-8950**

COMPUTER MAIL ORDER
501 E. 3RD ST., WILLIAMSPORT, PA 17701

To Order:

Phone orders invited (800 number is for order desk only). Or send check or money order and receive free shipping. Pennsylvania residents add 6% sales tax. Add 3% for Visa or M.C. Equipment is subject to price change and availability without notice. Please call between 11 AM & 6 PM.

eight (8) 16K pages which can be selected under program control. Larger and more complex programming applications are now feasible utilizing bank selection.

The RAMDISK Memory System is the first memory product for the personal computer market using 64K RAM chips. Until now a product such as RAMDISK was not feasible because of the high cost of the new 64K RAM chips.

The RAMDISK module is installed in the second RAM slot in the Atari 800 with 16K RAM modules in the first and third RAM slots providing 160K bytes of RAM memory. Installation is accomplished in a matter of minutes and requires no modifications to the Atari 800 computer.

The RAMDISK Memory System includes the 128K RAMDISK module, operating manual, DOS Memory Management Software and utility software. The RAMDISK Memory System is available at Atari dealerships nationwide. The suggested retail price of the system is $699.00.

* Indicates Trademark of Atari Incorporated
™ Indicates Trademark of Axlon Incorporated

Market Information Now Provided To CompuServe

Columbus, Ohio — Archer Commodities, Inc. is now a provider of commodity market information on the CompuServe Information Service.

Archer Commodities is a commodity futures brokerage firm located in Chicago. A wholly owned subsidiary of Heinold Commodities, Archer Commodities provides its customers with market information, experience and needed personal attention in helping them trade commodities successfully.

Now through the CompuServe Information Service, owners of personal computers and computer terminals can receive current market reports and commentary, educational and informational material on futures trading and special announcements from Archer Commodities. Subscriber requests for charts, market newsletters and quotation equipment are handled by Archer on an individual basis through an interactive feedback procedure.

CompuServe subscribers can access information from Archer Commodities for the standard fee of $5 per hour weekday evenings, all day weekends and holidays. Weekday daytime access is also available. The equipment needed would be a personal computer or computer terminal, a telephone and a modem.

CompuServe also offers the latest news from major newspapers and a nationwide wire service, corporate stock information, home banking, electronic mail and real-time communications, computer games, family information and computing power for programming activities.

WHAT'S NEW?

Consumer Computers announces it's lowest prices ever.

Apple Add-Ons

Language System w/Pascal	379
Hayes Micromodem II	299
Novation Apple-Cat II	339
Videx Videoterm 80 w/graphics	269
Z-80 Microsoft Card	269
16K Ram Card Microsoft	159
ABT Numeric Keypad (old or new kybrd)	110
ALF 3 Voice Music Card	239
Heuristics Speechlink 2000	249
Alpha Syntauri Keyboard System	1399
Corvus 10 MB Hard Disk	CALL
Lazer Lower Case Plus	50
Micro-Sci Disk Drives (A40 & A70)	CALL
SSM AIO Serial/Parallel Card A&T	189
ThunderClock Plus	CALL
Integer or Applesoft II Firmware Card	145
Graphics Tablet	619
Parallel Printer Interface Card	135
Hi-Speed Serial Interface Card	135
Smarterm 80 Column Video Card	299
MusicSystem (16 voices)	479
A/D + D/A Interface	289
Clock/Calendar Card	225
Supertalker SD-200	239
Romplus + Card	135
Clock/Calendar Module (CCS)	99
Asynchronous Serial Interface Card (CCS)	129
We carry all CCS hardware	Please Call

FREE CATALOG!

Please mail us your name and address.

Solution Software for the Apple II.

Visicalc 3.3	169
CCA Data Management	85
DB Master	169
WordStar (Apple 80 col. ver.)	299
Desktop Plan II	169
Applewriter	65
Easywriter	225
Appleplot	60
Peachtree Business	1200
VisiTerm	129
VisiTrend/VisiPlot	219
DOS Toolkit	65
Dow Jones Portfolio Evaluator	45
Dow Jones News & Quotes Reporter	85
Apple Fortran	165

Atari Personal Computer

ATARI 800 16K $749

Atari Accessories

410 Program Recorder	59
810 Disk Drive	449
850 Interface Module	159
16K Ram Memory Module	89

Please call us for an Atari Software List.

S-100 Cards

	Kit	Assm & Tested
SSM		
Z-80 CPU CB-2	219	279
I04 2 P + 2S I/O	189	249
VB-3 80 x 24 VIDEO	329	379
VB-3 80 x 48 VIDEO	369	429
BV-2 VIDEO	159	219
PB1 PROM PRGMMR & EPROM BOARD	139	199
CCS		
Z80 CPU 2810	n/a	229
64K Dynamic RAM card	n/a	499
16K Static RAM 2 MHz	n/a	249
16K Static RAM 4 MHz	n/a	269
Dbl. Density Floppy Disc Controller	n/a	299
12 Slot S-100 Mainframe	n/a	349

Apple II Plus Computer

16K $1025 48K $1129

Disk II Drive w/controller **$519**

Disk drive w/out controller 439

We carry the Apple III

CALL FREE!

800-854-6654

In California and outside continental U.S.

(714) 698-8088

Warehouse (714) 698-0260
Service (714) 460-6502
TELEX 695000 BETA CCMO

AVAILABLE NOW . . .

The NEC Microcomputer

Please call for more details.

Exidy Sorcerer II

Please call for our price.

S-100 Unit	449
Word Processing Pac	179
Development Pac	89

PMC-80 Micro Computer

$599

Ohio Scientific Challenger Computer.

4P **$549**

1P MOD II 399

Printers, Etc.

Epson MX-80

$499

Silentype w/Apple Interface	349
Epson MX-80 F/T	649
Epson MX-100	829
Epson MX-70	339
Paper Tiger IDS-445	729
Paper Tiger IDS-460	1099
Paper Tiger IDS-560	1450
Qume Sprint 5/45	2495
Anadex DP-9500/ w/2K Buffer	1349
C. Itoh Starwriter 25 CPS	1649
C. Itoh Starwriter 45 CPS	2249
Centronics 737	699
Watanabe DigiPlot	1149

Save on Video displays.

Amdek/Leedex Video 100 12" B&W	139
Amdek (Hitachi) 13" Color	359
NEC 12" P31 Green Phospher	CALL
NEC 12" RGB Hi-Res Color	CALL
Panasonic 13" Color	449
Sanyo 9" B&W	159
Sanyo 12" B&W	239
Sanyo 12" P31 Green Phospher	279
Sanyo 13" Color	419

We carry much more than listed. Please call our toll free order line to request our complete price list.

How to Order

Ordering information: Phone orders using VISA, MASTER-CARD, AMERICAN EXPRESS, DINER'S CLUB, CARTE BLANCHE, bank wire transfer, cashiers or certified check, money order, or personal check (allow ten days to clear.) Unless prepaid with cash, please add 5% for shipping, handling and insurance. (minimum $5.00). California residents add 6% sales tax. We accept CODs, OEM's, Institutions and corporations please send for a written quotation. All equipment is subject to price change and availability without notice. All equipment is new and complete with manufacturer's warranty (usually 90 days). Showroom prices may differ from mail order prices.

Send orders to:

consumer computers

Mail Order

8314 Parkway Drive
La Mesa, California 92041

Math And Engineering Aids

Software is now available for performing higher mathematics and engineering of a general nature on the PET/CBM computers without extensive programming. The user merely programs his functions, which may be of any length from simple statements to complicated routines. These programs then integrate between inputted selected limits, to generate definite integrals, indefinite (functional) integrals, line integrals (plane or space curves), double or triple integrals, surface or volume integrals, or convolutions. Other programs do curve fitting to generate the best curve of the user-selected type to fit user-inputted data, analyze functions for derivatives, maxima, minima, points of inflection, roots and values, generate functions with specified real or complex roots, etc.

Another program plots the user's functions or data sets, with axis labels, automatic or selected scaling and range, optional grids, with either data points or connecting lines or both plotted. Plot is on screen or printer. Data sets may be saved on cassette or disk, and/or generated by user's own program with subroutine provided.

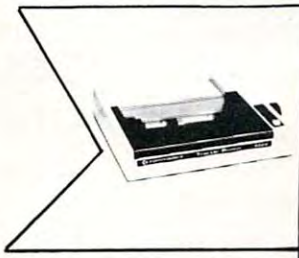
Another program allows very rapid writing of complicated or simple many-variable programs without syntax concerns. Independent variables have input-changeable default values. Data sets so generated may be printed and/or saved on cassettes or disk.

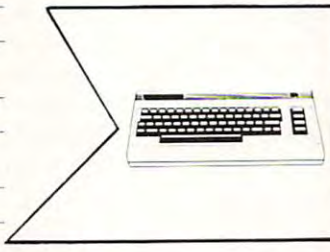
All programs internally documented, versatile, accurate, thoroughly tested, and easy to use. Prices range from $9.95 to $19.95.

James H. Burkhalter
2404 Seabreeze Court
Orlando, FL 32805
(305)843-8310
(305)589-7482

commodore summer spectacular

8032-32K 80 COL CRT	
REG. $1495	$1195
4032-32K 40 COL CRT	
REG. $1295	$995
4016-16K 40 COL CRT	
REG. $995	$795
8050-DUAL DISK 950K	
REG. $1795	$1395
4040-DUAL DISK 343K	
REG. $1295	$995
C2N-CASSETTE DRIVE	
REG. $95	$79
4022-80 COL PRINTER	
REG. $795	$649
8024-MANNSMAN TALLEY	
REG. $1995	$1595
25CPS-STARWRITER	
REG. $1895	$1445
CBM-IEEE MODEM	
REG. $279	$229
VOICE SYNTHESIZER	
REG. $395	$329
VIC 20	★★★★
REG. $299	$269
PET TO IEEE CABLE	
REG. $39.95	$34
IEEE TO IEEE CABLE	
REG. $49.95	$39





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

CUSTOM PRINTED CONTINUOUS MINI-FORMS

- LOOK PROFESSIONAL
- LOW PRICES FOR SMALL RUNS

1000-11"x9½" BLACK

PARTS	LETTERHEAD	INVOICE	STATEMENT
1	37.50	—	37.50
2		63.50	63.50
3		79.50	
5		115.50	

- SPECIAL LOGOS, 2 COLOR PRINTING & SPECIAL STOCK AVAILABLE

MINI-FORMS PRICES & ORDERS:
(800) 523-3442
PA & West of Rockies (215) 455-9705

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

Main Line Computer Discount

512 W. LANCASTER AVE.
WAYNE, PA 19087
215-687-8502

MAIL ORDER PHONE
800-345-8112
IN PA CALL 1-800-662-2444

32K Byte RAM/ROM Expansion

PROTRONICS has announced a new 32K Dynamic Random Access Memory (DRAM) board for the PET/SYM/KIM/AIM-65.

The 32K DRAM board contains sixteen RAM chips and five 24-Pin PROM sockets. The PROM sockets are for 2716/2516 2K type PROMs or 2732/2532 4K type PROMs (PROMS not included). This enables a total on-board memory expansion of 52K Bytes possible.

The Board also uses a new type of 16K DRAM chip which is pin compatible with the new 64K DRAM chips. This enables future on-board expansion to 128K of RAM (bank selectable, of course).

This Board has direct compatibility with the KIM-4 Buss which is used on most expansion chassis, such as those made by HDE, SEAWELL and other manufacturers.

It can connect directly to your computer with an adaptor cable which is available from PROTRONICS.

Addressing circuitry on board allows RAM to be configured in a contiguous 32K block placed above or below 8000 HEX. 4K blocks of RAM can be disabled using a Dip shunt (supplied) or a Dip switch. ROM configuration circuitry allows five 2K or 4K type PROMs to be located anywhere in the memory map. All configuration is done with short jumper plug wires (supplied) or with wire wrapping techniques.

Both the RAM & ROM are fully buffered to the data buss.

On-board circuitry synchronizes to the processor's clock to enable the refresh period to occur during cycles that are unused by the processor.

The Board requires only +5 Volts at 550 mA typical (without PROMs), which is supplied from the host computer through the expansion connector. There are no on-board voltage generators to go bad.

It all fits on a 6x4.5" board, allowing construction of a two board computer system. (Board 1: CPU, I/O. Board 2: RAM, ROM).

PROTRONICS recommends that if you already have 4K or 8K of static RAM and wish to keep it in your system, reconfigure the static RAM above 8000 HEX. (Most computers have this capability. Refer to your computer's reference manual.)

The 32K DRAM Board documentation includes a section for constructing your own power supply. This optional project will supply +12, +5, -5 or -12 volts for under $30 (including cabinet). (The 32K DRAM Board requires only +5 Volts to operate.)

Suggested retail price is $289.88 (U.S. Prices), for more information write or call: PROTRONICS, 1516 E. Tropicana STE 7A, Las Vegas, NV 89109, (702) 361-6331.

Expanded Game Line

Sunnyvale, CA —
"Monty Plays the Scrabble Brand Crossword Game" is a new member of the Personal Software Strategy Game line of programs for personal computers. The new program joins the "Monty Plays Monopoly" program. Both Monty programs were written by Ritam corporation and are computerized opponents, rather than computer reproductions of the games themselves.

"Monty Plays the Scrabble Brand Crossword Game" plays the Selchow and Righter board game with up to three human opponents. A Scrabble game set is required.

In operation, the Monty program turns a personal computer into a skilled game player with a vocabulary of over 54,000 words on diskette.

Graphic presentation of the Monty character, board positions

and scores, plus sound effects, heighten involvement with the program. For example, the four levels of difficulty which the program allows the user to choose are illustrated by Monty's increasingly academic garb, from a simple sport shirt to collegiate cap-and-gown.

Monty is available for use in 48K byte Apple II and II Plus computers with Applesoft Basic and one disk drive. Suggested retail price for the Monty game is $34.95.

Note: Scrabble is a registered trademark of Selchow & Righter; Monopoly is a registered trademark of Parker Bros.; Apple II, II Plus and Applesoft are trademarks of Apple Computer Inc.; Monty is a trademark of Ritam Corporation. Personal Software and Strategy Game are trademarks of Personal Software Inc. Monty is not sponsored or endorsed by Parker Bros. or Selchow and Righter.

Spelling For Very Young Learners

In July Edu-Ware Services, Inc. releases a new product under the EDU-WARE label, SPELLING BEE. Designed for children in kindergarten through third grade, SPELLING BEE meets three objectives: 1) develope computer literacy, that is, allow new learners to interact comfortably with the computer; 2) link the abstract verbal symbol (word) to the concrete (picture); 3) build basic spelling skills while identifying specific groupings (i.e. consonants, vowels, two- and three-letter words).

SPELLING BEE's high resolution graphics and musical sound effects appeal to capture a child's interest. Documentation speaks directly to parents and teachers, while EDU-Ware's Illustrated Children's Guide introduces youngsters to computer operation.

A System Generator allows the parent or teacher to tailor this system's length and emphasis to the individual child's needs. SPELLING BEE's demonstration mode allows the parent to review all spelling words before the child begins. As the child works within the tutorial, the system ignores incorrect responses. Only correct responses are reinforced. Then a drill mode tests the learner's performance, recording this information for the parent or teacher.

The system requires Applesoft, 48K, and DOS 3.3. It retails for $29.95.

For further information contact EDU-WARE Services, Inc., 22222 Sherman Way, Suite 203, Canoga Park, CA 91303 (213) 346-6783.

New Product releases are selected from submissions for reasons of timeliness, available space, and general interest to our readers. We regret that we are unable to select all new product submissions for publication. Readers should be aware that we present here some edited version of material submitted by vendors and are unable to vouch for its accuracy at time of publication.

STOP

BEING OVERCHARGED!

DON'T

Pay Credit Card Surcharges
Pay a U.P.S. C.O.D. "Fee"
Even Pay for the Call!

DO CALL OMEGA TOLL FREE!

Special of the Month!

EPSON MX-80

INTERFACES: IEEE $55, TRS-80 $35,
APPLE INTERFACE & CABLE $90,
RS-232 $70

EPSON MX-80FT $599
MX-100FT $769

WEST COAST
1-800-235-3581

EAST COAST
1-800-556-7586

CALL OMEGA AND FIND OUT WHY WE ARE AMERICA'S FASTEST GROWING COMPUTER SALES COMPANY!

ΩMEGA SALES COMPANY

WEST COAST / EAST COAST

ATARI 800 32K $ 749

Tractor option
$ 245

NEC 12" MONITOR
$ 229

NEC
OKIDATA
APPLE
TELEVIDEO
COMMODORE
RADIO SHAC
AMDEK
ATARI

DIABLO 630	$2099
INTERTEC SUPERBRAIN 64K RAM	$2799
QD SUPERBRAIN	$3195
NEC 5510 SPINWRITER	$2495
NEC 5530 SPINWRITER	$2495
OKIDATA MICROLINE-80	$ 399
OKIDATA MICROLINE-82	$ 529
OKIDATA MICROLINE-83	$ 769
APPLE II PLUS 48K	$1189
APPLE DISK w/3.3 DOS Controller	$ 545
APPLE DISK w/o Controller	$ 435
BASE II PRINTER	$ 599
HAZELTINE 1420	$ 799
NORTHSTAR HORIZON II 32K QD	$2975
ANADEX DP-9500	$1295
TELEVIDEO 912C	$ 669
TELEVIDEO 920C	$ 729
TELEVIDEO 950	$ 959
CBM 8032 COMPUTER	$1225
CBM 8050 DISK DRIVE	$1449
CBM 4032 COMPUTER	$1090
CBM 4040 DISK DRIVE	$1090
CBM 4022	$ 679
CBM VIC-20	$ 289
CBM C2N	$ 85
RADIO SHACK II 64K	$3245
RADIO SHACK III 16K	$ 839
LEEDEX/AMDEK 100	$ 139
LEEDEX/AMDEK 100G	$ 169
LEEDEX/AMDEK COLOR-1 13" Color Monitor	$ 349
MICROTEK 16K RAMBOARD for Atari	$99.95
MICROTEK 32K	$ 165
ATARI 400 16K	$ 349
ATARI 825 PRINTER	$ 619
ATARI 850 INTERFACE	$ 139
or both together	$ 749
ATARI 810 DISK DRIVE	$ 449
ATARI 800 32K	$ 749
Call for price list of ATARI software	
NEC 12" MONITOR	$ 229

PRICES ARE SUBJECT TO CHANGE WITHOUT NOTICE.

No Surcharge for Credit Cards Orders • We Accept C.O.D.'s
All Equipment Factory Fresh With MFT. Warranty • Stock Shipments Same Day or Next

WEST COAST
1-800-235-3581

OMEGA SALES CO.
3533 Old Conejo Rd. #102
Newbury Park, CA 91320
1-805-499-3678
CA. TOLL FREE 1-800-322-1873

ΩMEGA SALES CO.

EAST COAST
1-800-556-7586

OMEGA SALES CO.
12 Meeting St.
Cumberland, RI 02864
1-401-722-1027

ΩMEGA SALES COMPANY

Advertiser's Index

The 6502 Resource Magazine

COMPUTE! PET•ATARI•APPLE OSI•KIM•SYM•AIM

For Fastest Service,
Call Our **Toll-Free**
US Order Line
800-227-1617, Ext. 401
In California call 800-772-3545, Ext. 401

My computer is:
☐ **PET** ☐ **APPLE** ☐ **ATARI** ☐ **OSI**
☐ **KIM** ☐ **SYM** ☐ **AIM** ☐ **OTHER**_____
☐ **Don't yet have one...**

☐ Please enter my 1 year (12 issue) subscription to **COMPUTE!**
☐ New subscription ☐ Renewal subscription
☐ $20.00 U.S. subscription
☐ $25.00 Canada and international surface mail. Payment in U.S. funds, drawn on U.S. bank or money order
☐ Bill me (U.S. only) $100 billing fee

Name, Address: _____

_____

_____

_____

Charge my: ☐ Visa ☐ MC
Number_____ Expires_____/_____/

9 10 11 12

The Editor's Feedback:

My computer is:
☐ **PET** ☐ **APPLE** ☐ **ATARI** ☐ **OSI**
☐ **KIM** ☐ **SYM** ☐ **AIM** ☐ **OTHER**_____
☐ **Don't yet have one...**

During the next year I expect to buy:
☐ **computer** ☐ **printer**
☐ **disk drive** ☐ **other peripherals**

Content
Best Article This Issue (page #, title)

_____

Other suggestions

9 10 11 12

The 6502 Resource Magazine

COMPUTE! PET•ATARI•APPLE OSI•KIM•SYM•AIM

For Fastest Service,
Call Our **Toll-Free**
US Order Line
800-227-1617, Ext. 401
In California call 800-772-3545, Ext. 401

My computer is:
☐ **PET** ☐ **APPLE** ☐ **ATARI** ☐ **OSI**
☐ **KIM** ☐ **SYM** ☐ **AIM** ☐ **OTHER**_____
☐ **Don't yet have one...**

☐ Please enter my 1 year (12 issue) subscription to **COMPUTE!**
☐ New subscription ☐ Renewal subscription
☐ $20.00 U.S. subscription
☐ $25.00 Canada and international surface mail. Payment in U.S. funds, drawn on U.S. bank or money order
☐ Bill me (U.S. only) $100 billing fee

Name, Address: _____

_____

_____

_____

Charge my: ☐ Visa ☐ MC
Number_____ Expires_____/_____/

9 10 11 12

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 2312 GREENSBORO, NC

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE! Magazine
Post Office Box 5406
Greensboro, NC 27403

Place
Stamp
Here

COMPUTE! Magazine
Post Office Box 5406
Greensboro, NC 27403

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 2312 GREENSBORO, NC

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE! Magazine
Post Office Box 5406
Greensboro, NC 27403

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS / PERMIT NO. 178 / DELRAN, NJ

POSTAGE WILL BE PAID BY ADDRESSEE

datapro

DATAPRO RESEARCH CORPORATION
1805 Underwood Boulevard
Delran, New Jersey 08075

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 2312 GREENSBORO, NC

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE! Magazine
Post Office Box 5406
Greensboro, NC 27403

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS / PERMIT NO. 178 / DELRAN, NJ

POSTAGE WILL BE PAID BY ADDRESSEE

datapro

DATAPRO RESEARCH CORPORATION
1805 Underwood Boulevard
Delran, New Jersey 08075

BEFORE YOU BUY A FINE PRINTER, READ THE FINE PRINT.

That's where you'll find the prices—in the tiny type under the brief descriptions.

Which is only appropriate. Because with C. Itoh's full line—from the workhorse business printers to the racehorse word-processing printers—the descriptions could be much bigger.

But the prices couldn't be much smaller.

Moreover, as part of the Leading Edge Inventory Bank,™ our printers (along with dozens of other popular computer products) are available for 24-hour "withdrawal." Within a day of when we receive your order, your order will be on its way from our nearest warehouse (Boston, New York, Los Angeles, etc.), thus saving you both time and freight. So you can use our space, and our money, to inventory your product, and still be sure of having exactly what you need, exactly when you need it.

All the printers on this page are warranteed by Leading Edge™ for 3 months on parts and labor.

For the name of your nearest dealer—just pick up the nearest telephone.

TOLL-FREE 1-800-343-6833.

In Massachusetts, call collect (617) 828-8150.

Strictly business (Comet I)

The industry workhorse of dot-matrix printers. No frills; but almost nothing to go out of whack. 10" carriage; 80 columns (132 in compressed print mode). Industry-standard (teletype) ribbons. Retail: about $495.

Big business (Comet II)

The first 15" printer to retail for under a grand. Full 132 columns for standard-width accounting reports, giving you easy-to-read 10 cpi printout up to 66 lines deep. And, like with the Comet I, ribbons are available everywhere. Retail: about $995.

Beyond business (Starwriter I)

Lowest-priced letter-quality printer on the market. Comes complete and ready to use, requiring no changes in hardware or software. Industry-standard daisy wheels and ribbon cartridges. Plug-in compatible with a wide variety of systems. 25 cps; 163 columns; precise character placement; full graphics capability. Retail: about $1,895.

Busy business (Starwriter II)

The racehorse of letter-quality daisy wheel printers. Typical 45 cps throughput is about equal to (or, in bi-directional mode, actually *faster* than) "thimble" printers rated at 55 cps. Like the Starwriter I, it uses easily available daisy wheels and ribbon cartridges, and it's compatible with software packages, like Wordstar and Magic Wand, programmed for systems like the Qume Sprint 5. Retail: about $2,395.

LEADING EDGE™

Leading Edge Products, Inc., 225 Turnpike Street, Canton, Massachusetts 02021.

DEALERS: For immediate delivery from the Leading Edge Inventory Bank,™ call toll-free, **1-800-343-6833.**
In Massachusetts, call collect (617) 828-8150, Telex 951-624.

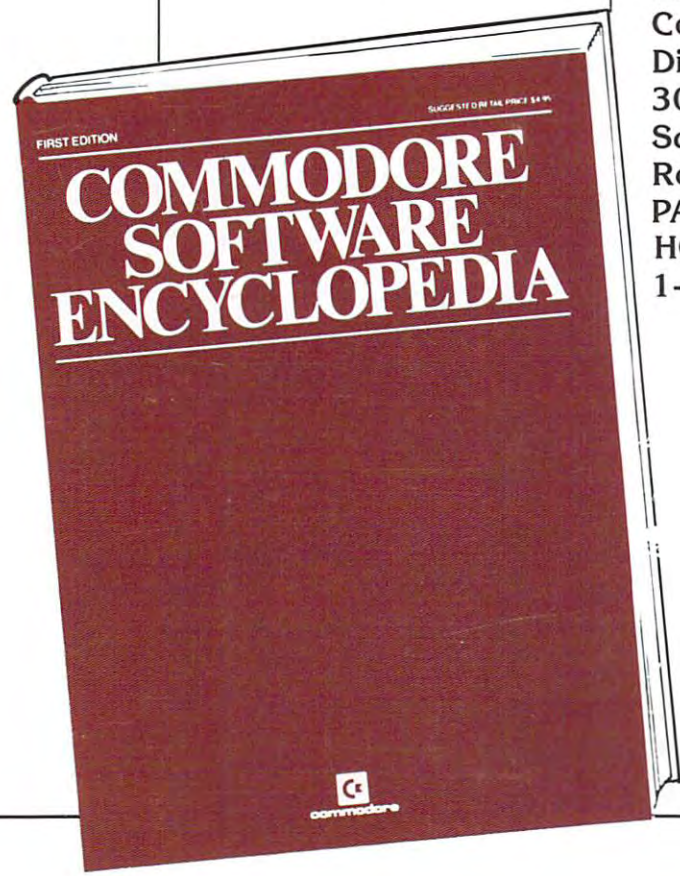
Commodore Software Encyclopedia

The COMMODORE SOFTWARE ENCYCLOPEDIA is now available from your authorized COMMODORE Dealer, for . . .

$4.95

The next edition of the COMMODORE SOFTWARE ENCYCLOPEDIA will be available in 90 to 120 days. If your software is not listed and you would like to have it listed in the next edition, please submit details to:

Software Department
Commodore Business Machines, Inc.
Computer Systems Division
300 Valley Forge Square, 681 Moore Road, King of Prussia, PA 19406.
HOTLINE Number
1-800-523-5622



commodore