open apple gazette

Eighth Edition Volume 2, Number 2 Oct./Nov. 1983

PKASO Interface and Driver for Apple /// and Matrix Printer

by Jim Linhart

This PKASO system enables the Apple /// to print graphics out on a dot-matrix printer. I have an Epson MX-100 but the PKASO system also supports the Epson MX-70, the Epson MX-80 with Graftrax, and the Okidata Microline 80, 82, 83, 82A,and 83A, as well as the Centronics 739, the Integral Data Systems 560, and the Integral Data Systems Prism.

I have a 256K Apple /// with Business BASIC, one external disk drive and an Epson MX-100 printer. I got the PKASO (pronounced Picasso) system about a month ago and have spent that time putting it through its paces while at the same time learning more and more about the Apple /// graphics capabilities. It's nice to see something really pretty on your screen get printed out on the printer.

It was about a year ago when I purchased my Epson MX-100 printer. I picked the MX-100 over the MX-80 primarily because it can use wide scrap paper I could get from work. Its graphics abilities were at the time only incidental to me (that was to change!). The entire printer set-up consisted of:

1. The Epson MX-100 printer.
2. An Apple /// Universal Parallel Interface Card
3. An Installation and Operating Manual
4. A diskette containing a .PRINTER driver for the Epson.
5. A cable.

I went through my diskettes and reconfigured them by installing the .PRINTER driver. Then I installed the Universal Parallel Card in slot 1, hooked up the cable, then checked things out by booting a utilities disk. I sent a disk catalog not to the .CONSOLE but to the .PRINTER. The darn thing worked!! Now I can produce program listings, disk catalog listings, and even get into changing what print style the Epson is using. Consulting the Universal Parallel Card manual was primarily for checking that I was installing the card correctly.

I got the PKASO system a month ago, and I've found that as nice as the Epson is in printing words, its nicer still in printing out graphics. Graphics can be important because you can impart about three times the information on a chart or graph than you can in a page of figures. In fact, graphics figured in a great deal in a document a management consultant was preparing--his report was seventy pages in length. He included bar charts and graphs and couldn't imagine relaying the same voluminous amount of information just using the printed word.

Now, with the PKASO system, I am able to print such a report on my matrix printer and in one of three sizes, small, medium, and large. The only thing I don't have is the business or graphics software to generate the charts and pictures (although there is plenty out there to buy).

The PKASO system consisted of the following:

1. The PKASO interface card.
2. A PKASO /// Users Manual for Printer Driver for the Apple /// and Matrix Printers.
3. A PKASO Epson MX-80 Users Manual
4. A PKASO Driver Disk.
5. A PKASO Printer Interface Demonstration Disk. (Apple II)
6. A cable.

I've set up the system in these stages:

1. Installing the card.
   a. Plug in the card in slot 1.
   b. Connect the cable from the card to the printer.

2. Testing it in Apple II emulation. (This is a quick and easy test to make sure the printer, the card, and the cable are all working in harmony.)

   a. Get into emulation mode.
   b. Get into Applesoft.
   c. Boot the PKASO Printer Interface Demonstration Disk.
   d. Run the running demo program. Some of the graphics that result can be seen in Figures 1 thru 4.

3. Creating a disk to run on the Apple ///:

   a. Format a disk, calling it PABLOPKASO. Copy Business Basic onto this disk. Note that the PKASO system generates graphics with a Business Basic program

Figure 1

INTERACTIVE STRUCTURES INC. PRESENTS

GRAY SCALE

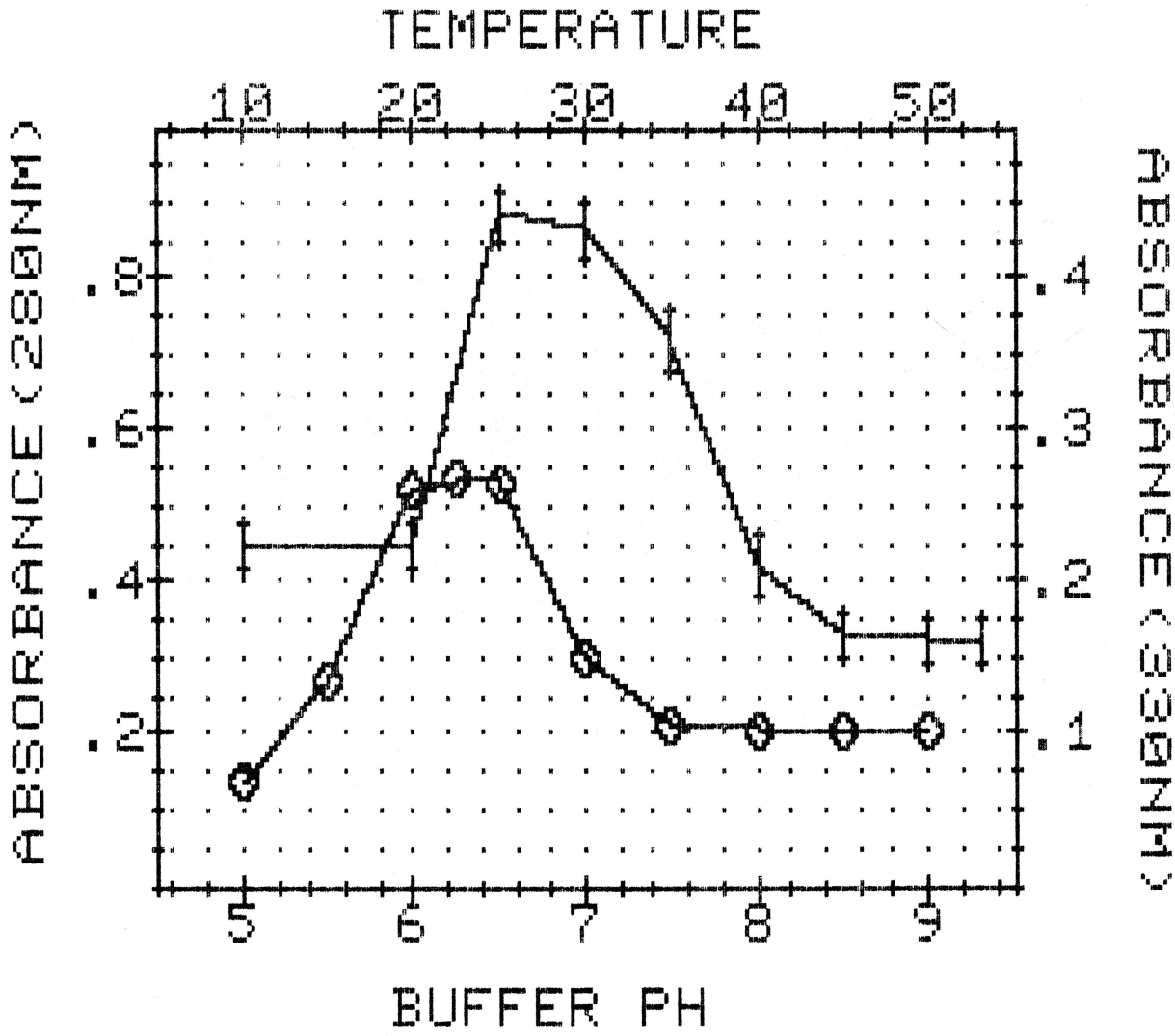
PRODUCED BY THE PKASO (TM) INTERFACE

THE 16 NIBBLE GRAY SCALE CHARACTERS:

0 1 2 3 4 5 6 7 8 9 J K L M N O

CORRESPONDING APPLE LORES COLOR VALUES:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15



called PRINTPIX.(Figure 5) You'll need to have Business BASIC to run this program.

b. Reconfigure the disk with the following:

1. .GRAFIX (if you don't already have it)
2. .PKASO.DRIVER from the PKASO DRIVER disk, calling it .PRINTER.
3. Follow the PKASO manual's directions for changing system parameters to make the .PRINTER driver fit the matrixprinter you own. In my case, the instructions for the Epson were found in Appendix E.

c. Copy the following over to PABLOPKASO:

1. PKASO.INV and PRINTPIX from the PKASO DRIVER disk.
2. PIE.PIC from the Apple /// System Demonstration disk.
3. BGRAF.INV from any Business BASIC disk.

d. Check PABLOPKASO's catalog. It should look like Figure 6.

You now have a disk, PABLOPKASO, from which you can use the PKASO in Apple ///'s native mode. It's important to know about Apple ///'s FOTO files, these are the files Apple /// stores graphic images on. It so happens that the System Demonstration disk which came with my system has a graphics presentation including reading in a FOTO file called PIE.PIC. When you run PRINTPIX and use PIE.PIC for your input, the picture that results can be seen in Figure 7. It's in Apple ///'s graphics mode 2 (560 x 192 Black and White.)

I really am impressed with the system and that I have access to both Apple II's and Apple ///'s graphics capabilities which are:

1. Apple II:

a. Lo-Res. 40x48, 16 colors.
b. Hi-Res. 280x192, 6 colors.

2. Apple ///:

a. Mode 0. 280x192, Black and White.
b. Mode 1. 280x192, 16 colors (restricted).
c. Mode 2. 560x192, Black and White.
d. Mode 3. 140x192, 16 colors (no restrictions).

I very much enjoy what this system has enabled my Epson to do. The only serious thing to complete this system is to get access to Apple /// software to create the graphics in the first place, then storing these screen images on FOTO files for PRINTPIX to read in and print from.

I wrote a small program called BUILDAPIX to generate FOTO files in any of the 4 modes you choose. It uses a Cursor /// to scribble on the screen, sort of a high-tech ETCH-A-SKETCH. If you don't have a joystick, then you substitute code of your own in the part of the program actually generating the graphics (lines 1290-1460.) Figure 8 is a listing of the program. Note that after I wrote this program, I stuck it onto PABLOPKSO.

Some oddities of the system I need to mention. Sometimes a picture looks better, more detailed when it gets printed on its side. Figures 9 and 10 are two printings of the same picture, one rotated ninety degrees, and the other not rotated at all.

Another quirk is what happens when you specify to PRINTPIX an incorrect graphics mode (remembering that Apple /// has four graphics' modes). This is the bar chart you've just seen, rotated and normal. It is in graphics' mode 3, 140 x 192 16 colors (no restrictions). When I asked PRINTPIX to print it, I specified mode 1, 280 x 192 16 colors (restricted). Figure 11 is an example of what happened.

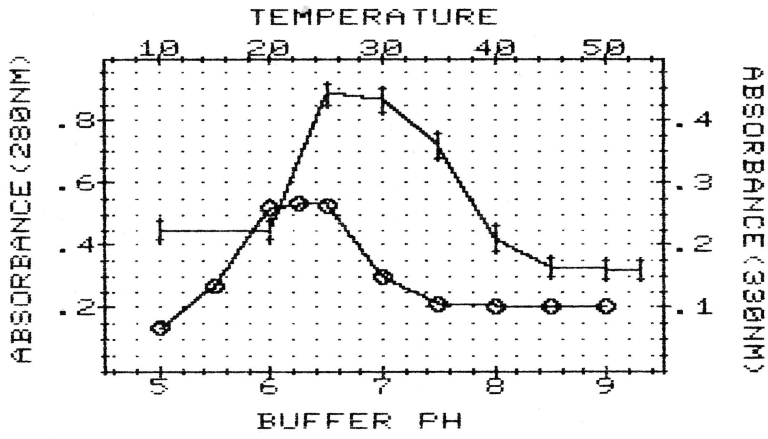
In summary, I am very much taken by the capabilities of the PKASO system. I have generated graphics yet am still able to list stuff from APPLEWRITER ///. My only regret is not knowing about this card before I got the Universal Parallel Interface Card.

There's one last thing I'd like to show you. Figure 12 is a small picture of an Apple II picture. Figure 13 is a tiny fraction of the same picture printed very large. Note that this shows a giant close-up of the woman's left eye!

INTERACTIVE STRUCTURES INC. PRESENTS

HIRES GRAPHICS

PRODUCED BY THE PKASO (TM) INTERFACE



MOVE PICTURE TO THE LEFT MARGIN AND USE INVERSE PRINTING

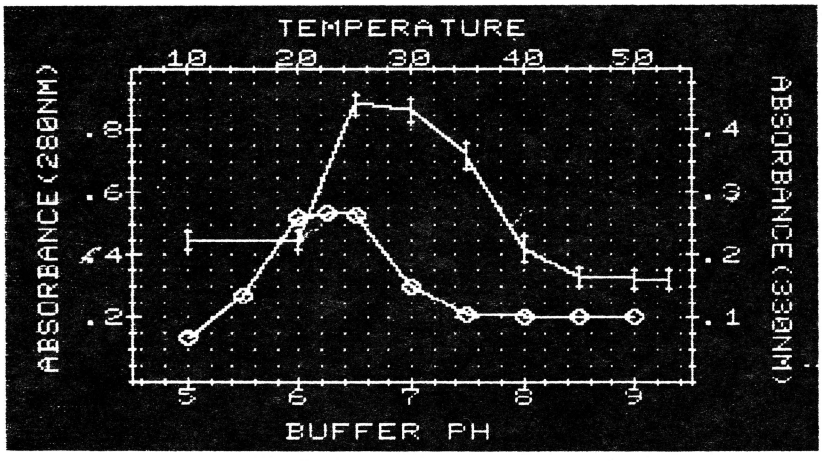


Figure 2

INTERACTIVE STRUCTURES INC. PRESENTS

LORES GRAPHICS

PRODUCED BY THE PKASO (TM) INTERFACE

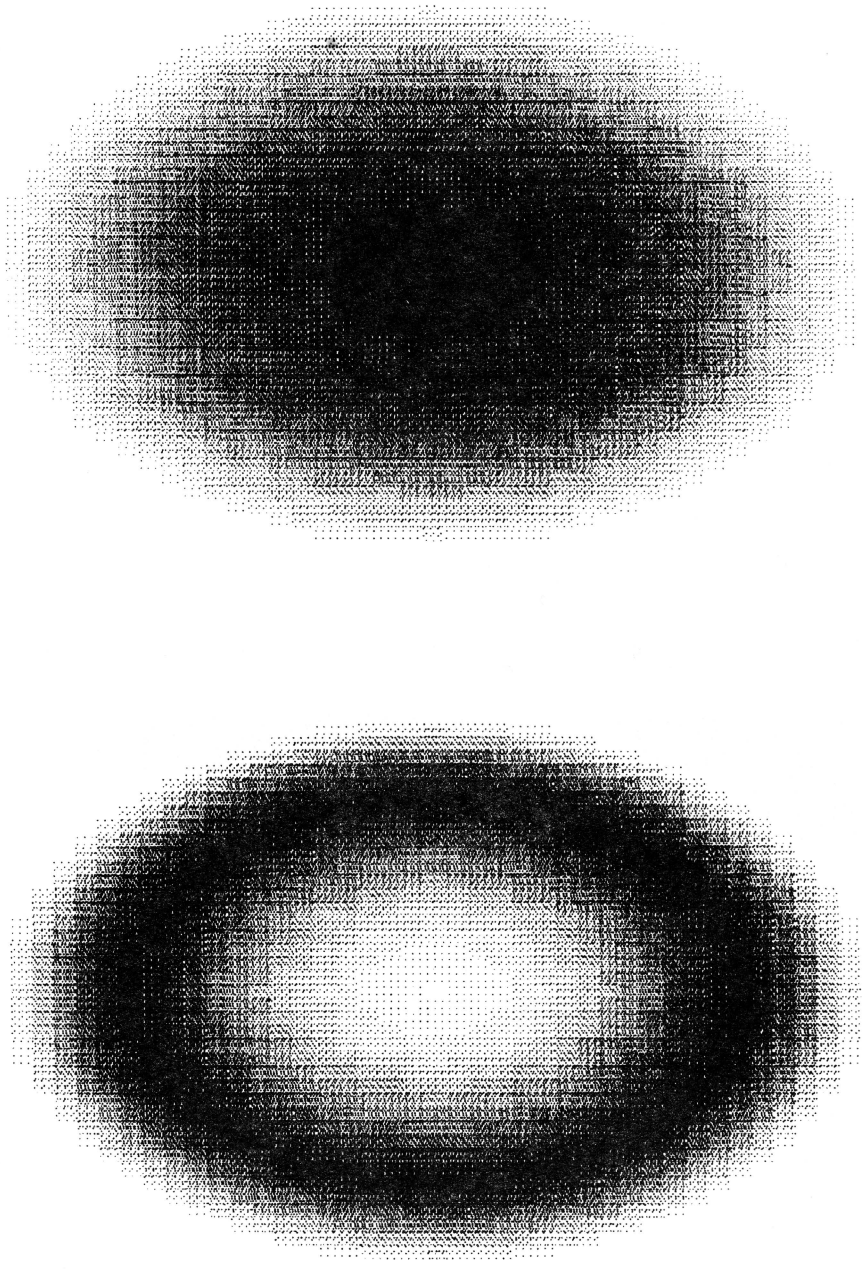


Figure 3

INTERACTIVE STRUCTURES INC. PRESENTS

A SPECIAL CHARACTER DEMONSTRATION

PRODUCED BY THE PKASO (TM) INTERFACE

FILE CONTAINS:

IS 1 2 3 4 5 6 7 8 9 0 $^{1 2 3 4 5 6 7 8 9 0}$ © µ ∩ ω ∆ π ° Å ± ● ☼ — TM CTRL

® ≤ ≥ ≠ ~ ≈ ∫ ⌠ ∮ λ σ ε Σ ¯ – θ

THIS IS HOW SPECIAL CHARACTERS CAN BE USED:

PKASO TM DEMO DISK – © IS 1981

HIT CTRL S TO STOP A BASIC LISTING.

THE SIGNAL IS 27.5 VOLTS ± 1.0.

THE TEMPERATURE IS PRESENTLY 54° F.

THE FORMULA FOR SULPHURIC ACID IS H_2SO_4.

VOLUME OF A SPHERE = $^4/_3 \pi R^3$.

FOR WHAT ANGLE IS SIN θ = 1?

A ≠ B IF AND ONLY IF B ≤ 7.

4^{-2} WORKS OUT TO .0625.

Figure 4

"PRINTPIX" Listing

```
5   PP$= PREFIX$:REM SAVE CURRENT
7   prdev$=".printer":REM change for a different device name
20   TEXT:HOME
30   PRINT CHR$(16)0
40   VPOS=08:HPOS=06:PRINT"Interactive Structures Inc."
50   VPOS=12:HPOS=09:PRINT"PKASO /// - PKASODMP"
60   VPOS=16:HPOS=04:PRINT"FOTO File Graphics Dump Utility"
65   ON ERR OFF ERR:GOTO 8000
70   INVOKE"BGRAF.INV","PKASODMP.INV"
80   OFF ERR:TEXT:HOME
100   WINDOW 1,20 TO 80,24
105   cp$= PREFIX$
110   PRINT"Current Prefix: " PREFIX$
115   PRINT:INPUT"Enter new Prefix, <RETURN> for same Prefix, or <*> to end: "
      ;a$
125   IF a$="*" THEN TEXT:HOME:GOTO 8999
130   IF LEN(a$)=0 THEN 200
140   ON ERR GOTO 190
145   flag=0
150   PREFIX$=MID$(a$,1,LEN(a$)*(flag=0))+MID$(pp$,1,LEN(cp$)*(flag<>0))
155   IF flag THEN OFF ERR:GOTO 100
160   OFF ERR:GOTO 200
190   flag=1:RESUME
200   WINDOW 1,1 TO 80,19:HOME
205   ON ERR OFF ERR:HOME:GOTO 8900
207   WINDOW 1,20 TO 80,24
210   OPEN#1, PREFIX$:INPUT#1;a$:PRINT a$:INPUT#1;a$:INPUT#1;a$
215   n=0:ll=3:mm=1:WINDOW 1,1 TO 80,19
217   IF mm>80 THEN mm=1:ll=ll+1:n=n+1:IF n>90 THEN WINDOW 1,20 TO 80,24:PRINT
      :INPUT"<Return> to continue: ";a$:WINDOW 1,1 TO 80,19:n=0
220   INPUT#1;a$:IF LEFT$(a$,5)="     " THEN 250
230   IF MID$(a$,3,5)<>"FOTO " THEN 220:ELSE VPOS=ll:HPOS=mm:PRINT MID$(a$,16,
      16);:n=n+1:mm=mm+16:GOTO 217
250   ON ERR GOTO 300
260   PRINT a$:INPUT#1;a$:GOTO 260
300   OFF ERR:CLOSE
310   IF n=0 THEN PRINT"No FOTO files found!":FOR i=1 TO 1000:NEXT i::HOME:GO
      TO 8900
315   cp$= PREFIX$
320   WINDOW 1,20 TO 80,24:HOME
325   PRINT"Current prefix: " PREFIX$
330   INPUT"File: ";a$:IF LEN(a$)=0 THEN 100
340   ON ERR OFF ERR:GOTO 320
350   PERFORM initgrafix:PERFORM gload.a$
360   OFF ERR
370   PERFORM grafixon:INPUT"";a$:PRINT CHR$(15);
380   INPUT"<RETURN> to print, <*> to change Prefix, or type a new filename: "
      ;a$:IF a$="*" THEN 100:ELSE IF LEN(a$)>0 THEN 340
400   WINDOW 1,20 TO 80,24:HOME
410   INPUT"Enter picture mode (0..3): ";pmode:IF(pmode<0) OR(pmode>3) THEN 41
      0
420   INPUT"Enter select size (0..2): ";size:IF(size<0) OR(size>2) THEN 420
430   INPUT"Rotated? ";a$:rot=0:GOSUB 8910:IF flag THEN rot=4
440   INPUT"Direct Black and White correspondance? ";a$:inv=0:GOSUB 8910:IF fl
      ag THEN inv=64
450   INPUT"Randomized Gray Patterning? ";a$:rand=0:GOSUB 8910:IF flag THEN ra
      nd=128
```

Figure 5

"PRINTPIX" Listing (con't)

```
460   HOME
465   PRINT"Print code: "pmode+size*8+rot+inv+rand"  ";
470   PRINT"Mode: "pmode"  Size: "size"   ";
472   IF rot<>0 THEN PRINT"Rotated  ";:ELSE PRINT"Not Rotated  ";
474   IF inv<>0 THEN PRINT"Direct   ";:ELSE PRINT"Inverted  ";
476   IF rand<>0 THEN PRINT"Randomized";:ELSE PRINT"Fixed Patterns";
478   PRINT
480   INPUT"<RETURN> to print, <C> to change these settings, <*> to select new
       file: ";a$:IF a$="*" THEN 320:ELSE IF a$="c" OR a$="C" THEN 400
490   PERFORM bpkasodmp(@prdev$,%pmode+size*8+rot+inv+rand)
492   INPUT"Another dump of this picture? ";a$:GOSUB 8910:IF flag THEN 400:ELS
      E GOTO 320
8000   HOME
8005   VPOS=8
8010   PRINT"BGRAF.INV or PKASODMP.INV not found!"
8015   PRINT
8020   PRINT"Use your SYSTEM UTILITIES disk to copy"
8025   PRINT
8030   PRINT"BGRAF.INV from your Business Basic disk,"
8035   PRINT
8040   PRINT"PKASODMP.INV from your PKASO /// disk,"
8045   PRINT
8050   PRINT"onto your working disk.":PRINT:PRINT
8060   END
8900   PREFIX$=cp$:GOTO 100
8910   flag=0:a$=LEFT$(a$+" ",1):flag=(a$="y") OR(a$="Y"):RETURN
8999   PREFIX$=pp$:END
```

Figure 5

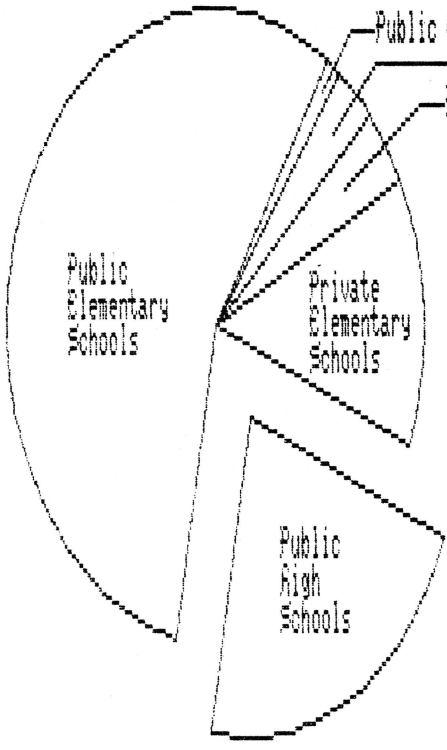
```
PABLOPKASO          Size   Modified  Time  File type  Eof Phys

  *SOS.KERNEL         43    1-Nov-82  0:00  Sosfile    512   44
  *SOS.DRIVER         41    5-May-00  0:00  Sosfile    512   42
  *SOS.INTERP         50   11-Sep-81  0:00  Sosfile    512   51
  *PIE.PIC            32   11-Sep-81  0:00  Fotofile   512   33
  *BGRAF.INV          20   11-Sep-81  0:00  Codefile   512   21
  *PRINTPIX            5   <no date>  0:00  Basicprog  303    6
  *BUILDAPIX           4    5-May-02  0:00  Basicprog   50    5
  *PKASODMP.INV       12   22-Mar-82  0:00  Codefile   512   13

 8 files listed, 58 blocks available
```

Figure 6

DISTRIBUTION OF U.S. SCHOOLS BY GRADE LEVEL
(Source: U.S. Dept. of H.E.W., 1977)



Level of School	Percent
Public Elementary	56%
* Public High Schools	22%
Private Elementary	14%
Private High Schools	4%
Private Colleges	2%
Public Colleges	1%

Figure 7

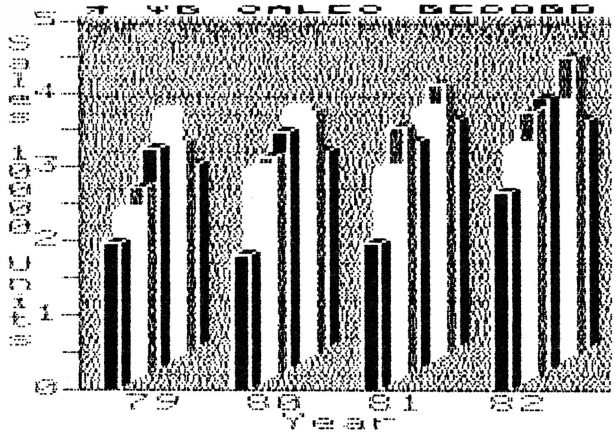


Figure 9

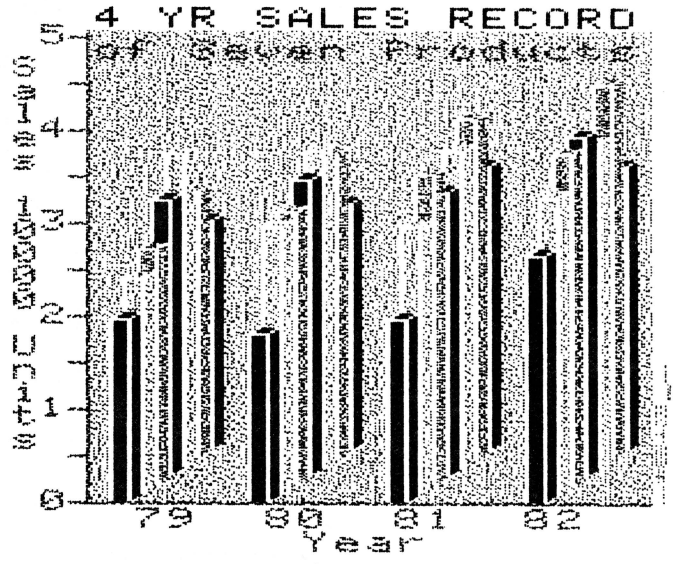


Figure 10

"BUILDAPIX" Listing

```
1000   FOR i=1 TO 24:PRINT" ":NEXT i
1010   VPOS=04:HPOS=13:PRINT"PROGRAM TO CREATE A FOTO FILE"
1020   VPOS=07:HPOS=25:PRINT"This is a utility program to create a FOTO file"
1030   VPOS=11:HPOS=25:PRINT"    0 -- B&W280 Black & White 280x192"
1040   VPOS=12:HPOS=25:PRINT"    1 -- COL280 Color         280x192"
1050   VPOS=13:HPOS=25:PRINT"    2 -- B&W560 Black & White 569x192"
1060   VPOS=14:HPOS=25:PRINT"    3 -- COL140 Color         140x192"
1070   VPOS=09:HPOS=25:PRINT"WHICH GRAFIX MODE WOULD YOU LIKE"
1080   VPOS=09:HPOS=57:INPUT A$
1090   MODE=99
1100   IF A$="0" THEN MODE=0
1110   IF A$="1" THEN MODE=1
1120   IF A$="2" THEN MODE=2
1130   IF A$="3" THEN MODE=3
1140   IF MODE=99 THEN GOTO 1070
1150   VPOS=17:HPOS=25:PRINT"WHAT WOULD YOU LIKE TO NAME YOUR FILE"
1160   VPOS=17:HPOS=62:INPUT B$
1170   VPOS=19:HPOS=25:PRINT"Your file will be GSAVEd with the name of"
1180   VPOS=20:HPOS=25:PRINT b$;".MODE";mode
1190   VPOS=21:HPOS=30:PRINT"(hit enter to continue)"
1200   VPOS=21:HPOS=53:INPUT c$
1210   OPEN#1,".GRAFIX":INVOKE"BGRAF.INV"
1220   buffer=1
1230   PERFORM INITGRAFIX:PERFORM GRAFIXMODE(%MODE,%BUFFER)
1240   COLOR%=3
1250   IF MODE=0 THEN COLOR%=0
1260   IF MODE=2 THEN COLOR%=0
1270   PERFORM FILLCOLOR(%COLOR%):PERFORM FILLPORT
1280   PERFORM grafixon
1290   REM this program uses a CURSOR /// to draw lines on a screen.
1300   REM When the CURSOR ///'s button is pushed, drawing is stopped
1310   REM and the screen is GSAVEd to a FOTO file.
1320   REM
1330   PERFORM moveto(%PDL(0),%PDL(1))
1340   IF mode=0 THEN xmult=280/256
1350   IF mode=1 THEN xmult=280/256
1360   IF mode=2 THEN xmult=560/256
1370   IF mode=3 THEN xmult=140/256
1380   ymult=192/256
1390   REM
1400   REM
1410   REM LOOP to draw lines.
1420   REM
1430   IF BUTTON(0)>0 GOTO 1470
1440   x=xmult*PDL(0):y=ymult*PDL(1)
1450   PERFORM lineto(%x,%y)
1460   GOTO 1430
1470   TEXT
1480   D$=B$+".MODE"+A$
1490   PERFORM GSAVE.D$
1500   END
1510   REM This program was written May 7th, 1983 by Jim Linhart
1520   REM of Berkeley, California and is donated to the Original
1530   REM Apple ///'ers Users Group.  It is in the public domain.
```

Figure 8

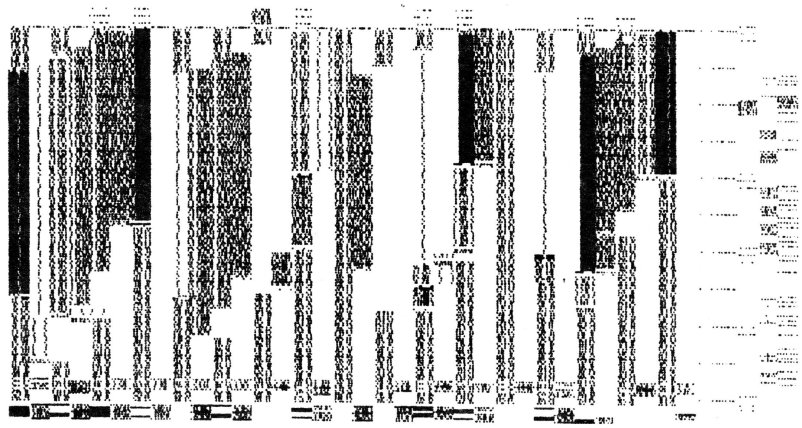


Figure 11

Figure 12

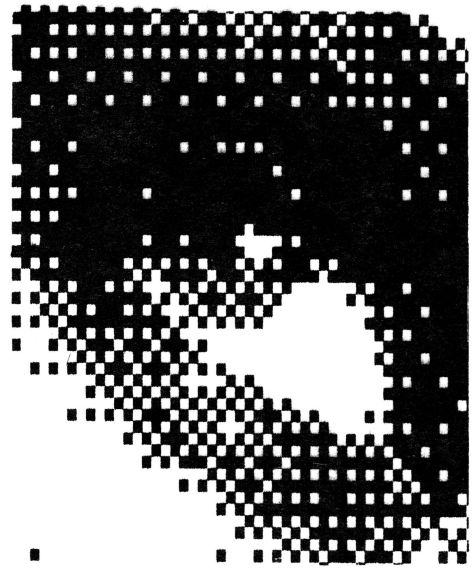


Figure 13

APPLEWRITER ///

by Rudolph H. de Jong, M.D.

AppleWriter /// (AW3,) written by Paul Lutus, is copyrighted and supported by the Apple Company. Its $225 price tag is in the middle range of comparable word processor (WP) packages. Like Quark's Word Juggler (not tested), **AW3** is uniquely powerful because it is tailored for Apple /// hardware and the SOS operating system.

For instance: SOS, device drivers, **AW3** program code, and text deletion and insertion buffers are loaded into one 64K memory stack, leaving a second 64K stack completely free for text. Compare that with an Apple][WP package in 48K emulation mode, or a product designed for SoftCard CP/M for instance; you'd be lucky to have more than 32K working core left over for text. With all editing done in core, those frequent and annoying delays for in-and-out disk swapping are avoided altogether. **AW3** thus is well suited to lengthy manuscripts and fast turnaround.

AW3 is a screen-oriented WP written in assembly language, which makes it fast and responsive for it does away with plodding language interpreters. A copy-protected master disk and backup are provided with the package. Following the 90-day warranty, defective disks can be replaced for $15. Regrettably, there is no hot-line for consultation, intimidating potential buyers. Noteworthy is a utility disk that enables ready exchange of files between AW2, **AW3** and MailList Manager. QuickFile /// files too can be merged with **AW3**, operating as a simplified MailManager list that can be merged with a form letter.

By experimentation, I found that **AW3** can be used to edit Business BASIC text files. In fact, you can write a BASIC program with **AW3**, then save it to disk. Boot BASIC and load the (text) file with the EXECute command. Make whatever syntax corrections are needed, RUN the program to see that all is well, then store the file as a BASIC program with the SAVE command. In writing BASIC program text, the split-screen feature is fantastic as you can hold your place and scroll the text in the bottom viewport till you find the line number for a GOTO or a formating statement. In going through my back issues, I found an article in the September 1982 "Gazette" by Paul Wilson; it provides all the details for using **AW3** as a BASIC Editor.

The **AW3** manual is poorly printed, looking as if it were photo-copied typewritten text; but it is crisply written, detailed and quite complete. Lacking a decent tutorial section, beginners will have tough sledding getting started. Fortunately, a "Product Training Pak" disk and tutorial manual go far at making **AW3** a lot friendlier (for an extra $40.) It took my wife just three 45-minute sessions with the Pak to compose, edit, correct, save, load and print a business letter. Perhaps the weakest point of the **AW3** manual is the lack of a good index, though the detailed table of contents and the handy tear-out summary of commands table compensate to some extent.

AW3 does just about everything expected of a complete WP-package. Editing, correction, deletion, insertion, global searches, cut-and-pasting, saving and loading all are executed by one-stroke mnemonic control commands. An extensive Help menu is available for the ten major WP operations. The split-screen feature is helpful, as when inserting text blocks in distant paragraphs. Underline, bold-print, sub- and super-scripting, shadow-print and other printer instructions are easily executed -- and enormously simplified -- with the handy "Glossary" (CTRL-G) command. A personal glossary is easily built so that a single symbol can represent a complete character string. For instance, I defined the up-arrow to send superscript, and the down-arrow subscript, escape-codes to the printer.

A pity that **AW3** does not provide user-definition of special keys to execute common command functions. For instance, erasing of characters, words or paragraphs proceeds only from right to left. Hence, the cursor must be moved to the end of a string before it can be erased. As the search-and-find commands leave the cursor in front of the string, time is lost in re-positioning the cursor. There is a reason, of course, for when the direction of the deletion command is reversed, the character, word or paragraph is regurgitated for insertion elsewhere in the text.

Included in the **AW3** package is an unsual 'plus', the word processing language "WPL", a batch-processor that executes all keyboard editing and printing commands from a text

file. With WPL, you can automate procedures like renumbering of codes, searching and replacing text segments across files, merging of address lists with a formletter, and so on. WPL takes some experimenting to learn well; it seemed included in the manual as an afterthought with rather off-hand rambling instructions. But learning WPL is time well spent! The WPL examples provided on the master disk are immediately applicable, as in putting together a mailing of formletters.

Printer formating options are numerous and extensive, and can be saved for later retrieval. Additional printer formating commands such as centering or formfeed can be embedded in the text, making for flexible, powerful, yet easily executed control of output. Control and escape command sequences for a particular printer likewise can be embedded, a task made much lighter by the glossary option. I have used both the Silentype and the Spinwriter printers to their full capability. Text-justification includes a 'fill' option to expand spaces between words for smooth left and right margins.

True proportional spacing depends on a printer so equipped, but fill justification is an attractive and inexpensive alternate. "Keeps" to reserve space for tables, figures or footnotes are easily executed with conditional form-feeds, where the paper advances if there isn't room left to print the next number of designated lines. In fact, provision is made to stop printing at designated text locations for display of a message on the screen. This allows insertion of a figure, for instance, or change of a printing wheel.

Screen text is displayed as 24 lines of 80 characters, with provision for slow or rapid scrolling up or down. The topmost display line is occupied by an inverse-printed "data-line" showing memory available, length of text, file name and special options selected. If you need the extra line, the data line can be toggled in or out with the escape key. Text wrap-around is implemented, and likewise can be toggled. Prior to hard printing, you can review page formating commands on the screen by "printing" to the console driver.

One sorely missed feature is a line counter, you can't tell where a new page begins without first printing a test run. Lacking a line counter, "widows" (2 or 3 lines of a new paragraph at the bottom of one page, then continued on the next) are difficult to predict. That problem is easily solved, however, with the printer instruction [.FF5] 'form-feed if fewer than 5 (or whatever) lines are left on the page', as I did in this manuscript. Along the same line, the screen doesn't show embedded commands such as underline, superscript, bold-facing, etc. In other words, you don't see what you get till you print it.

AW3 is specifically written for the Apple ///, so uses unique SOS features such as time-stamping of files, modification of printer drivers, selection of special character fonts, etc. **AW3** makes full use of Apple /// memory, thus is fast and responsive. With purchase of the Training Pak, the basic editing and printing commands are readily learned, preparatory to study of the detailed manual. **AW3** is a sophisticated word processor, but it will require time, and some experimentation, to be fully mastered. An **AW3**-compatible spelling dictionary is not available. That, of course, prevents one from inserting "soft" hyphens in the text.

I found the learning effort worthwhile because **AW3** is a powerful word processor that takes full advantage of the Apple /// hardware and SOS operating system, and at a reasonable price. **AW3** also does an excellent job of working on the text files generated by other Apple software such as ACCESS /// or QUICKFILE ///. You can even use it to write lenghty BUSINESS BASIC programs.

If Apple could put its skilled writing teams to work revising and upgrading the manual, it would have an outstanding WP package for the business world. At the very least, Apple would be well advised to include the Training Pak in the package at no additional cost to the purchaser. Till then, and without apparent factory interest or support, less powerful but friendlier packages likely will prevail. At the moment, **AW3** is best suited to the full-time author who needs the many unusual and practical features provided, and is willing to invest a bit of extra effort.

Pascal News for the ///

by Dennis Cohen

Apple has announced (and delivered) a number of long-awaited Pascal products for the ///. Trying to take these in some sort of haphazard order, "The Pascal Technical

Reference Manual" (order no. A3L0006) is back from the printers and being distributed; Record Processing Services (order no. A3D0018) is now available from the dealers; and the Pascal 1.1 upgrade is also being readied for distribution.

I've received my copies of RPS and the PTRM and am quite pleased with them. The PTRM is a well-written, technical reference that answers (practically) every question I had concerning Pascal on the ///. Included with the manual is a disk containing Tim O'Konski's "SOSIO" unit (see BYTE, Dec. 82) in both source and code form for use as either functions or procedures as well as a supplement pertaining to this disk.

RPS is a very nice collection of routines to assist the programmer wishing to develop an ISAM database. The major advantage of RPS is that Apple uses the package as its standard, internal database interface. This means that packages that use RPS will be compatible with products from Apple.

The Pascal 1.1 release is being handled in a new manner. Rather than leaving the upgrade process to the dealers, a low-profit activity for them and one that they do not like, Apple is setting up a PO Box to which you send your check and proof of purchase (original Pascal 1 disk). After a (hopefully) short wait, Apple will send back a new set of disks and (all new) manuals. Some of the features in the upgrade are:

1. PMOVE - a utility to move your SYSTEM disk (e.g. to a Profile)
2. SANE - the Standard Apple Numeric Environment, an intrinsic unit that allows 64-bit integers; 32-bit, 64-bit, and 80-bit reals; and string <--> real routines. This package is completely IEEE Floating Point Standard compatible.
3. Extended Library Search -- This means that you can construct an ASCII file containing the names of libraries to be searched, which extends the Program Library concept.

Apple is not the only one with new products for our favorite. Volition Systems now has a complete Modula-2 system running under SOS. This compiler is compatible with Pascal, which presents some very exciting prospects to the innovative programmer. The Modula package includes some interesting examples in source form,a UNIX look-alike called pNix (including pipes and the Shell), a screen window manager, and others. Though Apple isn't saying anything officially (as usual), they do seem quite interested in the health of Modula on their products. My copy of Modula is scheduled for shipping next week according to Volition, so I should have a follow-up soon.

There is also a very fine Apple /// magazine being published in Ventura, CA by Bob Consorti named "On Three". I've just received the third issue and it keeps improving. The articles attack everything: BASIC, Pascal, Assembly Language, Emulation Mode, Apple Writer ///, WPL, Visicalc, etc. For those who don't like to spend the time typing in programs listed in magazines, there's the Disk of the Month containing all the programs and routines in that issue for a nominal fee ($9.95 + 1.50S/Ht). The subscription rate is $30 for twelve issues. Contact "On Three", PO Box 3825, Ventura, Ca 93006, (805) 644-3514.

If the Pascal (and Modula) programmers among you are looking for birds of a feather, the UCSD p-System User Society (USUS) is an organization to be seriously considered. They have a very active Apple SIG with ///rs holding both co-Chairs and the Technical Liaison to Apple. The last two Chairmen were also ///rs (one of them is now the Apple Computer Company's Pascal Product Mgr.). This is a non-profit organization and Apple is packaging USUS membership applications in theirPascal upgrades.

The Portable A///: The Radio Shack Lap Computer as a Peripheral

By Ed Horodko

I remember the first time I saw a TRS-80 Model 100. A computer to put in your lap. A new gimmick, I said to myself. I glanced again. It didn't look junky. But it surely couldn't do much since it was so small. I was in your typical Radio Shack, so I figured the salesman wouldn't know computers. I bought my batteries and left. But the seed was planted.

In a week or so I was in a different R.S. and there was that little computer again. The size of my dictionary, but less than two inches thick, it just sat there in the shiny display case. "$799," it said.

Okay, I'm an Apple /// user. I even was the first kid on my block with 256K, PROFILE, and CATALYST. But I asked anyway. I felt like a kid in a candy store asking where the prophylactics were but I asked anyway.

The salesman took it carefully from its shrine. "These are great," he said. "What could he know," I thought.I asked him one question after another. He answered them all. All the answers were good answers.

How big is the Liquid Crystal Display? 40 characters by 8 lines. To scroll, just push the arrow buttons. Are those function keys? Yes, each program defines and electronically labels up to eight of them.

What do you need to get it to print? Just about any parallel printer. Does it have a modem built into it? A direct-connect modem. Where do you load a program from? You probably won't have to; this computer comes with integrated softwareon built-in ROM. Just select the program or document you want from the main menu.

What are the built-in programs? A word-processor, a communications program, a telephone directory that can even dial for you, an address program that can print labels, a scheduling program that lets you search a notebook data base, and a basic language you can write in yourself.

"Oh," I said. My skepticism was waning. "Don't you have to write to tape or disk or something?" If you want to, you can save to a normal portable cassette or load from it. But the easy way is to use your RAM as your storage. It comes with 8k and is expandable to 32K.

You can also up and down load ASCII Text Files easily using its communications program and its RS-232 Serial Port. There's even a BAR CODE READER PORT for future program and data loading, but we don't sell the wand or programs yet.

"What about power?" "If you stop keyboarding for more than four minutes, it shuts itself off. When you next turn it on, it continues where it left off. It runs for a week or so of normal use on four AA batteries. Self-charging interior ni-cads keep your RAM alive for a day or two, long enough to get more batteries. We'll have a $6 AC adapter for it in a few weeks."

Out of questions, I meekly left for my car to think it over.

The TEXT EDITOR (by MICROSOFT, like the rest of the software) was crude by WORD JUGGLER or APPLEWRITER standards, but by normal micro standards, it was very good as far as it went. Easy cursor movement, block moves, string search (but not replace), automatic insertion, easy deletion... considerably better than pen and paper, my current briefcase method.

The Communications program was a delight. Put your phone book in it. Searchfor a string and there's your number. If you've got your computer plugged into a phone line, push one more key and the computer will (rotary) dial the number for you.

Or using your modem-eliminator, RS-232 direct-connect the Model 100 to your Apple ///, run ACCESS/// or the like, and agree with yourself on your protocol (I like 9600 baud, 8 bits, No parity, 1 Stop bit, X-on/X-off enabled). You can now shuffle ASCII Text Files to and from a computer that fits in your briefcase sideways.

And now, that's what I do. And it's great. I originate documents on either computer, work on the /// when I'm near it, work on the Model 100 when I'm not. Its display is upper and lower case, and it has an elaborate set of graphics characters that could be useful if you worked at it.

The full-sized keyboard is a little noisy but feels good, though it has no friendly little dot on the K and D. One button turns a section of the keyboard into a numeric keypad. I haven't been able to get any of its programs to crash, though it has several levels of RESET for recovery.

Naturally, there are limitations to the Model 100:

-Its maximum storage RAM is a shade over 28,000 bytes (if you get it fully loaded for about $1150). So good housecleaning is imperative.

-If you forget which way is up or down when you're up or downloading, you may delete your file instead of transfer it.

-You want to be careful not to drop it. It's a computer.

-You can't (yet) crunch numbers on it, though you can use it as a calculator in Basic or write a math program yourself. (Expect a Something-Calc ROM in early '84; but conversion may cost you 8k of RAM.)

-Your Model 100 data base program is no substitute for PFS, ThinkTank, etc., so lots of A/// tasks are not transportable.

-Overall, it's a little slow compared to what I'm used to on my other computers. But its type-ahead buffer helps.

It's not an A///. And because of its awkwardness at large scale storage without another computer to plug it into, I wouldn't recommend it for solo business applications other than executive notebook/phonebook.

But as an Apple /// peripheral, it's fantastic.

Member Inquiries About Publication Delays:

Since our last renewal mailing, we have received many inquiries from members regarding delays in publication of The Open Apple Gazette. These letters range in nature from the curious to the suspicious to the down-right irate. In this editorial, I hope to provide some insight into our production procedure thereby satisfying the curious, de-mystifying the suspicious and calming the irate among you.

When we began to publish The Open Apple Gazette, we sincerely hoped to produce a bi-monthly newsletter that would ideally be twenty-four pages in length. Those of you who have been with us since the beginning have seen the Gazette grow from the spindly eight pages of Volume One, Number One, to our most recent and highly informative Volume One, Number Six, which numbered twenty-four pages.

We realized we would need time to build-up a membership base to generate enough contributed articles to produce a quality, bi-monthly newsletter of twenty-four pages. Starting with Volume One, Number Two, the Gazette has been a consistent twenty-four pages.

In terms of production expense, our printing costs are about the same for twenty-four pages as they would be for six or eight pages. However, the time spent in gathering up the articles for publication, editing them and performing the lay-out didn't seem justifiable for an inferior production of six or eight pages.

The period of silence between Volume One, Number Five, and Volume One, Number Six, was due in part to some necessary, internal re-organization. The individual who was helping with the club's mailings and newsletter production walked out one day and was not heard of again. We have since expanded our staff to accommodate the overwhelmingly positive response we have received from Apple /// owners to our organization. In March of this year we relocated to a larger facility since we had virtually outgrown our former office.

The above explanations are in and of themselves reason for some delay. They do not however, include one essential aspect: the lack of contributed articles from our members. The purpose of our newsletter is to provide much-needed, quality information to you, our members.

The majority of this information must come from you, our reading audience. We are interested in anything you have to contribute having to do with your experiences with the ///, or any of its software programs or hardware attachments. The more contributions we receive the more frequently we will be able to produce a quality, twenty-four page publication.

Our newsletter is only as outstanding as we collectively work to make it. We have instituted a new policy of sending members who have programs or articles to contribute two blank diskettes. One diskette they may keep for their own personal use while on the other they send back to us the program or article they wish to contribute. We have seen a very positive response to this procedure and will continue it. All you have to do is let us know that you have a program or an article for the Gazette and we will mail the diskettes to you.

You belong to a club which offers more than a newsletter. Some of you will remember when our Public Domain Library consisted of only one Business BASIC .001 and one VISICALC Template .001 diskette. We now have a library containing over ten software programs. Obtaining software for the /// has always been a frustating experience. As members you have access to these valuable programs created and used by /// owners and programmers. In addition, with our combined buying power we are able to offer you discounts on hardware like the GAMEPORT III and the MICRO-SCI disk drives.

Another service the group performs for its members is in answering the many inquiries we receive daily. If we do not know the answer to your question, we get in touch with the people who do so you will receive the information you need. As we have grown, so has the volume of letters we receive from our many readers worldwide. Many of you have found this service extremely helpful especially when dealers and other sources have been unable to give you the answers to your questions.

We need to work together to help spread information about the ///. Your contributions and continued support will be greatly appreciated.

WORD JUGGLER VS. APPLEWRITER ///: A Worm In My Apple?

by Pamela Whitty

Last summer, I was introduced for the first time to an Apple /// computer. Had anyone asked me if I ever thought I would some day operate one, I would have said unequivocably "Never". Computers were so foreign to me, they have a language all their own, and I guess I was terrified that I couldn't keep up with everyone else. Given no choice in the matter by my employer, however, I made-up my mind that it had to work for me, no matter what the obstacles.

Since I am a secretary and my skills are centered around typing letters and other forms of text, I was mainly interested in the word processing software packages that would be available to me for the Apple ///. I was given **Apple Writer ///** to learn on. I met my first obstacle. I have to be frank when I say that I did not find this to be a very friendly piece of software.

The **Apple Writer ///** package includes an operating manual and tutorial. The tutorial is a new addition and is much easier to understand than the operating manual. In my opinion, the **Apple Writer ///** operating manual was written by computer programmers for computer programmers. The language is simply too difficult to be understand by someone who does not know a lot about computers. There really is no rhyme or reason to the order of the book. The table of contents is not much better. You must have your manual at your side at all times. Here again, the language of the commands is such that it makes it very difficult to memorize any of them.

I suppose my biggest complaint is that I am unable to display my document to the screen without going back to my print set-up menu and changing my print commands. This is inconvenient and time-consuming, especially if it is a long document and I need to view it several times before I print it.

After many futile attempts of trying to master Apple Writer ///, I was given a word processing package designed by Quark called **Word Juggler.**

Word Juggler is incredibly simple to use. You don't even have to memorize any of the command keys. The **Word Juggler** package comes with a template that goes around the number pad on your Apple /// which identifies functions of certain number keys on the keyboard. These include printing, displaying to the screen, returning to the menu and so on. A second strip fits above the top row of keys on the keyboard and identifies special functions such as centering, spacing, margins, page lengths, etc. A reference card is also included.

The manual for **Word Juggler** is well-written, thorough, and understandable (you don't have to be an engineer to figure it out). The manual provides a tutorial to help familiarize yourself with **Word Juggler.** The tutorial takes you through the ABC's of **Word Juggler**, and it really is that simple, learning how the arrow keys move the cursor around the screen (and the text), how to delete characters, how to store a letter on a disk and retrieve it, how to search for a particular word or string of words and replace it/them with another/others, page numbering, and other tasks that are important to you in making the appearance of your letter or document as good as it can be.

One especially helpful feature is the ability to move blocks of text around. **Word Juggler** allows me to copy, move, load, store, and delete blocks of text, all by simply moving the cursor to the beginning and end of the block in question and pressing the space bar.

Word Juggler allows me to insert other Apple /// programs into any **Word Juggler** document, such programs as VisiCalc and PFS. This can be extremely useful when doing letters that contain charts or graphs, or mail lists when you don't want to be limited to characters or lines.

Word Juggler enables me to view my document on the screen in exactly the same form I will see in print simply by hitting the period key on my number pad. One drawback, however, is that you cannot do any editing while in this mode.You must return to the edit mode to do this. After struggling with Apple Writer /// to accomplish this same thing, however, I do not find this inconvenient in **Word Juggler** at all.

Three months ago, I purchased my own Apple /// and was given Apple Writer /// as part of the package. I have never taken it out of the package. As much as the Apple has changed my life in that I can now take on a much larger workload with little or no difficulty, I would not have invested in my own personal Apple had Apple Writer been the only word processing package available to me. It is simply not friendly and is limiting for my purposes.

After 20 years as a secretary, it's hard to break habits, even good ones. I want a software package I can sit down with and start using immediately. In other words, I want my computer to be as easy to use as my typewriter, but faster and more efficient. **Word Juggler** has given me that ability. I can even type at the Apple /// keyboard and have the results print directly on the printer.

After almost nine months with **Word Juggler**, it is still as friendly as ever. I have always been in the habit of bringing work home with me. I can now say it is almost a pleasure. When I finish at the office, I pack up my things and **Word Juggler** and continue at home. After I have completed my work and done my formatting, I bring it to work the next day and print it. I also have Lexicheck, which is a Spelling Checker with a 30,000 word dictionary. This is just one of many accessories from Quark to enhance Word Juggler's capabilities.

I know there are features I haven't mentioned about **Word Juggler.** The truth is, I am still discovering new ways to get more out of this word processing package. **Word Juggler** is easy to use, can be learned quickly, is always friendly, and forgiving. This is so important to a beginner. It is obvious to me that a great deal of care and planning went into this package. Until Apple comes up with something better, thank you, Quark, for **Word Juggler!**

The KOALA TOUCH PAD and the GAMEPORT III:Doing More With Your ///

by Ed Horodko

I have an Apple][at one of my offices, and a /// at the other. You may realize already that this is asking for trouble, since many computer product manufacturers consider the /// to be a replacement ("upgrade") for the][, not a computer you may want to have additionally. That dull idea is not the only problem confronting the][and /// user. Strangely, the ///'s ability to emulate the][has some sinister 'limitations'.

There's a new product that helps. No, you're still stuck with using only 48K of your 256K. Your "free" printer card is still in Slot 6 instead of Slot 1. But at least you'll finally have an Apple][Game I/O port, thanks to MICRO-SCI'S **GAMEPORT III**, a card for one of your precious slots that provides a reasonable facsimile of a real 16-pin][gameport.

You can plug in Apple][paddles and joysticks. You can even connect I/O devices, like remote control systems and software protection keys. Your][games finally work right on your ///, as well as your][graphics-drawing programs.

To run many][programs, you'll need a specially modified Emulation Disk which MICRO-SCI shows you how to make very easily. But when you emulate with that disk, your computer may freeze later when pressing RESET. A small price to pay. This card's presence will not interfere with most native mode operations. You cannot, however, use the **GAMEPORT III** in native mode.

The retail price for the **GAMEPORT III** is $75. Documentation is adequate and use is simple. Without the **GAMEPORT III**, I don't think you can use the very remarkable **KOALA TOUCH PAD** on your ///. I spent hours trying to wire it to the game holes on the back of my ///. No success.

The **TOUCH PAD** offers a new kind of user-interface. It performs just like the mouse, graphics tablet, game paddle, joystick, and game buttons. And though it doesn't work in Native, plugged into the **GAMEPORT III** it's a delight in emulation.

The **KOALA TOUCH PAD** is handsome and well designed. It has two big buttons and a 4.25-inch square surface you press or draw on with with your finger nail or a stylus.

Retailing for $125 (!), the **KOALA TOUCH PAD** comes with a program called MICRO-ILLUSTRATOR, which has a mouse-driven menu. This program lets you easily draw freehand, fill areas with color, and make lines, circles and boxes. Pictures are saved as standard 3.3 picture files (about twelve to a disk).

This program is somewhat limited in scope, but who cares? A more elaborate version called ILLUSTRATOR gives you lots more colors and lets you type in text in various fonts and sizes. It costs $150.

My impression is that software makers are starting to line up outside Koala headquarters so they can write their ware to utilize the Koala Pad as Mouse or, with template, as new dedicated function keys. A good time will be had by all.

New Periodical To Aid Physician Computer Users

A new medical newsletter, Physician Computer Monthly , provides information to the growing number of doctors who use micro and minicomputers in their practice.

This 12-page, independent periodical covers computer applications for practice management, patient care, continuing medical education, and communications. Written in non-technical language, Physician Computer Monthly emphasizes practical uses of computers by physicians.

One year subscription is $95. A sample issue will be provided free to physicians upon receipt of letterhead request; non-physician samples $2 each. Write Physician Computer Monthly , 67 Peachtree Park Dr., Atlanta, GA 30309.

Physician Computer Monthly joins eleven other newsletters published by American Health Consultants, a ten-year-old medical communications firm.

STEMS AND SEEDS:

UPGRADING YOUR 128K APPLE /// TO 256K FOR LESS $$$

If you have a relatively recent vintage 128K Apple /// and feel the need to expand your system memory to 256K, you may be able to do it yourself very easily and for considerably less that the $600 charged by Apple. Last Fall, while installing a clock chip in my ///, I noticed that the memory board on my machine had a row of 16 empty chip sockets. Next to it was an identical row of sockets filled with 16 chips marked HM4864-3. Looking this number up revealed that this was a 1 X 64K, 200 nanosecond dynamic RAM made by Hitachi. I wondered if I couldn't just buy 16 similar chips, plug them into the empty sockets, and maybe double the memory.

Well, it worked! If you shop around, you can buy equivalent RAM chips for as little as $7.50 each and double your memory for as little as $120! This will undoubtedly void your warranty, if you are concerned about such things. You may also have an earlier version of the memory board. There have been 2 or 3 versions apparently, and if you have an earlier one, you are out of luck.

Incidentally, I recommend the use of an IC insertion tool when installing the chips. It makes its easier.

Ken Johnson
Amherst, MA.

-///-

Beginning BASIC Series

Due to the quantity of information we have for this 32 page issue, Stan Guidero's "Beginning BASIC" series will continue in our next Gazette, Volume Two, Number Three.

///

THE AXLON RAMDISK 320

By Stan Guidero

For those of you who know nothing about pseudo disks, I will start with a short explanation. To put it simply, a pseudo disk or ramdisk is used the same way you would use a normal disk drive. The difference is that the unit uses RAM chips in place of a floppy disk. So what's the advantage you ask? Speed is the answer. The ramdisk does not use a mechanical mechanism which tends to slow things down. Because there are no moving parts, it takes a third of the time required by a floppy or hard disk. A program that takes almost 5 seconds to load may only take 1 second to load and run with a ramdisk. You can see that this would come in handy when handling data or a program with several moduals like an accounting system.

If you work with Pascal or use a Data Management System that requires the exchange of several different disks, then the **Axlon RAMDISK 320** is for you. It isn't intended to be used as a mass storage device (although it could be used as such,) but as a high-speed Data handler. The RAMDISK is handled by SOS in the same way it handles a hard disk or a Profile. The RAMDISK can be formatted either as two 143K disk drives or one 320K disk drive. Thus, two SOS.DRIVERS are included in the RAMDISK package. I found little use for the two drive set- up on the Apple /// and mainly used the single drive drive.

The **Axlon RAMDISK 320** comes in a separate case that looks like a disk drive][for an Apple][and sits next to your Apple ///. It has its' own power supply and a built-in three hour rechargeable battery to prevent memory loss. Installation is simple; an interface card is plugged into one of the slots in the ///, preferrably slot #4. Attaching the ribbon cable was the only difficult thing I had to do. If you didn't watch what you were doing you could easly plug it into the interface card upside down or even one or two pins off. Manufacturers should realize that people not technically oriented are the ones who usually buy Apple ///s and take that into account when designing computer peripherals.

After checking the manual to make sure I was doing it right, I plugged the RAMDISK 320 into the ///. After connecting the power cords (both the Apple and RAMDISK,) into the wall, I turned on the power. Nothing popped or sizzled so I figured I was safe. The next thing to do was to add the proper driver by using the SOS Utility Program. Two drivers are available on the accompaning disk. I first decided to use the two driver emulator.

First, I tried the RAMDISK 320 with the Pascal System. I tried the two-drive method but found the system was spending too much time looking through all the various drives every time I wanted to change system moduals. I switched to the single large drive. I then placed most of the Pascal System on the RAMDISK and found it took much less time searching but it still looked at .D1 and .D2 every time I went from the Filer to the command line or from the Editor to the command line.

After much experimentation, I finally came up with a method that sort of worked, not perfectly, but it worked. I placed the system files SOS.KERNAL, SOS.DRIVE, SOS.INTERP, SYSTEM.PASCAL, and SYSTEM.MISINFO on my boot disk. Next I placed all the other system and library files on the RAMDISK. Incidentally, the Pascal device number for the RAMDISK is #9 which is the same as a hard disk. I used the SOS Utility to do the transfering. I then named the Volume /RAMDISK. After booting up the system, I then pressed "F" for the filer making sure there was no disk in .D2 so that the only place it would find the filer would be the RAMDISK.

I then did the same for the EDITOR. The main problem came when I tried to compile a program. The system work file is kept on the onboard (.D1) drive making compiling as slow as ever. I avoided this problem by writing my work files to the RAMDISK. Thus, the Apple spent far less time accessing the work file. The only time the onboard drive is accessed is when going back to the main command line.

It was a pleasure to simply strike a key to go to the EDITOR and wham: in less than two seconds I was there! Fixing programming errors was no longer a lengthy chore and through all this the RAMDISK performed flawlessly.

The RAMDISK will work with the Softcard /// under CP/M. You use the Softcard /// utility disk to change the DRIVER.SOS on the

boot CP/M disk. From there on it is handled like a hard disk. I'm also told it will work in the emulation mode, but who cares?

Not all is perfect. There are a few things I don't like about the RAMDISK, foremost being the price. At $1350.00 it's no bargain. If you're going to pay that kind of money, you should have a definite need for it. Second, the case that it comes in is identical to the Apple][disk drive and not the Disk /// case. Of course I've already mentioned the problem with hooking-up the ribbon cable to the interface card.

One other thing bothers me. Even though during the month I had the unit for testing it showed no memory failure problems, and not even so much as a clitch, I can't help thinking about all those little RAM chips jammed together in that non-ventilated box and how hot they must get. The outside of the box is much hotter than either the Apple /// or my monitor. Some ventilation would make me, and probably the chips, breath easier.

If you need fast data access for a specific job, such as Pascal programming or multi-modual program package usage, then the **Axlon RAMDISK 320** is for you. But you will have to decide if having the extra speed is worth the price.

-///-

Release of BACKUP /// For PROFILE Owners

Late last year, as many of your readers may know, Apple Computer, Inc., announced the availability of BACKUP ///, a utility program for backing-up the PROFILE. Apple indicated that it would soon start shipping BACKUP ///'s with all new PROFILES and that it would send a copy of BACKUP /// to all current PROFILE owners of record.

BACKUP /// is a fine utility program. It allows back-up of volumes, subdirectories or files in three different ways; (1) by entire volume, subdirectory or file, (2) by files modified since the last back-up or (3) by files modified after a certain date and time.

If current PROFILE owners cannot get their copy of BACKUP /// through their dealers, they may obtain a copy by sending a written request, together with their PROFILE serial number to:

Peripheral Systems Product Support
Apple Computer, Inc.
Mail Stop 20-C
2730 Orchard Park Way
San Jose, CA., 95134

Apple ///'rs Beware: Time Clock Mail Fraud?

Dear Mr. Norris,

I am an Apple /// owner, and am writing this letter to inform your Apple /// user organization of a company which is, at my best perception, engaging in mail fraud dealing with a proposed Apple /// product.

The product is a kit for modifying a personal computer (namely the Apple ///) for a real-time clock function. To the best of my knowledge, this kit consists of an integrated circuit (MM58167A clock chip), a battery holder, some batteries, and instructions. The advertised price for this product was $60.00.

I had sent out my order during the week of May 16, 1983 and had waited four weeks until I became uncertain of the product's delivery. I found from my monthly checking statement that System Fabricators had cashed my check on June 3, 1983. I then called up their advertised phone number, (213)372-6273 and found out that their phone is disconnected. I sent the enclosed letter out via registered mail, but I have not received confirmation that the letter was received from the Post Office.

I am enclosing a copy of the letter which I have sent to System Fabricators, located in Hermosa Beach, California. I am having difficulty in obtaining a product from them which I have requested through the mail and for which I have paid $60.00 in the form of a personal check. The product was advertised in Apple Computer Users magazines like Apple Orchard, Vol. 4 No. 1, and in Softalk magazine.

I would like to get my money back from System Fabricators, for I have no confidence in their integrity as a company, and therefore, their product. If your organization can help me towards this goal, I would be grateful. At the very least, your organization should be notified that this suspicious vendor exists and may prey on other unsuspecting customers.

Sincerely,

Chester M. Lee
Framingham, MA.

Installing APPLE WRITER /// Utilities On PROFILE With CATALYST

Dear Sirs;

In Volume 1, No. 6, of the Open Apple Gazette , Allan Bloom wrote of his difficulties in installing APPLE WRITER /// Utilities On PROFILE with CATALYST. I have found that the installation can be done by following the procedure listed below:

1. Transfer the following files:

FROM AW3UTIL

SYSTEM.LIBRARY
MLMTRAN.CODE
SOS.INTERP
SYSTEM.STARTUP

TO .PROFILE/CATALYST/AWUT

STARTUP.LIB
MLMTRAN.CODE
AWUT (or your preferred name)
STARTUP.CODE

DO NOT TRANSFER SYSTEM.PASCAL

2. Add AWUTIL to the CATALYST menu and enter the program options as follows:

1). Display Name: AWUTIL (or your preferred name)

2). Interpreter Path: CATALYST/AWUT/AWUT (or however you name it)

3). Character set path: DEFAULT

4). Screen: Normal

5). Keyboard Path: DEFAULT

6). Initial Prefix: .PROFILE/CATALYST/AWUT (see above)

7). Maximum files allowed open: 3

8). Programs path:
CATALYST/AWUT/STARTUP.CODE

TO RUN AW3OUTILITIES:

1. Call AWUTIL from CATALYST Menu. You will get the message "Put Pascal System Disk in built-in drive, press Return".

2. Put AW3OUTIL Disk in built-in drive (.d1) and press "Return".

3. Run Program.

Transfers will operate between built-in drive (.d1) for Apple II disk and .d2, etc. for Apple /// disk.

CATALYST will still be operating so you can use double Apple-Escape to return to CATALYST menu when finished with AW3OUTIL.

This way you need not boot the AW3OUTIL disk. You will be able to return to the CATALYST menu at any time. The only discomfort is in the exchanging of the CATALYST disk and the AW3OUTIL disk.

In effect, you are using a two-stage boot, as in non-CATALYST times,and when the System Pascal is taken from .d1 the built-in drive is properly identified for these programs. I have been using this method I developed after having the same disaster strike me as struck Mr. Bloom. No disaster has occurred using this method.

If someone can tell us how to use CATALYST to get Apple II Emulation I think that that would eliminate the only non-CATALYST booting I am still doing.

Hope this helps,

Harold A. Drob
Academic Affairs Office
S-115
University Of California
San Francisco, CA., 94143

More CATALYST Notes:

Dear Don,

Thanks, very much, for your July 20 note with Harold Drob's July 13 attached. (See this issue's Reader Reaction Section). As I said in my reply to him, it's awfully nice to find helpful, caring people. I also sent him a copy of my own solution to the "Apple Writer Utilities" problem, a simpler and more general one that I sent you some months back.

I'm delighted with the SOS 1.3 Revision Utility. I'd swiped a copy of the SOS Kernel file from a colleague's "Discourse" spooler some months back, but the new drivers were worth the freight. I also appreciate the note in Vol. 1, No. 6, about the drivers that shouldn't be installed under CATALYST.

The "AppleWriter /// Bug??" (See our last issue, Vol. 1, No. 6) question from Joe Pase and Charles Bryant has a familiar ring to it. If they are running under CATALYST, the problem and solution are simple. When AppleWriter is booted up, the "Press RETURN" message causes a read to .D1 to find the SYS.PRT and SYS.TAB files. If either is not found, a SOS "File Not Found" message pops up, but it doesn't prevent you from continuing. AppleWriter does not initialize values for the PRT and TAB data. Tabbing is particularly spectacular with garbage in the TAB data area. You go to the end of the 64K workspace, and the program hangs-up. A cold re-boot is necessary. Sound familiar?

This particular problem is yet another example of poor programming practice. If a program absolutely needs a file to function, and if it doesn't find one, it should have the courtesy to tell you so and to not proceed until you correct the problem. Unless/until that bug is fixed, you have to look out for yourself. I keep a Profile subdirectory containing my default SYS.PRT and SYS.TAB files (the HELP stuff from the AppleWriter master should be there too, if you're into such) and I just copy its contents to every new data disk.

Yours truly,

Al Bloom
2303 San Marcos St.
Blacksburg, VA., 24060

STEMS AND SEEDS:

Counting the Total Number of Words In A File Using APPLE WRITER ///

Here's a hint for APPLE WRITER /// users: If you want to count the total number of words in a file (very necessary to writers), you can load the program "COUNTER" from the master disk and change the line beginning "LOOP...." to this:

```
LOOP     F<$B<$B<
```

After saving the above as "WORD.COUNTER", you can run it in the usual way for WPL programs ([P] DO WORD.COUNTER), but when prompted for "Enter word:", you should type in this:

(spacebar)?

To be more exact, press the spacebar, then the question mark with no spaces between. After pressing RETURN, the program will count all the words in the document. Pressing CONTROL-5 (5 on the numeric keypad) will darken the screen and speed up the count. When the count is finished, the screen will come back on automatically.

Sharon Webb
Blairsville, GA.

[**Ed. Note:** Adding "PPR [V][N][V]" to your program will execute the "CONTROL-5" function automatically without having to perform this extra step.]

-///-

STEMS AND SEEDS:

EPSON Printer Interface Cards

Alpine Computing, Inc. of 84 West 100 North, Logan, Utah 84321, is marketing Apple /// driver routines which allow you to drive the EPSON parallel printers through the EPSON interface card, thus saving the cost of buying Apple's Universal Parallel Interface Card. The routines sell for $20.00 and have worked well for me.

Barry R. Weller
Assoc. Prof. of Economics
Penn State University

-///-

Exploring Business Basic - Part Seven

By Taylor Pohlman
Reprinted from Softalk Magazine

Last month we explored some of Business Basic's unique formatted output and arithmetic capabilities. There is a good more to say on those topics, but such exposition will be left to some future month. This month we'll undertake a journey through some of the thickest jungles found in the Apple III: the infamous .GRAFIX driver and its faithful Indian companion, BGRAF.INV (the preceeding collection of mixed metaphors was just a sample of what some enterprising explorers have encountered on their own trips).

The new Business Basic manual (which everyone who purchased Basic should have received by now), contains a sixty-page section in volume two that describes the programming possible with the BGRAF.INV invokable module. In addition, the "Standard Device Drivers Manual" contains a section on .GRAFIX. Rather than repeat all of that material, this column will briefly describe the functions of BGRAF.INV and then take up a subject that's not mentioned at all: how to draw a circle.

Drawing a circle may sound easy, but given that BGRAF allows only dots and lines, and given that none of the graphics modes have equal horizontal and vertical resolution, and given that monitors distort images because of "aspect ratio" differences, drawing a circle of arbitrary radius with an arbitrary center that actually looks like a circle and doesn't take forever to finish is non-trivial. Non-trivial is a favorite word of mathematicians and engineers, principally because it allows them to assert that a task is difficult without requiring that they figure out how difficult.

Well, enough cheap shots at mathematicians. We will discover later that some handy mathematical principles will serve us well in our quest for the perfect circle.

The BGRAF Invokable Module. As has been discussed before, Business Basic is almost infinitely extensible by the use of "invokable" assembly language routines. These routines can be loaded into memory only when needed, and have the effect of adding extra commands to the language. Furthermore, as many invokables as will fit into memory can be used at once, with Apple III's SOS operating system responsible for making sure that there are no conflicts. BGRAF.INV is one of the most useful of the invokable modules. It is supplied on the Business Basic program disk, and is loaded into memory with the command:

```
INVOKE "bgraf.inv"
```

Those of you who have Apple II's, or who have used Applesoft in emulation mode on the Apple III, know that there are several commands in that language to manipulate color graphics. Among these are "gr" and "hgr," "color" and "hcolor," "plot" and "hplot" that permit initializing graphics modes, changing colors, and plotting points and lines. In addition, Applesoft has special commands that permit the manipulation of shapes based on special tables that describe the bit patterns of the images.

The BGRAF invokable has commands for all these capabilities, and a great deal more besides. Unlike Applesoft, which has a fixed high-resolution page for drawing, the Apple III graphics modes permit plotting points within a range of -32768 to 32767. The concept of a "viewport" (like the window in text mode) is what defines which dots actually get plotted on the screen. Only the dots within the current viewport are actually plotted, and the viewport is limited to the maximum resolution of whatever graphics mode is selected.

We'll see shortly how handy this is, because it permits plotting generally without regard to whether the physical screen limits are exceeded. Exceeding the valid range in an Applesoft program causes an error. In addition, setting the viewport to an area smaller than the physical screen permits us to draw without worrying about overwriting other areas outside the viewport. To keep track of where the plotting operations are to take place, an invisible cursor is maintained, to which all draw and print operations relate.

There are two capabilities of the Apple III graphics driver which are not well understood but can be extremely powerful. These come under the heading of the color table and transfer option. Used properly, they can save an incredible amount of programmer effort.

The color table allows you to set the

priority of a given color. Imagine that you want to draw some blue lines on a screen that contains some yellow squares. Suppose, furthermore, that you don't want to cross the yellow squares (in effect, you want to draw the line behind the squares). If the color table is set up properly, the graphics driver will automatically change any blue dots you plot to yellow if you try to plot them over a yellow dot. In any other system, your program would have to check the color of each dot before plotting, thus grinding everything to a virtual halt. Imagine what this capability could mean if you wanted to animate shapes of various colors over a background.

One additional capability in Apple III graphics is really convenient. At any time you can print text directly to the .GRAFIX driver and it will be written at the current dot position. Since you can also change the definition of the character set with the "newfont" procedure, hi-res animation tricks of the sort found in the Apple II DOS "Toolkit" package are essentially built-in!

To give you an idea of the functions of the BGRAF module, the command table lists the commands available, along with a brief description of each. Remember that to use these in Basic, the module must first be "invoked," and the word "perform" prefixed to each command. Quite a collection of goodies, right?

As was said earlier, there is really too much here for one article; indeed, a whole book could be written about the Apple III graphics. Rather than tackle that task, let's start with something seemingly simple. As you noted from reading the list of functions above, the major component that's missing is anything to do with curves. Rather than throwing you a curve (groan!), we'll try drawing some.

Getting Around in Business Basic. To start, some quick math is required. You can think of the Apple III screen as a coordinate system, with "x" and "y" locations depending on the mode. In all cases, "y" (vertical) values are displayable between 0 (bottom) and 191 (top). X (horizontal) values range between 0 and 139 (lowest resolution) up to 0 to 559 (highest). Circles are nothing more than sets of points with a common attribute; namely, they are of equal distance from a single point called (surprisingly enough) the center. There are formulas for determining the points that lie on a circle, generally derived from the formula below:

$$x^2 + y^2 = r^2$$

This formula works for circles starting at a center of 0,0, but since we want to draw circles anywhere, and since the general form of the circle equation is more difficult to solve, we'll rely on another fact about circles. The trigonometric functions sine and cosine define "x" and "y" values for the unit circle, and it is possible to obtain values for any circle by multiplying these values by the radius and adding the center coordinates; that is, to find a point on a circle of radius 30 at an angle of 30 degrees from horizontal, when the center is at x=70 and y=96, these formulas can be used:

```
x = COS(30 degrees)*30+70
y = SIN(30 degrees)*30+96
```

These simple formulas suggest that we might be ready to write a program.

Program 1 is a relatively straightforward program, except that the trig functions ("sin," "cos") work in radians, of which there are 2 x pi in a full circle. That value is approximately 6.2832, which is further approximated in line 100. One other thing of note: Since this graphics mode is not "square," some adjustment must be made for the fact that there are more points proportionately in one axis than in another.

For simplicity, we have scaled the x-axis value (since that is the only one that varies in the different modes) by multiplying by the constant 140/192, the ratio of horizontal to vertical dots. This is done in line 110. The other factor in lines 110 and 120 is the constant 30, which represents the radius. Note also that line 170 cleans up the graphics memory and closes the driver. This is **very** important. If you don't release the memory, it will stay around, unusable by Basic. Also, doing the "invoke" in line 180 removes the BGRAF module from memory. If you have other invokables normally resident, you should delete this statement.

Run this program several times, with different values for the step. You will notice that in addition to being pretty slow, it takes a step size of about .1 to draw a good circle. You'll probably also notice that this program can't draw a very good circle. Depending on the aspect ratio of your monitor, the circle will probably

Command Table

Initgrafix	Initializes the viewport, cursor position, color table and transfer options
Grafixmode	Sets the current graphics mode. The four modes are: 0 - 280 x 192 black and white 1 - 280 x 192 color (sixteen colors with limitations) 2 - 560 x 192 black and white 3 - 140 x 192 color (sixteen colors with no limitations)
Grafixon	Displays the current graphics screen
Viewport	Sets the boundaries for graphics operations
Pencolor	Sets the color of the pen for draws, plots or characters
Fillcolor	Sets the background color for filling and erasing
Fillport	Fills the current viewport with the fill color
Moveto	Moves the cursor to a specified point
Moverel	Moves the cursor relative to the current point
Dotat	Plots a point at a specified point
Dotrel	Plots a point relative to the current point
Lineto	Draws a line from the current point to a specified point
Linerel	Draws a line to a point relative to the current point
XYColor	A function that returns the color of a specified point
Xloc,Yloc	Functions that return the current position of the cursor
Newfont	Defines a new character font for printing text on the screen
Sysfont	Restores the default system font
Gsave	Saves a graphics screen to disk as a PIC file
Gload	Loads a PIC file from disk to the current graphics screen
Release	Gives graphics memory back to Basic

Program 1

```
10   OPEN#1,".grafix"                          Open the graphics driver
20   INVOKE"bgraf.inv"                         Load BGRAF into memory
30   PERFORM initgrafix                        Initialize the graphics screen
40   PERFORM grafixmode(%3,%1)                 Set mode 3 (sixteen-color)
45   INPUT"step value: ";stepval               Ask for an increment for plotting
50   PERFORM grafixon                          Turn on the graphics display
60   PERFORM pencolor(%13)                     Set the color for drawing (yellow)
70   PERFORM fillcolor(%3)                     Set the background color (purple)
80   PERFORM fillport                          Fill the viewport with purple
100  FOR i=0 TO 6.28 STEP stepval              Step around the circle (2pi=6.28)
110  x=COS(i)*30*(140/192)+70                  Calculate x (center=70, radius=30)
120  y=SIN(i)*30+96                            Calculate y (center at y=96)
130  PERFORM dotat(%x,%y)                      Plot the resulting dot location
140  NEXT i
150  INPUT a$                                  Pause when finished
160  TEXT                                      Switch to text mode
170  PERFORM RELEASE:PERFORM RELEASE:CLOSE
180  INVOKE                                    Clean up the graphics memory
190  END
```

look like a flattened circle, that is, an elipse. This is so because all monitors differ in the relationship between horizontal and vertical resolution and size. We will see a little later that this is an easy problem to correct.

As was just mentioned, this routine suffers from being very slow. The main problem stems from the fact that it takes a large number of dots to create a circle, and that number of dots translates into a large number of steps to draw a circle. In the example above, it took approximately sixty-four dots to draw a filled-in circle of radius 30 units. Had we tried to draw a larger circle, or had we used a higher resolution mode, the problem would have been even worse. The solution to this problem lies in understanding the real nature of the task at hand.

Mathematics and mathematical physics is sometimes called the realm of the perfect. Energy truly is exactly related to mass times the speed of light squared (Einstein's famous formula). But in the world of measured events, nothing is ever exact and perfect.

The same is true of circles. Geometry allows us to dream of perfect circles, but the realities of trying to draw even one (especially freehand!) are such that we are willing to settle for reasonably good representations as long as they are not too lumpy. In fact, the resolution of any graphics screen--no matter how good--is a far cry from the perfection of a real circle. Therefore, when we set out to draw a circle on the screen, we should first ask, "How good a circle do we want?"

A quick lesson from geometry will help with the answer. As you may know, a circle can be approximated as a polygon (a figure with many sides). The more sides the polygon has, the more it looks like a circle. Since the resolution of the graphics screen is limited, at some reasonable point a polygon will be indistinguishable from a circle.

The advantage of this approach is that the Apple III has graphics commands available for drawing lines. Since a polygon has sides that are all straight lines, we can use the line drawing commands to represent a circle, finding the number of sides in each resolution which make reasonable looking circles. The number of sides necessary to make a good circle will also vary according to the radius of the circle, since large circles will be more likely to show the lines as straight segments.

Program 2 will let you play with the number of sides necessary to make good circles and to experiment to determine the aspect ratio that is correct for your monitor. Modify this program to try other graphics modes and see how the different resolutions affect the results.

```
10   OPEN#1,".grafix"
20   INVOKE".d1/bgraf.inv"
30   PERFORM initgrafix
40   PERFORM grafixmode(%3,%1)
45   INPUT"step value: ";stepval
46   IF stepval<=0 THEN 170
48   INPUT"aspect ratio: ";aspect
49   aspect=1/aspect
50   PERFORM grafixon
60   PERFORM pencolor(%13)
70   PERFORM fillcolor(%3)
80   PERFORM fillport
85   scale=140/192
90   PERFORM moveto(%30*scale*aspect+
     70,%96)
100  FOR i=stepval TO 6.28 STEP stepval
110  x=COS(i)*30*scale*aspect+70
120  y=SIN(i)*30+96
130  PERFORM lineto(%x,%y)
140  NEXT i
150  INPUT a$
160  TEXT
165  GOTO 45
170  PERFORM RELEASE:PERFORM RELEASE:
     CLOSE
180  INVOKE
190  END
```

Several new things are done in this program 2. Notice that we invert the aspect ratio because we are adjusting the x-axis only. Further, we have named the ratio between the "x" and "y" resolution "scale" for use in the repetitive calculations. Next, because we are drawing lines this time, the program uses the "moveto" procedure to move the graphics cursor to the initial point on the circle (in this case, the horizontal point to the right of the center (origin) of the circle).

Once a starting point is established, subsequent "lineto" commands will draw the circle as a series of line segments. Try experimenting with widely varying numbers of steps, from 1 to .02 as an example. You will find that at some point the circle looks the same, no matter how many line segments make it up. By choosing the fewest steps that still produce a decent circle,

you can speed up the drawing process considerably. Don't forget to experiment with values for aspect ratio as well. For the Monitor III, a value of 1.3 usually works pretty well. Try several values until you are happy with the results.

The next technique for speeding up this routine is even more interesting. Notice that we keep calculating the sine and cosine of each angle, no matter how many times we run the program. Furthermore, it should be apparent that it's possible to draw a circle of any reasonable radius by just varying the multiplication factor. Further, it's obvious that steps of less than .05 for any reasonable radius do not produce better circles. All that suggests program 3, an enhanced version of the original program:

```
10   OPEN#1,".grafix"
20   INVOKE".dl/bgraf.inv"
25   DIM xcos(63),ysin(63)
26   FOR i=0 TO 63:xcos(i)=COS(i/10):
     ysin(i)=SIN(i/10):NEXT i
30   PERFORM initgrafix
40   PERFORM grafixmode(%3,%1)
46   INPUT"aspect ratio: ";aspect
47   aspect=1/aspect
48   INPUT"radius: ";r
49   IF r<=0 THEN 170
50   PERFORM grafixon
60   PERFORM pencolor(%13)
70   PERFORM fillcolor(%3)
80   PERFORM fillport
85   scale=140/192
90   PERFORM moveto(%r*scale*aspect
     +70,%96)
100  FOR i=1 TO 63
110  x=xcos(i)*r*scale*aspect+70
120  y=ysin(i)*r+96
130  PERFORM lineto(%x,%y)
140  NEXT i
150  INPUT a$
160  TEXT
165  GOTO 48
170  PERFORM RELEASE:PERFORM RELEASE:
     CLOSE
180  INVOKE
190  END
```

Notice that this time two arrays have been set up, both with sixty-four values each. Rather than recalculate the "sin" and "cos" functions, the program does them once at the beginning and stores them for use later in lines 110 and 120. If you intend to do a great deal of this kind of work, or if you want to expand the number of steps significantly, you may want to create another program that calculates the values and writes them to a data file. Then your circle program could simply read the values in at the beginning.

Notice, too, that these routines are essentially identical to the previous ones, except that this time you may experiment with circles of different radii. After experimenting with this routine, you should see that a general purpose routine can be written that will satisfy all circumstances. It's most practical to express this as a subroutine, with the variables being the circle diameter, the mode, the scale factor and the center coordinates. The result might look like this program 4.

```
10   OPEN#1,".grafix"
20   INVOKE".dl/bgraf.inv"
25   DIM xcos(126),ysin(126),xdot(3)
26   FOR i=0 TO 126:xcos(i)=COS(i/20):
     ysin(i)=SIN(i/20):NEXT i
27   xdot(0)=280:xdot(1)=280:xdot(2)=
     560:xdot(3)=140
30   PERFORM initgrafix
35   INPUT"Mode: ";mode
37   IF mode<=0 THEN 180
40   PERFORM grafixmode(%mode,%1)
50   INPUT"pencolor,fillcolor: ";pen,
     fill
60   PERFORM pencolor(%pen)
70   PERFORM fillcolor(%fill)
75   INPUT"clear screen? ";a$
80   a$=MID$(a$,1,1):IF a$="y" OR a$="Y"
     THEN PERFORM fillport
81   INPUT"radius: ";r
82   aratio=1.3
83   xcen=xdot(mode)/2:ycen=96
85   PERFORM grafixon
87   scalefac=(1/aratio)*(xdot(mode)
     /192)
90   GOSUB 900
150  PERFORM moveto(%0,%8)
160  PRINT#1;"Press RETURN:";
165  INPUT"";a$
170  TEXT
175  GOTO 75
180  PERFORM RELEASE:PERFORM RELEASE:
     CLOSE
190  INVOKE
200  END
900  xscale=r*scalefac
905  xcen=xcen+.5:ycen=ycen+.5
907  density=(mode=2)+2*(mode<2)
     +3*(mode=3)
910  firstx=xcos(0)*xscale+xcen
915  PERFORM moveto(%firstx,%ycen)
920  stepamt=INT(20*(5-density)
     /r)+density
930  IF stepamt>6 THEN stepamt=6
940  FOR i=1 TO 126 STEP stepamt
950  PERFORM lineto(%(xcos(i)*xscale
```

```
      +xcen),%(ysin(i)*r+ycen))
960   NEXT i
970   PERFORM lineto(%first,%ycen)
980   RETURN
```

Program 4 is a considerably enhanced version. Notice that we have doubled the number of points that can be used, as well as introducing the capacity to change modes and colors. Since we now have different possibilities for the mode, we introduce the xdot array, which contains the horizontal dot density required to figure the center and the scale factor. The actual drawing routine is now a subroutine at line 900, in such a form that you could incorporate it into other programs.

To speed up the subroutine for the various graphics modes, the concept of "density" is introduced. This factor varies between one and three depending on whether the horizontal resolution is 140, 280 or 560 (modes 3, 0 and 1, and 2 respectively). Note the use of the logical statements in line 907 to replace a lot of "if" and assignment statements. Be sure you work through that program line in your mind to assure yourself that the assignments work as intended.

The other thing to note is that the density factor is used in line 920 to calculate a reasonable step value for drawing the circle. You might want to substitute a few values to see just how this works. Line 930 makes sure the program uses a reasonable number of steps even if the circle is extremely small.

Notice also that in line 905 the value .5 is added to the center coordinates. This has the effect of rounding the values when they are passed to the "lineto" procedure in line 950, ensuring more accurate plotting. Another new feature is that in line 970 an additional "lineto" is added to draw a line back to the original point. This ensures that if the step value is such that the circle is not fully completed, the last point drawn will be connected to the beginning point.

Line 160 presents another new concept. By simply printing to the .GRAFIX driver file, you may write text on any graphics screen. Furthermore, the text can begin on any dot boundary anywhere on the screen. That requires some pretty tricky software on the Apple II, but is a built-in feature of the Apple III graphics modes.

Many more enhancements could be added to this program, but instead of going on and on, here's an example of how the program and routines could be modified to draw circle segments (arcs) and pie slices, using essentially the same techniques. The new program (program 5) looks like this:

```
3     REM arc draw subroutine
10    GOSUB 1000:REM initialize
20    PRINT"Arc drawer program"
35    INPUT"Graphics mode: ";mode$
36    IF mode$="" THEN 180
37    mode=CONV(mode$)
40    PERFORM grafixmode(%mode,%1)
50    INPUT"pencolor,fillcolor: ";pen,
      fill
52    draw.radius=0
55    INPUT"draw the radii? ";a$
56    a$=MID$(a$,1,1):IF a$="y" OR a$="Y"
      THEN draw.radius=1
60    PERFORM pencolor(%pen)
70    PERFORM fillcolor(%fill)
75    INPUT"clear screen? ";a$
80    a$=MID$(a$,1,1):IF a$="y" OR a$="Y"
      THEN PERFORM fillport
82    horiz=xdot(mode)/192
85    scalefac=(1/aratio)*horiz
87    PERFORM grafixon
88    FOR loop=1 TO 25
90    r=INT(50*RND(1)+30)
91    xcen=INT(192*horiz*RND(1))
92    ycen=INT(192*RND(1))
93    start.rad=3.14*RND(1):end.rad=
      start.rad+3*RND(1)
95    GOSUB 1100
100   NEXT loop
150   PERFORM moveto(%0,%8)
160   PRINT#1;"Press RETURN:";
165   INPUT"";a$
170   TEXT:GOTO 35
180   PERFORM release:PERFORM release:
      PERFORM release
190   CLOSE:INVOKE
200   END
1000  OPEN#1,".grafix"
1010  INVOKE".d1/bgraf.inv"
1020  DIM xcos(126),ysin(126),xdot(3)
1030  FOR i=0 TO 126:xcos(i)=COS(i/20)
      :ysin(i)=SIN(i/20):NEXT i
1040  xdot(0)=280:xdot(1)=280:xdot(2)=
      560:xdot(3)=140
1050  aratio=1.3
1060  PERFORM initgrafix
1070  RETURN
1094  REM r=radius, scalefac=aspect ratio
      * relative density
1095  REM xcen= x coordinate of center
1096  REM ycen= y coordinate of center
1097  REM start.rad= starting point of
      arc in radians
```

```
1098  REM end.rad=ending point of arc in
      radians
1099  REM draw.radius=1 means draw the
      radius lines to the endpoints
1100  xscale=r*scalefac
1105  xcen=xcen+.5:ycen=ycen+.5
1110  density=(mode=2)+2*(mode<2)
      +3*(mode=3)
1115  IF draw.radius THEN PERFORM moveto
      (%xcen,%ycen):PERFORM
      lineto(%(COS(start.rad)*
      xscale+xcen),%(SIN(start.rad)*r+
      ycen)):ELSE:PERFORM moveto(%(COS
      (start.rad)*xscale+xcen),%(SIN
      (start.rad)*r+ycen))
1120  stepamt=INT(20*(5-density)/r)+
      density
1130  IF stepamt>6 THEN stepamt=6
1140  FOR i=INT(start.rad*20+.5) TO end.
      rad*20 STEP stepamt
1150  PERFORM lineto(%(xcos(i)*xscale+x
      cen), %(ysin(i)*r+ycen))
1160  NEXT i
1170  PERFORM lineto(%(COS(end.rad)*
      xscale+xcen),%(SIN(end.rad)*r+y
      cen))
1175  IF draw.radius THEN PERFORM
      lineto(%xcen,%ycen)
1180  RETURN
```

Program 5 is set up to use lines 88 through 100 to create random centers, radii and arc lengths (in radians) and to use the subroutine at line 1100 to draw the resulting arcs. This routine and the one above for drawing circles should equip you to do most of the interesting tasks in graphics. It's likely that these routines will also give you ideas on how to approach other specific projects you might wish to tackle.

Normally, when you run the arc program you'll get some arcs that are partially off the screen. Notice that the .GRAFIX driver handles such a situation perfectly because it treats its graphics area as a space of points from -32768 to 32767, with the screen as a window into the total space. This eliminates immeasurable amounts of bounds checking within programs, a process that usually ends up slowing down the drawing. In addition, as was mentioned already, the graphics window can be set to anywhere on the screen, with any values outside the window automatically clipped.

There are a thousand more topics to be covered in exploring the graphics capabilities of the Apple III. Next month we will tackle a few biggies, "area fill" (especially for the circles and arcs we have been drawing) and the whole area of user-definable character sets. With luck, we'll get to some animation examples. Until then, dig into your device driver manual documentation on .GRAFIX and the writeup on BGRAF.INV in the Business Basic manual. There's a whole world inside this system!

STEMS AND SEEDS:

"INCOMPATIBLE INTERPRETER" EXPLANATION

We have received many inquiries from members who have purchased Public Domain disks through the group about the above error message appearing on their screens. The "INCOMPATIBLE INTERPRETER" error message is caused by using the SOS 1.3 kernal with older versions of the Pascal Interpreter, such as used on the DOS to SOS Text File Converter, and the Business BASIC interpreter version 1.0, which is on the File Cabinet /// diskette. Our master diskettes for these programs were updated to SOS 1.3, and once we discovered the error of our ways, we downgraded our master diskettes to SOS 1.0.

File Cabinet /// can be upgraded to Business BASIC 1.1 by deleting the SOS.INTERPRETER from your FILE CABINET /// boot diskette. Replace it with the SOS.INTERPRETER from your Business BASIC version 1.1 diskette using the System Utilities Program. You will then be able to upgrade your File Cabinet /// to SOS 1.3.

The DOS to SOS Text File Converter will probably require changing SOS.INTERPRETER and SYSTEM.PASCAL in order to be compatible with SOS 1.3. Try deleting both of these from your DOS to SOS Text File Converter diskette and replacing them with the ones on your System Utilities diskette.

We apologize for any inconvenience this may have caused any of our members.

-///-

open apple gazette

1850 Union Street #494
San Francisco CA 94123

Business Apple Group, Inc.

CLUB INFORMATION

MEETINGS

Meetings are held at 7:30 PM on the third Wednesday of each month. Call Don Norris for current meeting location.

MEMBERSHIP

Annual membership dues are $30 from the date application received. Your check payable to the Business Apple Group, Inc. may be mailed to the address below.

OPEN APPLE GAZETTE POLICY

All manuscripts, photographs, and other materials are submitted free and released for publication. They become the property of the Business Apple Group, Inc. and the Open Apple Gazette. Authors should clearly mark all material submitted for publication so that credit may be given.
The publishers/editors do not necessarily agree with, nor stand responsible for, opinions expressed or implied by other than themselves in this publication.
The Business Apple Group, Inc. is a non-profit organization comprised of, and supported by, Apple /// owners and users. We endeavor to aid other Apple /// users through this educational publication - "OPEN APPLE GAZETTE." Address all inquiries to: Business Apple Group, Inc., 1850 Union Street #494, San Francisco, CA 94123.

REPRINT POLICY

ARTICLE SUBMISSION POLICY

The Open Apple Gazette welcomes any and all articles dealing with the Apple /// Computer and its associated hardware and software. Articles should be submitted on diskette as an ACSII text file, such as those produced by either Word Juggler, Apple Writer /// or the Pascal Editor. Typewritten double spaced articles are also acceptable.

OFFICERS

PRESIDENT	Don Norris	(415) 921-3774
VICE PRESIDENT	Kent Hockabout	(415)521-5414
SECRETARY	Charles Coles	(415) 386-8623
CONSULTANTS	Ransom Fields	
	Ken Silverman	
ASSISTANT EDITOR	Stuart Henderson	

Back Issues

Volume 1, Numbers 2-6	$4.00 each
Volume 2, Number 1	$4.00 each

Mail Requests for Back Issues to:

Open Apple Gazette
1850 Union Street, #494
San Francisco, CA 94123

-///-