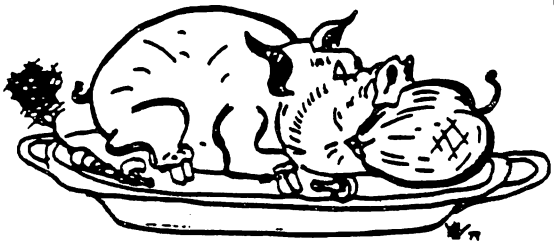
H.A.A.U.G.



HOUSTON AREA APPLE USERS GROUP

THE APPLE BARREL

VOLUME 5 NO. 4 JUNE 1982

PRESIDENT, MIKE KRAMER VICE PRES., BRIAN WHALEY
EDITOR, MIKE KRAMER

*** CONTENTS ***

Houston Area Apple Users Group
APPLE BARREL
2218 Running Springs
Kingwood, TX 77339

CLUB NOTES

The HOUSTON AREA APPLE USERS GROUP is an Apple user club, not affiliated with Apple, Inc., or any retail computer store. HAAUG is a member of the International Apple Core and supports its publications and purposes. General membership meetings are held on the second Thursday of each month in the rear chapel of Memorial Lutheran Church, 5800 Westheimer, between Chimney Rock and Jungman Library, beginning at 6:30 P.M. An additional general meeting is held at 2:00 P.M. the last Saturday of each month at the University of Texas School of Public Health in the Medical Center at 6905 Bertner at Holcomb. This meeting features tutorials, problem-solving sessions, and access to the HAAUG software library. The meeting is held in the main floor meeting room to the left of the entrance. Bring your Apples!!

---==*==---

OFFICERS / EXECUTIVE BOARD

President	Mike Kramer
Vice Pres	Brian Whaley
Treasurer	Wally Edmiston
Secretary	Ruth Dill
Software Lib.	Jim Good
Hardcopy Lib.	Larry Baumann
Membership	Richard Parrish
IAC Rep.	DeWayne Van Hoozer

---===*===---

APPLE HOTLINE

713-668-8685

The APPLE HOTLINE has been established to provide an easy means to learn of meeting topics, news, etc. It can also be used to obtain answers to puzzling Apple - related questions. If you get a recording, leave your name, date, and time. You should get a return call within 24 hours.

---===*===---

MEMBERSHIP INFORMATION

Dues are $20 per 12-month period for regular memberships, $10 for students through high school where no adult member of the family is an Apple user. Please make checks payable to Houston Area Apple Users Group and mail to Richard Parrish, 7306 Arbor Oak Dr., Houston, TX, 77088.

---==*==---

APPLE BARREL REPRINT POLICY

---==*==---

SPECIAL INTEREST GROUPS

Members who share interests are encouraged to join or form Special Interest Groups to more fully explore their fields. These groups meet separately from the regular meetings at times convenient for the members. If you would like to become involved in a special interest group, either call the HOTLINE. Lists of members with specific interests can be generated on request from the HAAUG MEMBERSHIP SURVEY data base.

SIG CHAIRMEN

Business	Rudge Allen
CP/M	Jim Huck
Pascal	Jon Stevens
Sci/Engg	Mike Conway
Statistics	Lindsay Reed
Stocks	George Marsden
Education	Fred Yates
FORTH	Steve Knouse
Assembler	Robin Cox
BASIC	Gus Gusmorino
DOS	Bill Zahrt
Games	Bill Muhlhausen

---===*===---

SPEEDING UP ACCESS TO AND FROM YOUR DISK

BY MIKE CONWAY

The purpose of this article is to provide an alternate way of transferring large amounts of data to and from the disk more quickly and more efficiently. This method has applications in areas where you might use matricies to store data (e.g. stock prices, experimental data, etc.). In this article I will try to show how data is stored in the APPLE and provide sample programs demonstrating how to get data to and from the disk. It is recommended that you try these out as you are reading the article to help you to better understand the process.

PART 1 SENDING DATA TO THE DISK

To start out lets look at the following APPLESOFT program:

```
10  REM  ** DATA SENDING PROGRAM **
20  REM
30  REM  THIS PROGRAM WILL SEND 1000
40  REM  10 DIGIT NUMBERS TO THE DISK
50  REM  IN A TEXT FILE CALLED
60  REM  SIN DATA
70  REM  **************************
100  DIM A(1000)
120 D$ =  CHR$ (4)
130  FOR I = 0 TO 1000
140 A(I) = 100 *  SIN (2 * 3.14159 * I / 100)
145  NEXT I
150  PRINT D$;"OPEN SIN DATA"
160  PRINT D$;"WRITE SIN DATA"
170  FOR I = 0 TO 1000
180  PRINT A(I)
190  NEXT I
200  PRINT D$;"CLOSE SIN DATA"
210  PRINT  CHR$ (7);"DONE!!"
220  END
```

Once you have typed it in correctly, save it to your disk as the program "DATA SENDER". Then run the program. If you time the length it takes to complete the program you will find that it takes almost two minutes to run this program; one minute to do the calculations and another minute to save the data to disk. The time it took to save the data to the disk didn't seem like much but if you had to load and send the the data 10 times in an evening, like when you are updating your stock prices , you can easily spend 20 minutes each day waiting for the in use light to go off.

TOGETHER, LOCKSMITH™, THE INSPECTOR™ AND WATSON™ GIVE YOU TOTAL CONTROL OF YOUR APPLE AND ITS DISKS.

LOCK SMITH™ Our new 4.1 version is by far the most reliable nibble-copy program for the Apple. There simply *is* no competition. Allows you to backup just about *any* diskette. Includes read/write Nibble Editor, Quickscan Analysis, Media Surface Check, Degauss and Erase, Inspector Interface and Disk-drive Speed Calibration utilities. All for just $99.95 at your local dealer or direct.

THE INSPECTOR™ by bill sefton Puts all your disk and memory utilities together where they belong — *inside* your Apple. Eprom or disk version is always at your fingertips. Search memory and disks forward and backwards, read nibbles, map disk space, locate strings, the uses are endless. At your local dealer or direct— THE INSPECTOR, $59.95.

Watson™ T.M. The Inspector's Assistant Includes such goodies as scrolling screen dump, disassembler that shows ASCII, file follower of file, track/sector list-finder by *name*, disk-sector lockout, disk comparer, much more. At your local dealer or direct— WATSON,* $49.95.

If you're at all serious about programming or about business use of your Apple, you must have these interactive utilities. MasterCard and Visa holders order toll-free, 1-800-835-2246.

OMEGA MICROWARE, INC. T.M.
222 SO. RIVERSIDE PLAZA • CHICAGO, IL 60606 • 312-648-4844

*Requires The Inspector

Apple is a registered trademark of Apple Computer, Inc.

1+1+1=4

3 PRODUCTS TO ADD A 4TH DIMENSION OF POWER TO YOUR APPLE.

RAMEX-16™ *(Look Ma, no straps!)* The one 16-K memory expansion card for your Apple that *requires no unnecessary surgery.* This board just plugs in with no strap or additional connections. In spite of its quality, the Ramex-16™ costs just $139.95, complete with a *one year* limited warranty.

The Manager™ A complete turnkey memory management system on a disk— using either one or two 16K cards. HIDOS™ loads DOS onto one RAM card and with the second card loads an alternate language onto another. SOLIDOS™ turns a 16K card into a fast, 45-sector disk-drive emulator. At your local dealer or direct for just $34.95.

THE CONSOLIDATOR™ FOR VISICALC FILES If you use VisiCalc™, then you must have THE CONSOLIDATOR. It will save you hours of keyboard time, by allowing you to manipulate totals of separate files *without reentering them.* Easy to use, invaluable to own. Just $49.95 at your dealer or direct.

All three of these together help make your Apple a more complete business system — giving you expanded memory, extra convenience, sure control. MasterCard and Visa holders order toll-free, 1-800-835-2246.

OMEGA MICROWARE, INC. T.M.
222 SO. RIVERSIDE PLAZA
CHICAGO, IL 60606
312-648-4844

Apple is a registered trademark of Apple Computer, Inc.
VisiCalc is a registered trademark of Personal Software, Inc.

Now that you have run the program we will now take a look at where this data is located in the APPLE'S memory. If you look on page 137 of your APPLESOFT Basic Refference Manual you will see the section on APPLESOFT variable maps. We will use this information to find the data generated in the program. To find where the first and only matrix is located you want to go into the monitor and look at locations $6B and $6C.

```
 CALL -151

*6B.6C

006B- A4 09
```

What you have just found is the initial location of the area that is reserved for storing matrix data. From the values located in locations $6B and $6C the matrix area begins at location $9A4. Remember, the lower order byte is printed before the higher order byte. This location may be different depending on your system configuration. Use the values you obtain in place of the ones used in this example. Lets now look at the first few bytes of the matrix memory area:

```
*9A4.9B4

09A4- 41 00 94 13
09A8- 01 03 E9 00 49 0F DA A2
09B0- 83 48 ED F3 42
```

The first seven bytes give the information about the matrix and how it is stored in the APPLE'S memory. The first two bytes gives the variable's name in ASCII (pg 138 of guide), which in this case is the matrix A. The second byte is used for variables with two letter names; in our case it is the null character. The next two bytes give the number of bytes between this matrix and the next matrix, $1394 bytes. This when added to $09A4, in hexadecimal gives the start of the next matrix. In our case we only have one matrix but if you wanted to look at at more than one, you would have to go to the location defined by these two bytes.

The fifth byte tells what dimension the matrix is; in our case it has a dimension of 1. The sixth and seventh bytes tell how many entries are in the first dimension; in our case the number of entries is $3E9 (1001 in decimal) this agrees with our dimension statement we used in the program. If a multidimensional variable was used, e.g. A(1000,5), there would be two additional bytes describing the length of the second dimension, 5.

Following the matrix description, the rest of the memory area is used to describe he values in the matrix A. The APPLE represents each number in five bytes. So for our matrix, the numerical data is stored in 5005 bytes of memory, 5*1001.

To summarize, locations $6B and $6C contain the beginning of the memory area allocated to matrix A. The first seven bytes of the memory area tell what the matrix name is, where the next matrix starts, how many dimensions the matrix has and how many entries are in that particular dimension. The remaining area is filled with the numerical data requiring 5 bytes per entry. Now a question to ask is if you dont have any entries does the APPLE still allocate all the memory that it did before?

To find out the answer, lets get out of the monitor and load the "DATA SENDER" program again. We will delete the major portion of the program and then run it. After running it we will get back into the monitor and find out how the empty matrix is set up.

```
* 3D0G

]LOAD DATA SENDER

]DEL 120,210

]LIST

10  REM  ** DATA SENDING PROGRAM **
20  REM
30  REM  THIS PROGRAM WILL SEND 1000
40  REM  10 DIGIT NUMBERS TO THE DISK
50  REM  IN A TEXT FILE CALLED
60  REM  SIN DATA
70  REM  **************************
100  DIM A(1000)
220  END

]RUN

]CALL-151

*6B.6C

006B- D9 08

*8D9.8E7

08D9- 41 00 94 13 01 03 E9
08E0- 00 00 00 00 00 00 00 00
```

The matrix is set up identically (except for the data)

as before but there is a difference; the matrix now starts in a different location. This is because the program length has been shortened and the matrix starts in memory right after the program. So any change in the program will probably result in a change in the location of the data matrix. Now we will use this procedure of locating the matrix in memory to shorten the length of time required to store the data on disk.

Now, lets type out another small program that you should save to your disk as "DATA SENDER2".

```
10  REM  ** DATA SENDING PROGRAM2**
20  REM
30  REM  THIS PROGRAM WILL SEND 1000
40  REM  10 DIGIT NUMBERS TO THE DISK
50  REM  IN A MACH.LANG. FILE CALLED
60  REM  SIN DATA2
70  REM  ************************
100  DIM A(1000)
120 D$ =  CHR$ (4)
130  FOR I = 0 TO 1000
140 A(I) = 100 *  SIN (2 * 3.14159 * I /
100)
145  NEXT I
170  PRINT D$;"BSAVE SIN DATA2,A"; PEEK
(107) + 256 *  PEEK (108) + 7;",L5005"
210  PRINT  CHR$ (7);"DONE!!"
220  END
```

Line 170 finds the location of the data contained in matrix A. PEEK 107 and 108 are the decimal equivalents of $6B and $6C that contained the pointer to the beginning of the area of memory reserved to matrix A. The numbers obtained from the PEEK command are converted to the decimal equivalent of that pointer and 7 is added to the product to jump over the first 7 bytes in the file. This is the start of the numerical data contained in matrix A. Line 170 also saves that portion of the matrix to the disk. The length of the file saved is the number of matrix entries times five (1001*5). When you run this program you find that it takes only eight seconds to store the same data on disk rather than the minute it took earlier. Another benefit can be seen by CATALOGing your disk.

```
]CATALOG

DISK VOLUME 254

 A 007 HELLO
 A 003 DATA SENDER
 T 046 SIN DATA
 A 003 DATA SENDER2
 B 021 SIN DATA2
```

This form of data storage takes half the amount of disk

C T I

Verbatim®

31.50 BOX—WITH HUB RINGS

RIBBONS

EPSON MX 70/80

$13.50 EACH

3 FOR $36.00

FLIP 'N' FILE

$27.50

VISIT OUR NEW LOCATION

2802 LOUISIANA
HOUSTON 77002

526-9666

space than the text file method. So, you now have a faster, more compact way of storing your data. But, you do give up the luxury of being able to easily extract the data out of the disk file like you can in a text file. For most cases, this seems like a worthwhile trade.

PART 2. OBTAINING DATA FROM THE DISK

To get the data from the disk we will use the same procedure to locate the beginning of the array but we will BLOAD the data from disk instead of using the BSAVE command. Lets now type in and then save to disk the following program called "SIN PLOTTER":

```
10 REM   **SIN PLOTTER PROGRAM**
20 REM   THIS PROGRAM WILL PLOT EVERY
30 REM   FOURTH POINT ON THE SCREEN
40 REM   FROM THE MACHINE LANGUAGE FILE
50 REM   SIN DATA2
60 REM   ************************
100  DIM B(1000)
120 D$ = CHR$ (4)
125  PRINT D$;"BLOAD SIN DATA2,A"; PEEK
(107) + 256 *  PEEK (108) +7
129  HGR2
130  FOR I = 0 TO 250
131 Y = INT ((120 + B(I * 4)) / 1.5)
132  HPLOT I,Y
145  NEXT I
220  END
```

If the program is running correctly then you should see a sine wave plotting across the screen. If you get a subscript error in line 131 check to make sure line 125 is written correctly. If the procedure is followed correctly you don't have to use the same variable name. Again, you never get something for nothing so, you have to be careful on a couple of items. The dimensions of the sending matrix and the recieving matrix should be the same. Also the calculation of the start of the numerical area of the matrix should be made in the print statement. If you define any new variables between the PEEK statements and the PRINT statement in either the sending or recieving program you will not be able to make the tranfer.

For more complicated matricies (i.e. A(5,1000)) you can write data to a portion of the matrix as well, provided you set up your matrix correctly. For example if you have five files of 1000 entries that you wish to recall and send to the disk separately, dimension your matrix A(1000,5) rather than A(5,1000). This will allow you to add your file in a solid block of 5005 bytes rather than in 1001 five block bytes. Also remember that with multidimensional matricies that the number of bytes defining the matrix grows as well. You have to add 9 bytes to the beginning of a two

dimensional matrix to get to the numerical data rather than the 7 bytes required for a one dimensional matrix.

For people with a APPLESOFT compiler program you can use the same technique to store the array data as shown here. You must note where in the compiler's memory map the array is located. Once you know the location, the method is identical to the method shown here. One advantage with a compiler is that it gives you the start of the data so you do not have to add any bytes to correct for the start of the data. As an example of what you can do with a compiler program and this disk technique, the data sender program can be completed in only 44 seconds using the EXPEDITER II program by ON-LINE and this method of data transfer.

If you need any help with this procedure you can direct any questions/suggestions to the H.A.A.U.G. APPLE HOT LINE and I will get back to you.

----====*====----

ENHANCED POOR BOY WORD PROCESSOR
BY
LOUIS PITZ

I am writing about the article "Poorboy Word Processor", which was in the Fall 81 issue of APPLE ORCHARD. I enjoyed it, and it was a good introduction to word processing for me. I'm even writing this letter with a modification of the program! I've had my Apple][Plus a year now, so I got myself an MX-80 F/T printer for Christmas (with Graftrax also). So when I started to learn and use it, I turned to "Poorboy Word Processor" to learn a bit about word processing. Of course, I'm the kind of person who can't resist the temptation to tinker, so I made some modifications.

There are four main modifications. The first is in Lines 1-8 to show the credits on the screen, not just in the remarks. In the programs I have that I did not write, I want to make sure I give visible credits. The second modification is in Lines 9-11 to give the option between single and double line spacing. Also see Lines 160 and 165. One can only change spacing by ending (via CTRL-E) and starting again by RUN and choosing a new option, since I wanted to make minimal modifications and keep the program short. The third modification is to add a "SHIFT-LOCK" capability for entering a lot of capital letters without having to hit the ESCAPE key each time. See Lines 20, 85, and 90. CTRL-S is a fairly standard toggle command for SHIFT-LOCKING, so I can't take any credit for that idea.

Finally, the fourth modification is to the handling of backspaces. See Lines 30, 60, 70, and 500-540. In the original, backspaces worked O.K. unless you backspace over capital letters and their invisible ESCAPE key commands. Since the ESCAPE key adds to the length of LINE$, but does not advance the cursor, where you see the cursor on the screen can get out of sync with where you are in LINE$. My solution is to not print the backspace before checking if I'm at an invisible character, namely ESC or CTRL-S. If so, I decrease LINE$'s length and clear the keyboard strobe (POKE -16368,0) instead of backspacing. In fact, I keep going back in case I have typed multiple invisibles as I'm a fumble-fingered klutz! Line 500 does a backspace if I was at a non-invisible character.

So the net effect is to always remove the current right-most character of LINE$, whether visible or not, and also any number of immediately preceding invisibles. That way you don't have to trust to memory as to where you had them. Maybe you could find a more elegant way -- I wanted to stay as close as possible to the original program logic and make minimal changes.

I hope you like the modifications - a LISTing follows. Lines changed or added to the original version are marked with an asterisk. Happy Appling.

Editor's note: The above is a slight modification of a letter from Louis Pitz, a fellow Appler from De Witt, Iowa.

```
 1  TEXT : HOME : VTAB 5: PRINT "P
      OORBOY WORD PROCESSOR"
 2  PRINT : PRINT "MX-80 LOWER CAS
      E OUTPUT"
 3  PRINT : PRINT "BY MIKE KRAMER"

 4  PRINT : PRINT "SEE P56&57 APPL
      E ORCHARD FALL 81"
 5  VTAB 15: PRINT "REMEMBER TO SE
      T UP & TURN ON PRINTER": VTAB
      20
 6  PRINT "CONTINUE? Y/N? ";: GET
      A$: PRINT A$
 7  IF A$ = "N" THEN  END
 8  IF A$ <  > "Y" THEN 6
 9  PRINT : INPUT "SINGLE OR DOUBL
      E SPACING? S/D? ";A$
 10  IF A$ <  > "S" AND A$ <  > "D
      " THEN 9
 11 D = 0: IF A$ = "D" THEN D = 1
 15  GOTO 110
 20 LINE$ = "":SL = 0
 30  GET A$: IF A$ <  > CHR$ (8) THEN
       PRINT A$;: NORMAL
   40  IF A$ =  CHR$ (13) THEN  RETURN

   50  IF A$ =  CHR$ (5) THEN  END

 60  IF A$ =  CHR$ (8) AND  LEN (L
      INE$) <  = 1 THEN  PRINT A$;
      : NORMAL :LINE$ = "": RETURN

 70  IF A$ =  CHR$ (8) AND  LEN (L
      INE$) > 1 THEN  NORMAL : GOTO
      500
   80 LINE$ = LINE$ + A$: IF  LEN (L
      INE$) > 75 THEN  PRINT  CHR$
      (7);
 85  IF A$ =  CHR$ (19) THEN SL =
       NOT (SL)
 90  IF A$ =  CHR$ (27) OR SL THEN
       INVERSE
   100  GOTO 30
   110 D$ =  CHR$ (4)
   120  HOME
   130  PRINT "^";:CV =  PEEK (37): GOSUB
       20
   140  IF  LEN (LINE$) = 0 THEN  POKE
       36,0: POKE 37,CV: GOTO 130
   150  PRINT D$"PR#1"
 160  IF D THEN  PRINT  CHR$ (9)"8
      0N": GOTO 170
 165  PRINT  CHR$ (9)"80N";
 170 CAP = 32:SL = 0
   180  FOR I = 1 TO  LEN (LINE$)
   190 CH$ =  MID$ (LINE$,I,1)
   200  IF  ASC (CH$) = 27 THEN CAP =
       0: GOTO 240
 205  IF  ASC (CH$) = 19 THEN SL =
       NOT (SL): GOTO 230
   210  IF  ASC (CH$) < 65 OR  ASC (
      CH$) > 90 THEN CAP = 0
   220  PRINT  CHR$ ( ASC (CH$) + CA
      P);
 230 CAP = 32 - SL * 32
   240  NEXT I
   250  PRINT  CHR$ (9)"I"
   260  PRINT D$"PR#0"
   270  GOTO 130
 500  IF  RIGHT$ (LINE$,1) <  > CHR$
      (27) AND  RIGHT$ (LINE$,1) <
       > CHR$ (19) THEN LINE$ =  LEFT$
      (LINE$, LEN (LINE$) - 1): PRINT
      A$;
 510  IF  LEN (LINE$) = 0 THEN  PRINT
      A$;:LINE$ = "": RETURN
 520  IF  RIGHT$ (LINE$,1) =  CHR$
      (27) THEN LINE$ =  LEFT$ (LI
      NE$, LEN (LINE$) - 1): POKE
       - 16368,0: GOTO 510
 530  IF  RIGHT$ (LINE$,1) =  CHR$
      (19) THEN LINE$ =  LEFT$ (LI
      NE$, LEN (LINE$) - 1): POKE
       - 16368,0:SL =  NOT (SL): GOTO
      510
 540  GOTO 90
```

APPLE BARREL QUIZ RESULTS

The winner of the Apple Barrel Quiz included in the April/May issue was Keith Ebner, one of HAAUG's teenage members. Both Keith and Charles Conger submitted entries postmarked May 12 with the correct number of hidden words. Charles found 39 "official" words plus 2 that had been generated randomly by the computer. Keith found all 41 "official" words listed below plus IBM, key, END, TTY, RUN, and RTU (remote terminal unit). Keith was presented the prize, Flight Simulator, at the May 29 meeting.

ALPHANUMERIC	APPLE	APPLESOFT	ASCII
BAUD	BINARY	BIT	BOOTUP
BUFFER	BYTE	CATALOG	COMPUTER
CRASH	DIGITAL	DISK	DISKETTE
DOS	FILE	FIRMWARE	HARDWARE
HEXADECIMAL	HIMEM	INITIALIZE	INTEGER
KEYBOARD	MICROSECOND	MODEM	MONITOR
NIBBLE	PARALLEL	PASCAL	PERIPHERAL
POINTER	PRINTER	RAM	ROM
SECTOR	SEQUENTIAL	SERIAL	SOFTWARE
TRACK			

----====*====----

UNJUMBLER
BY LEE GILBREATH

Last month I didn't receive my subscription to the Apple Barrel until two weeks after it was posted in the mail. In the issue was the Apple Barrel Quiz which offered a prize worthy of the effort. Everybody had a head start on me and I felt that the odds had to be put back into my favor. Hence the following program was written to enlist the aid of my Apple in cracking the word jumble.

```
1  REM      UNJUMBLER
2  REM  BY LEE GILBREATH
3  REM       5-23-82
10  HOME
800  INPUT "HOW MANY CHARACTERS I
     N A LINE? ";E: IF E > 30 THEN
      PRINT "MY LIMIT IS 30, AGAI
     N, PLEASE.": GOTO 800
810  INPUT "HOW MANY LINES? ";B: IF
     B > 20 THEN  PRINT "MY LIMIT
      IS 20. AGAIN, PLEASE.": GOTO
     810
812  HOME : DIM A$(E,B),L$(B)
820  INVERSE : VTAB 1: HTAB 2: FOR
     I = 1 TO E: PRINT  MID$ ("
           1111111112222222223
     ",I,1);: NEXT
830  VTAB 2: HTAB 2: FOR I = 1 TO
     E: PRINT  MID$ ("12345678901
     2345678901234567890",I,1);: NEXT

840  VTAB 3: HTAB 1: FOR J = 1 TO
     B: PRINT  CHR$ (64 + J): NEXT

860  NORMAL : VTAB 23: PRINT "INP
     UT PUZZLE, LINE BY LINE.";
910  VTAB 3: FOR J = 1 TO B: HTAB
     2: INPUT "";L$(J): NEXT
960  VTAB 23: PRINT "HIT ANY KEY
     FOR NEXT DIRECTIONAL PASS.";

965  IF  PEEK ( - 16384) < 127 THEN
      GOTO 965
967  POKE  - 16368,0
970  FOR J = 1 TO B: FOR I = 1 TO
     E:A$(I,J) =  MID$ (L$(J),I,1
     ): NEXT I: NEXT J
1000  FOR V = 1 TO B: FOR H = 1 TO
     E:PASS = 1
1002  INVERSE
1005 I = H:J = V
1010  FOR X = 1 TO 9: GOSUB 1100:
      NEXT
1020  IF FLAG = 1 THEN FLAG = 0: GOTO
     1040
1030  HTAB H + 1: VTAB V + 2: GET
     A$: NORMAL :FLAG = 1: GOTO 1
     005

----====*====----

1040 PASS = PASS + 1: IF PASS = 9
      THEN  NEXT H: IF H > E THEN
      NEXT V: IF V > B THEN  END

1050  GOTO 1002
1100  HTAB I + 1: VTAB J + 2: PRINT
     A$(I,J);: IF FLAG = 0 THEN  VTAB
     10: HTAB 31 + X: NORMAL : PRINT
     A$(I,J);: CALL  - 868: INVERSE

1110  IF PASS = 1 THEN I = I + 1:
     J = J - 1
1120  IF PASS = 2 THEN I = I + 1
1130  IF PASS = 3 THEN I = I + 1:
     J = J + 1
1140  IF PASS = 4 THEN J = J + 1
1150  IF PASS = 5 THEN I = I - 1:
     J = J + 1
1160  IF PASS = 6 THEN I = I - 1
1170  IF PASS = 7 THEN I = I - 1:
     J = J - 1
1180  IF PASS = 8 THEN J = J - 1
1200  IF I > E THEN X = 10
1210  IF J = 0 THEN X = 10
1220  IF I = 0 THEN X = 10
1230  IF J > B THEN X = 10
1240  RETURN
```

I/O Slot Info
by
Guy Neill

In a recent newsletter Iincluded a short program by Mike Kramer to PEEK the slot locations and display the resultant values. I received some information from Mike Kramer and John Kilgore as well as some from anonymous sources. Lastly I set about to get what values I could. The results associated with each peripheral are shone below.

These values may be used within a program to locate a particular card. Assume you are modifying "File Cabinet" so that it will seek a printer regardless of the slot it is plugged in. If you use a routine such as the this:

```
10 FOR X = 1 TO 7 : N = 49152 + 256 * X
20 IF PEEK(N) = 24 AND PEEK(N+1) = 176 AND PEEK(N+2) = 56 AND
   PEEK(N+3) = 72 THEN SLOT = X
30 NEXT
40 HOME : PRINT "THE PARALLEL BOARD IS IN SLOT ";SLOT
```

will result in SLOT containing the number of the slot containing the parallel interface (which in this case is assumed to connect to the printer).

In this manner the values listed below can be very important if you are trying to write a program for a wide spectrum of users in which it is necessary to find a certain peripheral.

I hope this will be of use to you. If you have any other information or peripheral card values send them to me and I will pass the values on to the membership.

PERIPHERAL CARD	BYTE 1	BYTE 2	BYTE 3	BYTE 4
Parallel	24	176	56	72
Disk Controller	162	32	160	0
Hayes Micromodem	24	176	56	184
Z80 Softcard	160	255	0	64
AIO Serial	44	88	255	112
Videx Videoterm	44	203	255	112
MCI Clock	8	120	44	255
MCI Romplus	44	88	255	56
Strobe Plot /int	160	0	0	160
Corvus Omninet	36	32	160	0

As you can see, some of the cards only vary by one byte. This was originally pointed out to me by Mike Kramer after I had written the original article using a program from Beagle Brothers which only looked at the first byte. Using only the first byte will likely cause some problems as evidenced in the above information.

The next problem for which I would like a definative answer is whether or not there is a memory location which may be PEEKed to determine whether there is one or two disk drives connected to a disk controller card. It seems there should be some way of determining this, but I have not yet discovered it. I would appreciate any thoughts you have pertaining to this.

----====*====----

ROM BOARD SYSTEM

BECAUSE READ ONLY MEMORY WILL NEVER FORGET A FIRMWARE PROGRAM, IT WILL STILL BE THERE WHEN THE APPLE IS TURNED OFF AND ON AGAIN.

FORMAT ROM SYSTEM
$64.95

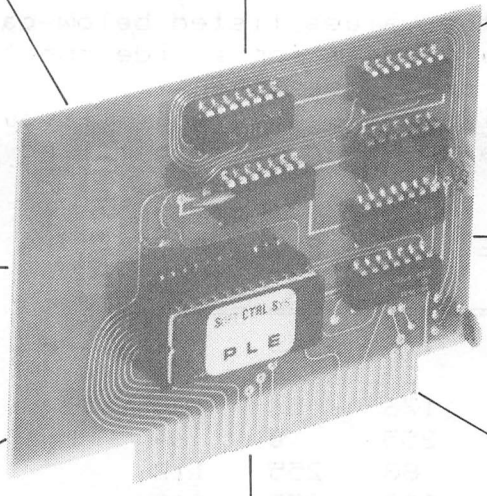
Word processing capabilities for Applesoft print statements and versatile 'print using' commands.

APPLESOFT UTILITY ROM SYSTEM
$59.95

Contains the five most needed utilities. Auto line numbering, program list control, revive a lost program, alphabetize a catalog directory, and expunge DOS.

SORT ROM SYSTEM
$59.95

Machine language sorting routines for string, integer, real, multi-dimension arrays, and record keeping arrays.



DISK COPY ROM SYSTEM
$59.95

Disk backup made easy and convenient.

APPLESOFT* RENUMBER/MERGE ROM SYSTEM
$59.95

Renumber all or part of a program or merge two programs together.

CATALOG COMMAND ROM SYSTEM
$59.95

One key control of disk files to delete, lock, run, load, etc. Also displays a map of a disk's used and unused sectors. Ideal for 'Hello' programs.

ROM BOARD (WITHOUT A ROM)
$29.95

Has built in I/O and device select lines. Will accept 2716 EPROMS or 2316 ROMS.

APPLESOFT* EDITROM SYSTEM
$59.95

Easily and conveniently search, change, or remove any variable, string, or basic command in a program.

The ROM BOARD SYSTEM can be used in any slot to suit your system configuration and is activated with the standard 'PR#' command.

(Check or M.O.) Visa or MasterCard Accepted

PLEASE ADD $3.00 FOR SHIPPING

SOFT CTRL SYSTEMS, BOX 599, WEST MILFORD, NJ 07480

*REGISTERED TRADEMARK
ALL FIRMWARE IS COPYRIGHTED

201-728-8750

Product Review
Format ROM

Soft CTRL Systems
Box 599
West Milford, NJ 07480
201-728-8750

Cost:
$49.95
Hardware required:
ROM Card

Reviewed by Mike Kramer

INTRODUCTION

One of the features of the Apple IIc which has great appeal to the serious programmer is the ability to access machine language programs from BASIC. Unfortunately, it is often difficult to find a memory location to load those programs where they won't interfere with the BASIC program accessing them. Various tricks are used, including loading the machine language between DOS and the DOS buffers or loading it into the top of the keyboard input buffer. The problem becomes particularly frustrating of you want to run a machine language utility with a BASIC program in memory and it loads right on top of your program. One way around this is to have your favorite utility programs in ROM (Read Only Memory) on a ROM card, such Mountain Hardware's ROMPLUS Board or Andromeda's ROMBoard. The best known of these is probably the ROMPLUS, which has sockets for six 2716 2K ROMs, for a total of 12K bytes of resident machine language space. Any one of the ROMs can be accessed either from the keyboard or from within a program. The Andromeda board can hold two 2716's but the desired ROM must be selected with a toggle switch on the rear edge of the board.

Until recently, the only way to obtain ROMs for use with these boards was to have and know how to use an EPROM (Eraseable Programmable Read Only Memory) "blaster" such as Mountain Computer's ROMWriter. Now there are few sources of preprogrammed ROMs at affordable prices ($30 - $60). One of these companies, Soft Control Systems of West Milford, Connecticut, produces a varied selection of EPROMs which, when installed in a ROM board in the Apple IIc, provide sophisticated machine language capabilities which are instantly accessible. Included in their line of ROMs are Disk Copy, Renumber/Merge, Dual Dos, PLE, Applesoft Utility, and Command ROMs. This review will briefly discuss one of the newest, the FMT Format ROM.

CAPABILITIES OF THE FORMAT ROM

The FMT ROM provides the BASIC programmer with many of the capabilities of a word processor for formatting output to either the screen or to a printer. These capabilities include definition of print margins, setting page width and length, center/right/left justification, indenting/outdenting, word wrap, etc. The FMT ROM also provides a powerful PRINT USING capability that handles four types of output - string, floating point, fixed point, and fixed point with commas. PRINT USING can be used to tabulate data, line up decimal points, right justify, and pad alpha or numeric data with a user-defined character.

Applesoft's Ampersand "&" command is used to activate the FMT ROM's various functions. When the Ampersand is the first character of a keyboard command or a BASIC program line, Applesoft jumps to a special location that may contain a jump vector (address) to a user machine language subroutine. It turns out that the Ampersand is quite overworked. The FMT ROM saves the Ampersand vector and then restores it when reactivated, a very nice feature.

DOCUMENTATION

The Format ROM comes with two manuals, one covering installation, the other, operation. The installation manual is short but gives detailed instructions for installing the ROM in both Mountain Computer's and Andromeda's boards. Photographs are used to illustrate where to insert the ROM and how to position it. The need to observe certain precautions to avoid damage due to static electricity is stressed. A registration form is provided in the installation manual, mainly so that customers can be advised of new products. There is, however, no warranty since damage to the ROM caused by the customer during installation is beyond the control of Soft CTRL Systems. Soft CTRL Systems does state that each ROM is tested before shipment to assure that it is functional. The second manual has 37 pages and explains in great detail how to use the many features of the FMT ROM. It takes a bit of study to learn to use the ROM effectively, but many example programs are given.

OVERALL IMPRESSION

The FMT ROM and the documentation provided with it are very well done. The ROM provides capabilities that could prove useful in formatting output to the screen or printer and is very easy to use. Any software taking advantage of the FMT ROM's power, however, becomes hardware dependent, which would not be of concern to the casual user. It is my personal opinion that ROMs are most useful when they provide utilities for program development, copying disks, etc., since they are always available when needed. Soft CTRL Systems markets that kind too. Write to them at the address given in their ad in this issue and ask for their catalog to see what their other ROMs can do.

----====*========

MEETING TOPICS

June 10 Thursday Meeting - Cyborg Corporation of Newton, Mass., will discuss ISAAC, their Apple][laboratory instrumentation interface. ISAAC includes powerful extensions to Applesoft to permit monitoring processes and presenting them graphically.

July 8 Thursday Meeting - Dave Reed of Apple's Sales Support Group in Dallas will discuss Apple LOGO and what he can about new developments. This is a rare opportunity so plan to attend.

B.I.S

Instant Information is Power

GENERAL LEDGER ON LINE

Instant Posting-Immediate Reporting

POWER to make fast, efficient management decisions. POWER to adjust to fluctuations in sales. POWER to increase productivity and profits.

B.I.S is more than just a general ledger . . . it's business information system that can produce selected business activity reports immediately.

B.I.S IS FAST

- Machine language saves time . . . accepts 3000 entries a day as fast as you can type. Every entry is saved for selective analysis.
- Instantaneous posting - 40 split entries can be completely posted in less than 20 seconds.
- Immediate reporting - within four minutes of any entry, a complete BALANCE SHEET, P&L, or DETAILED SELECTIVE ANALYSIS REPORT reflecting that entry can be printed and reviewed.

B.I.S IS FLEXIBLE

- Define your own Journal Names, Account Prompts, Activity Codes, Custom Report Generators, Account Ranges and Descriptions.
- A single B.I.S General Ledger handles THREE completely separate sets of books with consolidating capabilities.

B.I.S IS EASY TO USE

- Menus are simple.
- Accounts and Journals are already set up.
- Hand-holding Tutorial Manual outlines every key stroke.

Put the power of B.I.S to work for you.

B.I.S is available from your local Apple dealer for $295.
If he's out of stock, have him give us a call at (214) 341-1635 or write:

John Broderick, CPA
Broderick & Associates
8635 Shagrock
Dallas, Texas 75238

Verbatim

Datalife

DISKETTES

5 1/4"

$28.50

Per Box of 10

Printer Ribbons All Brands

Computer Supply Co.

789-3892

OPEN 6pm-10pm

8inch diskettes $35 per box

Getting It Out of Your System

by Ed Seeger

"I know that age to age succeeds,
Blowing a noise of tongues and deeds,
A dust of systems and of creeds."

-Tennyson

Bill Blue. Roger Wagner. The Aldriches. Randy Hyde. Nasir. Val Golding. If you've gotten beyond "Breakout" with your Apple you have heard at least a couple of these names. Each has made an important mark on the world of Apple computing. Each is in some way tops in making the 6502 chip inside your Apple do wonderful things to amuse you, teach you, do useful work for you.

But how about Ward Cristensen? Or Randy Suess? Or maybe "Zoso"? Keith Peterson, then? Don't know any of them? Perhaps "Lifelines"? Don't read "Lifelines" either? Probably think "Softalk" is the last word in computing. None of this second set of names meant anything to me, either, until eight months ago, but nowadays I don't think I could live without them!

If you're into assembly language you certainly recognize STA, LDA, JSR, CMP and lots of others. But how about MVI or MOV or LXI or CPI or the mysterious SHLD? Undocumented 6502 opcodes? Nope. They're from another chip, an alien, older and some think wiser than the 6502. It's the matronly 8080, or its debutante daughter, the Z80, who wears comfortably all of mom's designer fashions and is as much at home in my Apple these days as is the 6502. But instead of sitting all nestled down on the mother-board, it's stuck on a board 'way up in slot #7 and has a red light winking seductively on top. Meet the Z-80 Softcard, by Microsoft, and step into the cosmos of CP/M. And while you're at it, meet Ward and Randy and Keith and Zoso. Fine, fine people, who have pushed back my Apple horizons as much as have Bill and Roger and Randy and Nasir. And Val.

CP/M is to the 8080 and Z80 chips what Apple DOS 3.3 is to the 6502 -- an operating system, a unified series of commands, an atmosphere, if you will, within which software breathes and thrives and works. But unlike DOS 3.3, which only works on the Apple, CP/M works on a vast number of other computers, including the Apple with a Z-80 Softcard. Try running "DB Master" on your friend's Osborne computer, or on that brand new TRS-80 Model II down at the office, or on the Cromemco your lawyer's using for word processing. Won't work. "DB Master" is a superb database program, but it's built exclusively for the Apple and its DOS. In a word, it's not <u>portable</u>.

Now try it with, say, "dBASE II," which is an equally powerful database program, or more accurately, a database <u>language</u>. Get it running under CP/M on your Apple. Using a modem, send an address file of maybe 800 records over to a TRS-80 Model II which is also using the CP/M operating system as supplied by Pickles and Trout. Your file works perfectly over there, and it will work perfectly on an Altos or a Televideo or a

Vector or a NEC or a you-name-it, as long as it uses the CP/M operating system. Portability! From the Tower of Babel to the United Nations! CP/M is the lingua franca of microcomputers, the universal means to get them talking to one another, working together regardless of brand or design.

Christensen? Petersen? "Lifelines"? Prominent names among many CP/M users. Ward Christensen is the telecommunications whiz whose MODEM7 program, widely available for free in the public domain, has transferred hundreds of thousands of data files and programs over telephone lines from one computer to another. Ward's protocol, or set of operating standards, is so well established that when Southwestern Data Systems produced its "Z-Term Professional" terminal program for the Apple with CP/M, the Christensen protocol was made a part of the program.

Keith Petersen? Keith's name appears on dozens and dozens of public domain CP/M programs as author or as modifier. What one person writes, someone else is sure to improve, particularly in the public domain. In addition to placing his stamp on so much software, Keith operates one of the more impressive public-access bulletin board systems in the country. 10 megabytes (!) of highly useful utility programs, written in BASIC, 8080 assembler, and "C", a structured language with broad currency in the CP/M world. Phoning the Royal Oak, Michigan, Remote CP/M bulletin board on a Saturday for a peaceful hour of software transfers is like calling Republic Airlines for flight information; chances are you'll get a busy signal, so many other callers are after the same thing. Can't stay by the phone re-dialing? Use Christensen's MODEM7 in its autocall mode. It'll dial and re-dial all morning if you like, and signal you with frantic control-G's from the next room when it finally breaks through the pileup and connects you to the Royal Oak hard disks.

But by no means all of CP/M-dom is public domain, although in an age of anxiety when Apple programmers are running downwind to elude the pursuing pirates, it's refreshing and a bit astonishing how much top-flight material has been selflessly offered to one and all. Perhaps this is what happens when programmers have a common means of communication -- they cooperate, rather than merely compete.

"Lifelines" magazine is the preeminent channel of communication among CP/M users and programmers. Published monthly, it is an information service which offers vital news covering new products and new versions, tips for users, product comparisons and other feature articles to guide the reader before and after purchases. "Lifelines" is oriented toward the serious CP/M user, which includes both the "hacker" who ceaselessly writes or modifies code, and the end user, who knows little about computers but uses one as a work tool for accounting, word processing or database management. Note that this differs in concept from "Softalk," which focuses not on an operating system, but on a computer, the Apple, and only the Apple. "Lifelines" is important reading whether you own a CP/M Apple, a Vector, a North Star or an Imsai -- one of those "other" computers whose ads crowd the 500+ pages of "Byte".

Although CP/M as an operating system has its reasoned detractors, the software that runs under it has few peers. The venerable "VisiCalc" is mightily challenged by "SuperCalc" and "CalcStar" and other contenders that emerge almost monthly. "DB Master" and "PFS," strong in the Apple world, cannot touch "dBASE II" for power. "Home Accountant," number 3 on the "Top Thirty," pales beside the integrated Designer Software line of Palantir accounting, nor can the popular "Apple Writer" or "ScreenWriter II" come anywhere near the "Palantir Word Processor." And good terminal programs like "Data Capture 4.0," even with its 80-column capability, simply cannot perform the many essential functions of "Z-Term Professional."

Is Seeger a fanatic about CP/M? Maybe so, but my intention is not to applaud my good judgment in applying CP/M-based software to the many tasks I accomplish with the Apple. It is rather to point out that Apple DOS 3.3 is by no means the whole world. So many new Apple users ask what CP/M is, what it can do, "how it is different from Apple," as it is so often phrased, that something needed to be said here about it.

Your HAAUG has an active CP/M interest group, coordinated by Jim Huck, and aiming to make available much of the public-domain utility software I referred to above. There are bulletin board systems around the country and around the world whereon once you have read other people's messages and left your own, you can actually operate the remote computer from your own Apple, transferring programs and reading documentation files and exploring the intricacies of someone else's hobby. And there is application software of the sort mentioned here, serious and sophisticated tools to do work with a speed and accuracy and of a complexity addressable only by computer.

As Tennyson suggests, systems come and systems go. IBM would like the world to spurn CP/M in favor of its own MS-DOS. The new 16-bit computers have fostered a teriffic struggle for ascendance of a new operating system to supplant CP/M. And it may happen, even as CP/M-86 maneuvers Shogun-like for industry dominance. Now, if there were only a graphics standard so I could transport "Raster Blaster" pinball down to the office . . .

---===***===---

"Lifelines" is available for $18 for 12 issues from Lifelines Publishing Company, 1651 Third Ave., New York, NY 10028.

The Royal Oak RCPM system is available 24 hours a day at 313-759-6569. It is a call back system. Ring once. Hang up. Call back. The computer logic recognizes the sequence and answers automatically. Feed it a carriage return or two if nothing happens. To the question "How many nulls do you need?," answer 0. Follow your nose from there on. If you wish to practice on a less intimidating system, there are several experimental RCPM's in Houston. Ask Jim Huck for advice.

---====*====---

PAGE 23
FRIDAY, MAY 14, 1982

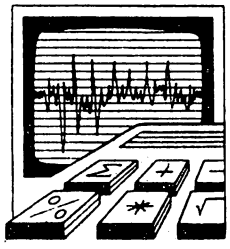
THE WALL STREET JOURNAL.

In a Rapidly Changing Field, Apple II Shows Staying Power

By Richard A. Shaffer
Staff Reporter of The Wall Street Journal

SOME PRODUCTS TAKE ON lives of their own. From its first day on the road, the Ford Model T, for example, was a plain, hard-riding, underpowered car, but it sold by the millions for two decades.

Ford wouldn't change the car. Dozens of other companies would, however, and they prospered by offering better tires, axles, suspensions, gears and stylish attachments for the Model T. A dozen years after its introduction, the car had long been obsolete in design. But it was still so popular that Henry Ford fired a relative for recommending a substitute and took a sledge hammer to a prototype of the car.



Mr. Ford, however, went too far. By making one standard product that buyers could improve if they wished, he created a legendary American success. By keeping the outmoded car alive for almost 20 years, he enabled General Motors, with its more advanced Chevrolet, to become the largest automobile producer.

In the world of high technology, a somewhat similar story may be unfolding at Apple Computer Inc., whose Apple II model has become the best-selling personal computer despite its inadequacies. It hasn't much memory, for example, and its brain is an aging microprocessor that operates about a tenth as fast as today's designs. It can't display many characters or letters on a video screen and has no lower case.

ABOUT 60 COMPANIES, however, have sprung up to sell Apple II attachments or replacements for most of its major parts. Even its brain can be changed, so that the Apple II acts like another brand of computer. As a consequence, although computers with newer technology abound, Apple Computer continues to sell about 20,000 IIs a month.

Just as Mr. Ford did when he eventually brought out Lincoln automobiles, Apple is preparing to expand its line with a much more expensive computer, known as Lisa. But with sales of the Apple II apparently leveling off, the company is readying a successor known as the Super II. And the sort of replacement computer Apple apparently has designed says a lot about the possibility that computers like the Apple II will be around for many more years.

Apple won't discuss products in development. But industry sources give this picture of the Super II: a revised Apple II with a full upper and lower case keyboard and video display. Forty letters or numbers on each video line, but a possible 80 with the addition of a $200 circuit. Able to run all Apple II programs, without modification.

The major expected change, however, isn't in design, features or price but in manufacturing approach. Functionally, the machine is said to be still an Apple II, not much more capable than what a customer willing to add a few parts could have today. But while the Apple II has 136 standard integrated circuits, the Super II is understood to have only 11—one for the microprocessor, eight for 64,000 characters of main memory and two for all the other electronics.

GIVEN THE VOLUME of Apple sales, the change should enable the company to reduce its manufacturing costs significantly. The savings could be used to lower the retail price of Apples if competition increases or to encourage dealers to sell more Apples by increasing their margins.

If speculation about the Super II proves correct, industry people think the new machine could be made profitably for perhaps five years, giving Apple's basic computer design a decade or more of life. Such longevity isn't unknown among computers. By improving the technology inside them, Digital Equipment Corp. has maintained a demand for some of its oldest minicomputers, such as the PDP-8, first built in 1965.

But longevity is rare. If the basic Apple design does survive, it will be due in large part to a feature that many people would think unremarkable—the eight slots inside the Apple II into which additional parts can be plugged. That feature, and the company's practice of providing technical, and in some cases financial, aid to outsiders who wanted to design circuits for those slots, has helped the Apple II compete with newer computers.

For example, because it uses a more advanced microprocessor, the International Business Machines Corp. Personal Computer is, in theory, able to keep more data in its memory than the Apple II.

BUT A CALIFORNIA company, Sorrento Valley Associates, of San Diego, has begun making a plug-in that raises Apple memory beyond the IBM level. Another plug-in, from Saturn Systems of Ann Arbor, Mich., allows the Apple to create financial planning models with VisiCalc, the most popular modeling program, that are several times larger than the IBM computer can handle.

Plug-in boards have made the Apple the computer most often used for the large library of programs based on CP/M, a popular foundation program that otherwise couldn't be used with Apples.

Metamorphic Systems Inc. of Boulder, Colo., recently announced its MetaCard, which opens the Apple owner's door to IBM Personal Computer programs. And one of the most powerful new microprocessors, Motorola's 68000, is available for the Apple from Digital Acoustics Inc. of Santa Ana, Calif.

"The slots allowed others to develop products for the Apple," says Jeffrey Mazur, who writes about Apples for Softalk, a trade publication. "The more products, the more reason to buy an Apple. The more Apples out there, the more incentive for product companies to come up with something for the Apple. The result is a lot of Apples that should be with us a long time."

Disk Cleaning Program

Steve Knouse

I'd heard that I should clean my disk drives periodically so I bought a head cleaning kit (complete with instructions). The instructions said to insert the cleaning disk in the drive and turn on the drive motor for 30 seconds. Wait a minute! How could I turn on the drive for 30 seconds? A CATALOG command would only turn it on for about five seconds before I got a beep and an I/O ERROR message. It wasn't very elegant but 6 CATALOG's did the trick.

Then along came Don Worth and Pieter Lechner's classic, "Beneath Apple DOS." Chapter 6 has a section on direct control of the disk drive. Referencing the following addresses (eg. by a PEEK or POKE from Basic) will perform the indicated action:

ADDRESS TO REFERENCE		ACTION
HEX	DECIMAL	
C088+Slot*10	49288+Slot*16	Turn motor off
C089+Slot*10	49289+Slot*16	Turn motor on
C08A+Slot*10	49290+Slot*16	Select drive 1
C08B+Slot*10	49291+Slot*16	Select drive 2
C08D+Slot*10	49293+Slot*16	Load data latch*
C08E+Slot*10	49294+Slot*16	Set read mode*

*Reading either of these locations after the other will sense the write protection status. If the high bit is set (a PEEK value > 127) it is write protected.

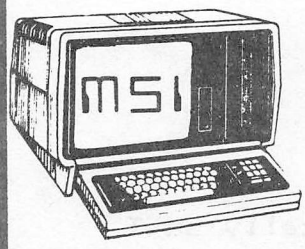
As an example, to clean both drives in slot 6, I now use the following set of commands:

```
]CALL-151       Get to the monitor

                Insert a cleaning disk in drive 1
:C0EA C0E9      Turn motor on - drive 1, slot 6
     .
     .          Wait 30 seconds
     .
:C0E8           Turn drive motor off

                Insert the cleaning disk in drive 2
:C0EB C0E9      Turn motor on, drive 2, slot 6
     .
     .          Wait 30 seconds
     .
:C0E8           Turn off motor

:3D0G           Return to Basic
```



MICRO Solutions, INC

SOFTWARE STORE

TRAINING - INSTALLATION - CUSTOMIZATION

INTRODUCING HOUSTONS SOFTWARE ONLY STORE

MICRO SOLUTIONS

We are Houston's first Software only store. We primarily sell software for the micro computer user. We also sell accessories for your computer when the demand is high. To serve you better we need to know which software you need so please ask!

****** SPECIAL GET ACQUAINTED OFFER ******
10 SINGLE DENSITY DISK FOR $19.95

TELEPHONE MODEM BY SIGNALMAN---------- 99.95
is the smallest, lightest, most compact modem available today. It has automatic mode selection which eliminates confusion of who is originator - ends need to manually switch from Originate to Answer and vice/versa.

MICROBUFFER FOR EPSON PRINTERS---------- 159.00
Your printer is capable of sending data faster than the printer can print. This means your computer is forced to wait for the printer. A waste of valuable time. Microbuffer model MBP-16K is a centronics-compatible parallel interface with 16,384 bytes of on-board RAM for data buffering. This buffer stores the data in its own memory buffer and then takes control of the printer. This frees your computer for more productive functions.(req. MX80, MX80F/T, MX100 & supports Graftrax).

BMC-12A GREEN SCREEN MONITOR---------- 109.00
This 12" diagonal green screen, gives you bright and clear resolution. It's compatible with just about any micro on the market, including APPLE.(req. RCA phono plug).

GAMES FOR YOUR APPLE

SARGON CHESS GAME disk ----------	34.95
OLYMPIC DECATHLON disk (from Microsoft) ----------	29.95
ADVENTURELAND disk (Scott Adams Graphic Adventure) ----	29.95
ELIMINATOR super arcade game (disk) ----------	29.95

OTHER COMPANIES REPRESENTED

MICROSOFT
MICROPRO
ON-LINE SYSTEMS
SYSTEM PLUS ACCOUNTING SOFTWARE
VISICORP (Personal Software)
BORDERBOUND SOFTWARE
SIRIUS
DATAMOST

==========

10 % DISCOUNT ON ALL SOFTWARE, BOOKS AND MAGAZINES

==========

9949 Harwin #E • Houston, Texas 77036 • Phone (713) 789-5443

My company has a number of Apple II's and Apple III's. I wanted to send disk cleaning instructions to the users but couldn't publish the above list of commands; most of our Apple III users don't have a DOS 3.3 bootable disk to get into the emulation mode. Instead I wrote the following program to guide them through the cleaning procedure. This program will run on either an Apple II or an Apple III in emulation mode. The program has been submitted to the club's Software Library (May 29, 1982).

One thing I did not accomplish with this program was to turn on drives 3 and 4 of the Apple III. They can't be reached in the emulation mode and I don't know how to turn on the drives in the Apple III native mode. If anyone has a solution or suggestion call the HAAUG Hot-Line (713-668-8685); better yet publish it in the Apple Barrel.

Program Logic Description

Lines 130-300 clear the screen, print the default choices, and print a set of "mini-instructions" near the bottom of the screen. The default slot is the one last accessed before the program was run and is taken from the RWTS parameter list. Line 190 uses information in a machine language routine in page 3 to locate the start of the RWTS parameter list. The last slot accessed is in the 16th byte (starting byte + 15) and is stored as slot * 16. The default drive is 1.

Lines 340-350 get the choice by reading the keyboard directly. I did this to eliminate the blinking cursor that I would have gotten with a GET or INPUT statement. The VTAB in line 340 positions the cursor (which is not visible) so multiple invalid choices will not scroll the screen. The keypress is converted to a character string in line 350 so the checks which follow would be more readable.

Lines 390-440 check which option was chosen, do the desired action and loop back for another choice.

A "1" (line 390) will select the next drive with a disk controller. I turn off the drive, waiting a while to ensure it is off, select the next slot with a disk controller card and set drive 1. You will not see the slot change on a one controller system but the drive motor will still go off (if on) and the selected drive will change to 1 if it was 2. Note that I set drive 1 by telling the program that drive 2 is currently selected and calling the subroutine (GOSUB 570) that toggles the drive.

A "2" (line 400) toggles between drive 1 and 2. If a drive motor is on it is turned off. As above the program waits for the drive to go off.

Yankee Micro
P.O. Box 601
Milford, NH 03055

A "3" (line 410) toggles the drive motor on or off. When a drive motor is turned on an error message is issued and the motor is turned off if the diskette is not write protected (cleaning disks don't have write-enable notches).

A "4" (line 420) turns off the drive (if on) and displays a screen of instructions.

A "5" (line 430) will exit the program after turning off any drive which was on. I clear the mini-instructions near the bottom of the screen by going to the line with the instructions (VTAB (21)), and clearing to the end of the screen (CALL -958). The VTAB (23) just before the EXIT puts the Applesoft prompt at the bottom of the screen without scrolling it.

Any other choice will ring the bell by printing a control-G (line 440).

Lines 480-810 are subroutines to select the next slot, toggle the drive, and turn the motor on or off.

The routine to select the next slot (lines 480-530) bumps the slot number (when it becomes greater than 7 it is reset to slot 1) and checks for a disk controller card. This is done by PEEKing the first four locations in the slot (CnO0-Cn04, where n is the slot, program variable SL). If these locations don't have the decimal values 162, 32, 160, and 0, no controller card is present and the program loops back to line 480 to bump the slot number and try again. If you have a one controller system you will not see the slot change.

Lines 570-580 toggle the selected drive between drives 1 and 2. For example if drive one is the current drive, program variable A1=1 and drive 2 will be selected (line 570). In selecting a drive a PEEK is done to a location on the controller card to set the appropriate drive (eg. hex C08B+(slot*10) to select drive 2). This way the routine which turns on the drive will not have to select it.

The routine at lines 620-630 turns off the drive motor if it is on. A null DO LOOP produces a time delay. The length of the DO LOOP was picked by trial and error to give a delay slightly longer than the time needed to turn off my drives (about three seconds). The count needed may vary if your drives take much longer to turn off.

Lines 670-810 turn the drive motor on if it was off (program variable A2=0) and turn it off if it was on (A2=1). Just after turning on a drive, a check is made to see if the disk is write protected (line 740); cleaning disks do not have a write-enable notch. This also sets the read mode in case your write protection switch is not working. If the disk is not write protected, the program prints a flashing message to that effect and the drive is turned off. This prevents you from accidentally

erasing a disk (like the one you ran the program from). Note the program goes back to the beginning (line 130) so the slot and drive will be reset to the slot last accessed before the program was run and to drive 1. The POP instruction in line 740 cleans up the return stack since the routine leaves the subroutine with a GOTO rather than a RETURN.

At lines 860-870 is the routine to ring the bell three times and print a warning that the disk is not write protected. The routine waits for a keypress before returning.

Lines 910-1110 print the instructions screen and wait for a keypress before returning to the main menu. Once again the program goes back to the beginning (line 130) so the slot and drive will be reset to the slot last accessed before the program was run and to drive 1.

The subroutine at lines 1150-1190 is used to wait for a keypress before continuing. This routine is used by the instructions screen and the screen that warns that the disk is not write protected.

```
LIST

10  REM
11  REM
12  REM DISK CLEANER
13  REM
14  REM BY STEVE KNOUSE
15  REM
16  REM HOUSTON AREA APPLE USERS
17  REM GROUP (HAAUG)
18  REM
100  REM
110  REM CLEAR SCREEN & PRINT HEA
     DING
120  REM
130  TEXT : HOME :G$ =  CHR$ (7):
      REM ^G (BELL)
140  INVERSE : VTAB (2): PRINT  SPC(
     50)"DISK CLEANING PROGRAM" SPC(
     49): NORMAL
150  VTAB (21): HTAB (1): PRINT "
     PRESS NUMBER TO TOGGLE SLOT,
      DRIVE OR": PRINT "MOTOR, OR
      TO GET INSTRUCTIONS OR EXIT
     ."
160  REM
170  REM GET DEFAULT DRIVE FROM L
     AST ACCESSED IN RWTS PARAMET
     ER LIST
180  REM
190 A =  PEEK ( PEEK (997) * 256 +
      PEEK (996)) * 256 +  PEEK (
      PEEK (1000) * 256 +  PEEK (
     999)):SL =  PEEK (A + 15) /
     16
200  VTAB (8): HTAB (10): PRINT "
     1 - SLOT ";: INVERSE : PRINT
     SL: NORMAL
210  REM
220  REM DEFAULT DRIVE IS 1
230  REM
240  VTAB (10): HTAB (10): PRINT
     "2 - DRIVE ";:A1 = 2: GOSUB
     570
250  REM
260  REM DEFAULT IS MOTOR OFF
270  REM
280  VTAB (12): HTAB (10): PRINT
     "3 - MOTOR ON/";: INVERSE : PRINT
     "OFF": NORMAL
290  VTAB (14): HTAB (10): PRINT
     "4 - INSTRUCTIONS"
300  VTAB (16): HTAB (10): PRINT
     "5 - EXIT"
310  REM
320  REM GET CHOICE
330  REM
340  VTAB (23):A =  PEEK (49152):
      IF A < 128 THEN 340: REM RE
     AD KEYBOARD FOR KEYPRESS
350 A$ =  CHR$ (A - 128):A =  PEEK
     (49168): REM RESET KEYBOARD
360  REM
370  REM GO DO IT
380  REM
390  IF A$ = "1" THEN  GOSUB 620:
      GOSUB 480:A1 = 2: GOSUB 570
     : GOTO 340: REM TURN OFF MOT
     OR, GET NEXT SLOT, SET DRIVE
      1
400  IF A$ = "2" THEN  GOSUB 620:
      GOSUB 570: GOTO 340: REM  T
     URN OFF MOTOR, TOGGLE DRIVE
410  IF A$ = "3" THEN  GOSUB 670:
      GOTO 340: REM TOGGLE MOTOR
     ON/OFF
420  IF A$ = "4" THEN  GOSUB 790:
      GOSUB 910: GOTO 130: REM  T
     URN MOTOR OFF AND PRINT INST
     RUCTIONS
430  IF A$ = "5" THEN  VTAB (16):
      HTAB (14): INVERSE : PRINT
     "EXIT": NORMAL : GOSUB 790: VTAB
     (21): CALL  - 958: VTAB (23)
     : END : REM TURN OFF MOTOR,
     CLEAR MESSAGE AND EXIT
440  PRINT G$: GOTO 340: REM INVA
     LID CHOICE - RING BELL (PRIN
     T ^G)
450  REM
460  REM  BUMP SLOT
470  REM
480 SL = SL + 1: IF SL > 7 THEN S
     L = 1
490  REM  CHECK FOR DISK CONTROLL
     ER CARD
500 A = 49152 + SL * 256: REM  C0
     00+SLOT*100H
510  IF  PEEK (A) <  > 162 OR  PEEK
     (A + 1) <  > 32 OR  PEEK (A +
     2) <  > 160 OR  PEEK (A + 3)
      <  > 0 THEN 480
520  VTAB (8): HTAB (19): INVERSE
     : PRINT SL: NORMAL
530  RETURN
540  REM
550  REM TOGGLE DRIVE
560  REM
570  IF A1 = 1 THEN  VTAB (10): HTAB
     (19): NORMAL : PRINT " 1/";:
      INVERSE : PRINT "2": NORMAL
     :A1 = 2:A =  PEEK (49291 + S
     L * 16): RETURN : REM C08B+S
     LOT*10H SELECT DRIVE 2
580  IF A1 = 2 THEN  VTAB (10): HTAB
     (020): INVERSE : PRINT "1";:
      NORMAL : PRINT "/2":A1 = 1:
     A =  PEEK (49290 + SL * 16):
      RETURN : REM  C08A+SLOT:10H
      SELECT DRIVE 1
590  REM
600  REM IF THE MOTOR IS ON TURN
     IT OFF AND WAIT UNTIL IT IS
     OFF
610  REM
620  IF A2 = 1 THEN  GOSUB 790: FOR
     I = 1 TO 1700: NEXT
630  RETURN
640  REM
650  REM TOGGLE MOTOR ON/OFF
660  REM
670  IF A2 = 1 THEN 790
680  REM
690  REM TURN MOTOR ON
700  REM
710  VTAB (12): HTAB (20): INVERSE
     : PRINT "ON";: NORMAL : PRINT
     "/OFF":A2 = 1:
720 A =  PEEK (49289 + SL * 16): REM
     C089+SLOT*10H TURN MOTOR ON

730 A =  PEEK (49293 + SL * 16): REM
     C08D+SLOT*10H SET UP FOR WRI
     TE PROTECT READ
740 A =  PEEK (49294 + SL * 16): IF
     (A < 128) THEN  GOSUB 790: GOSUB
     860: POP : GOTO 130: REM C08
     E+10H SET READ MODE AND SENS
     E WRITE PROTECTION - IF WRIT
     E ENABLED (LESS THAN 128) TU
     RN MOTOR OFF, PRINT ERROR ME
     SSAGE AND RESTART
750  RETURN
760  REM
770  REM TURN MOTOR OFF
780  REM
790  VTAB (12): HTAB (20): NORMAL
     : PRINT "ON/";: INVERSE : PRINT
     "OFF": NORMAL :A2 = 0
800 A =  PEEK (49288 + SL * 16): REM
     C0E8+SLOT*10H MOTOR OFF
```

```
810  RETURN
820  REM
830  REM DISK IS NOT WRITE PROTEC
     TED
840  REM PRINT ERROR MESSAGE
850  REM
860  HOME : PRINT G$;G$;G$;: VTAB
     (11): HTAB (5): FLASH : PRINT
     "THIS DISK IS NOT WRITE PROT
     ECTED": HTAB (6): PRINT "IT
     MUST NOT BE A CLEANING DISK"
     : NORMAL : REM RING BELL THR
     ICE AND PRINT ERROR MESSAGE
870  GOSUB 1150: RETURN : REM  WA
     IT FOR KEYPRESS
880  REM
890  REM PRINT INSTRUCTIONS
900  REM
910  HOME : PRINT : PRINT
920  PRINT "TO CLEAN YOUR DISK DR
     IVE, INSERT A"
930  PRINT "CLEANING DISK IN THE
     DRIVE AND TURN THE"
940  PRINT "MOTOR ON FOR 30 SECON
     DS."
950  PRINT
960  PRINT "TO CHOSE THE SLOT, PR
     ESS '1'.  THE NEXT"
970  PRINT "SLOT WITH A CONTROLLE
     R WILL BE SELECTED"
980  PRINT "(ON A 1 CONTROLLER SY
     STEM THE SLOT WILL"
990  PRINT "NOT CHANGE). IF A DRI
     VE IS ON IT WILL BE";
1000  PRINT "TURNED OFF.  DRIVE 1
      OF THE NEW SLOT"
1010  PRINT "WILL BE SELECTED."
1020  PRINT
1030  PRINT "'2' WILL TOGGLE BETW
      EEN DRIVE 1 & 2."
1040  PRINT "IF A DRIVE IS ON IT
      WILL BE TURNED OFF."
1050  PRINT
1060  PRINT "'3' WILL TURN THE MO
      TOR ON OR OFF."
1070  PRINT
1080  PRINT "'4' WILL GET THESE I
      NSTRUCTIONS."
1090  PRINT
1100  PRINT "'5' EXITS THE PROGRA
      M."
1110  GOSUB 1150: RETURN : REM WA
      IT FOR A KEYPRESS
1120  REM
1130  REM WAIT FOR KEYPRESS TO CO
      NTINUE
1140  REM
1150  VTAB (24): HTAB (5): INVERSE
      : PRINT "PRESS ANY KEY TO RE
      TURN TO MENU";: NORMAL
1160 A =  PEEK (49168): REM  C010
       CLEAR KEYBOARD STROBE
1170 A =  PEEK (49152): IF A < 12
      8 THEN 1170: REM CHECK FOR K
      EYPRESS
1180 A =  PEEK (49168): REM C010
      CLEAR KEYBOARD STROBE
1190  RETURN
63000  REM
63001  REM
63002  REM STEVE KNOUSE
63003  REM
63004  REM APRIL 30, 1982
63005  REM
63006  REM
```

----====*====----

WILDCAT COMPUTING

25% OFF SOFTWARE

* * * * * * * * * * *

TERRIFIC DISCOUNTS ON PERIPHERALS!

CHOOSE FROM THE FOLLOWING LIST OF OUR BEST SELLERS, OR SEND FOR A COPY OF OUR SIXTEEN PAGE CATALOG OF OVER 600 ITEMS. WE WILL SPECIAL ORDER UNLISTED ITEMS WHENEVER POSSIBLE. JUST CALL US!!

* *

ITEM	MANUFACTURER	LIST PRICE	WILDCAT PRICE
TAXMAN	H.A.L. LABS	23.00	17.25
MINI-FLEX DISK BOX	ADVANCE ACCESS	24.95	20.00
THE HOME ACCOUNTANT	CONTINENTAL	74.95	56.21
SUPERFAN II	R.H.ELECTRONICS	74.95	66.00
SOFTCARD PREMIUM SYSTEM	MICROSOFT	755.00	579.00
A2 DISK DRIVE W/O CONTROLLER	MICRO SCI	479.00	401.00
ELITE DRIVE W/O CONTROLLER	RANA	449.00	404.00
DB MASTER 3.0	STONEWARE	229.00	171.75
THE WORD HANDLER	SILICON VALLEY	250.00	187.50
APPLE STATION II	TRACE SYSTEMS	129.00	105.00
THE VOICE	MUSE	39.95	29.96
BAG OF TRICKS	QUALITY SOFTWARE	39.95	29.96
METEOROIDS IN SPACE	QUALITY SOFTWARE	19.95	14.96
TERRAPIN LOGO	TERRAPIN	150.00	112.50
VISICALC 3.3	VISICORP	250.00	187.50
VIDEOTERM	VIDEX	345.00	249.00
WIZARD-16 RAM BOARD	WESPER MICRO	149.00	116.00
MICROMODEM II	HAYES	379.00	279.00
REVERSI	QUALITY SOFTWARE	29.95	22.46
LOW RES RGB MONITOR	ELECTROHOME	570.00	513.00
SUPER COLOR CARD	ELECTROHOME	299.00	272.00
MICROLINE 82A PRINTER	OKIDATA	649.00	479.00

TERMS: *Immediate delivery with Money Order, Cashier's Check, or approved Master Card/Visa. Allow ten days for check to clear. NO REFUNDS. Exchange only for defective items returned within ten days. Add 3% for charge card orders; Texas residents add 5% state sales tax on hardware items; 3% shipping charge ($2.00 min.)*

WILDCAT COMPUTING • 3711 WOODRAIL DRIVE • PLANO, TEXAS 75074

TEL. (214) 424-9151

FORTH INTEREST GROUP
BY STEVE KNOUSE

Anyone interested in FORTH should turn in the membership survey indicating such. This is the only way to find out who you are.

The first meeting will be in a few months. A notice will be placed in the Apple Barrel and on the Hot Line and notices will be sent to those sending in the survey form.

At the first meeting there will be a public domain version of FORTH available. Until then you should get the following publications and become familiar with FORTH:

"Starting FORTH" by Leo Brodie

"FIGFORTH Installation Manual", $15,
FORTH Interest Group,
San Carlos, CA 94070

----====*====----

WANT/DON'T ADS

For Sale: VisiCalc 3.3 $150, Desktop Plan II $200, Dos 3.3 Controller $75, Silentype Printer $300.
Call John Williams at 652-5435 7:00 AM - 3:00 PM.

----====*====----

MEMBERSHIP DUES INCREASED

At the Executive Committee Meeting held on May 13 the committee unanimously agreed that the annual dues would be increased to $20 for adults, $10 for students through high school, and $25 for foreign members. The last increase was about two years ago.

MEMBERSHIP RENEWAL NOTICE

Membership expiration dates are printed on the Apple Barrel mailing labels. Expiration notices are not mailed out so be sure to check your label. Renew early to continue enjoying the benefits of HAAUG.

HINTS AND TIPS COLUMN TO START

Beginning with the next issue Steve Knouse will edit a hints and tips column covering programming techniques, hardware modification, printer activation, or whatever you would like to share with others. This is your chance to get into print without having to write an article. Contributions to this column should be mailed to Steve Knouse, 14150 Limerick, Tomball, TX, 77375.

ADVERTISERS' INDEX

CALL FOR ARTICLES

Articles and program listings should be submitted in hardcopy form, and, if possible, on disk in Applewriter][or ///, Professional Easywriter, Wordstar, Palantir, or Pascal compatible files. If not submitted on disk, articles must be free of typing or spelling errors as they cannot be retyped. Diskettes will be returned to the author provided his name and address are on them. Printed material should be printed using normal size characters, a new ribbon and enhanced print if your printer is so equipped. Margins should be set at 10 and 70 for articles. Listings should be printed in 40 column mode and may be printed in compressed print. Thermal paper should be avoided because it does not reproduce well. Articles should be submitted to:

Houston Area Apple Users Group
Apple Barrel
2218 Running Springs
Kingwood, TX 77339

ADVERTISING RATES

FULL PAGE	HALF PAGE	QTR PAGE	8TH PAGE	BUS.CARDS
$50	$28	$18	$10	$5

Advertisements should be submitted in camera ready form to H.A.A.U.G., 2218 Running Springs, Kingwood, TX, 77339, by the 10th of the month. Charges will be billed and accompanied by a copy of the Apple Barrel containing the ad.

Houston Area Apple Users Group
APPLE BARREL
2218 Running Springs
Kingwood, TX 77339

BULK RATE
U.S. POSTAGE
PAID
HOUSTON, TEXAS
PERMIT 3936

Postmasters:

Address Correction Requested
Forwarding and Return Postage Guaranteed

H.A.A.U.G.

Renew before: 06/27/820

Robin A. Cox
Ch Membership /HAAUG
5401 Chimney Rock #607
Houston, TX 77081