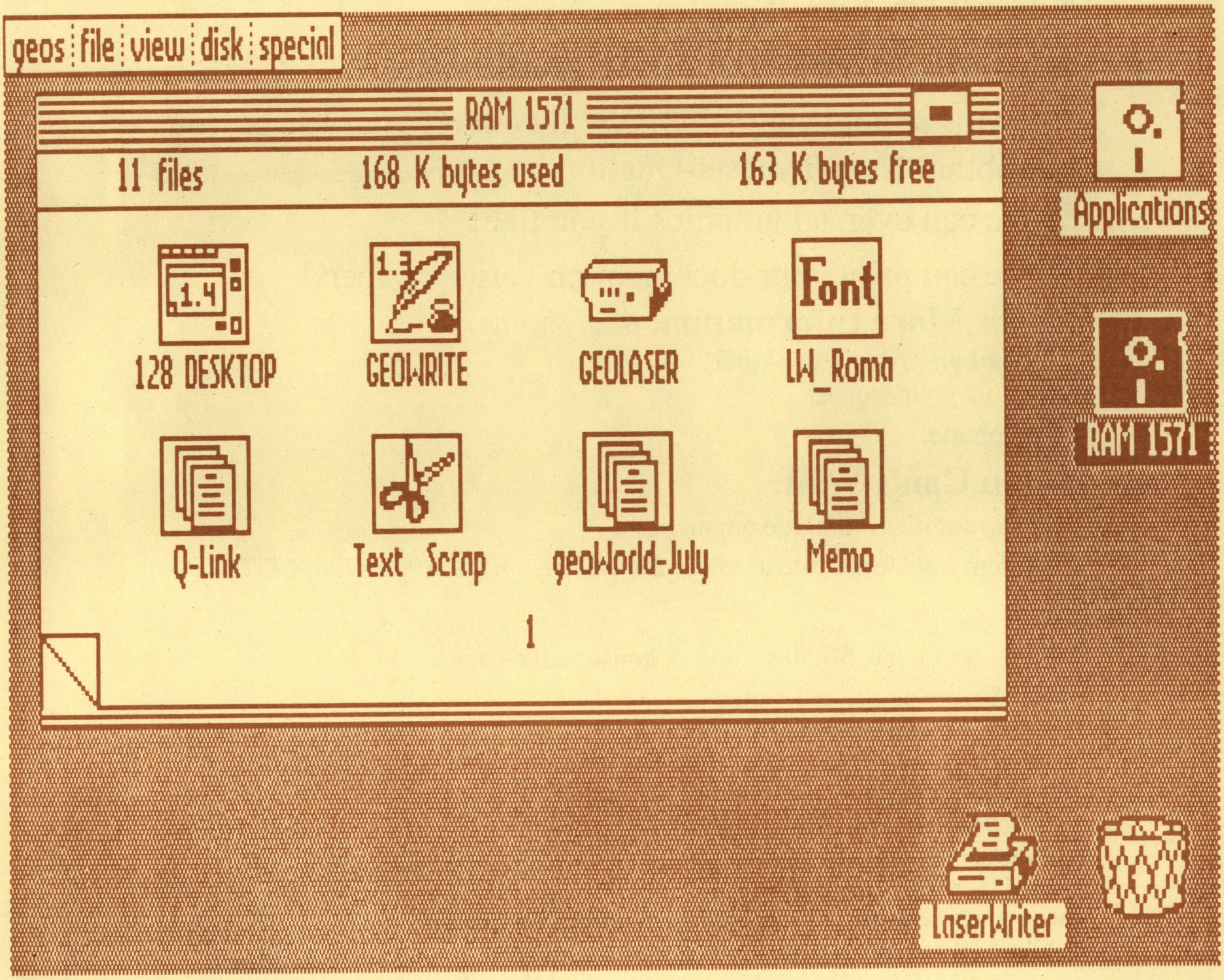
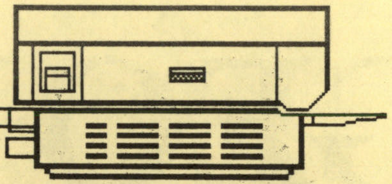
GEO WORLD

August, 1987 Volume 1 issue 5

Published with a Commodore 64/128 using the Graphic Environment Operating System



FAST LASER PRINTING



FAST Laser Printing means:

* GEOS documents once received will be printed & in the mail back to you within 24-hours!!!

* Or else you don't pay!

FRIENDLY Laser Printing means:

* If you're not using GEOS we'll convert your file so you can obtain Near-Typeset-Quality! (Conversion charges apply-see flyer.)

* We can even ad graphics if you like!

* We can print your document on various papers!

3 Ways for More Information: *Send for Flyer!*

* E-Mail via Q-Link to SkipG1.

* Mail us your request.

* Telephone.

Or, If You Can't Wait:

* Mail your disk. (Include payment.)

* Modem transfer. Up-load to our *LaserDirect* BBS. 10pm-7am CDT.

Rates:

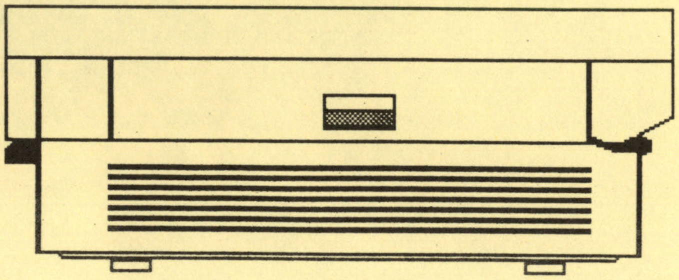
* $2.00 first page, $0.50 second & consecutive pages.

* $0.25/page additional copy.

Where:

Pro-Mark associates
POB 20913
Milwaukee, Wisconsin 53220-0913
(414) 543-9923
(Above # is the *LaserDirect* BBS # from 10pm-7am CDT)
9-5 daily (Generally here weekends)

All details and rates are contained within current flyer.



This Ad was created using GEOS GeoWrite only. It contains no mechanical paste-up of any kind.

GEOWORLD

August 1987

Publisher
Roger Ledbetter
Editor
Skip Goetzinger

Contributers
Mystic Jim
Scott Boydman
Master Blaster
Skip Goetzinger
Terry Mills
Rob Siegel
James Hastings

GEOWORLD is not affiliated with Berkeley Softworks, Qlink or Commodore Computers

Articles may be submited by modem to Fast Laser Printing service BBS (414) 543-9923 or by mailing on GEOS disk to:

GEOWORLD
38 Santa Ynez St.
Santa Barbara, CA
93103

Subscription
6 monthly issues
$10.00

As the world GEOS by...

I hope Everyone is having a wonderful summer! I'm sure that You have noticed that this is the August issue #5 rather than July issue #5. Ever since issue #1, which was mailed on March 18, I have been trying to publish it earlier in the month. Because GEOWORLD has been a one person operation and growing in size and circulation, it has been impossible for me to make up time mailing late in the month. Well July is almost over so now You can enjoy GEOWORLD early in the month. I promise that You will still get 6 issues for the $10 charter subscription.

Skip Goetzinger has offered to become editor of GEOWORLD. What this means is that He will now format the articles to be printed on the laserwriter and send them to me for publication. When I get the laserwriter (hang on folks they don't grow on trees!), printing GEOWORLD and other publications with GEOS will be very easy. As publisher, I will still be looking for material for the magazine and will still accept articles on disk sent to me by mail or will call You for modem transfer. By making this change, I think things will continue to become bigger and better.

Now, lets talk about GEOS. Last month GEO FILE was released and if You have had a chance to see it I'm sure You would agree it is by far the most powerful and versitile data base available for the Commodore 64/128. Unfortunately there has been some bugs reported in the program and BSW has stopped distribution until they can fix them. BSW assures Everyone who bought GEO FILE will get a revised copy as soon it becomes available.

It is obvious that there are alot of excellent fonts becoming available for GEOS. Last month Rabid Concepts introduced Fox Fonts 1, and now TT Grafix premiers Fonts 1. These fonts rival the quality from "other" desktop publishing systems and the price is right. Many hours are spent making fonts and They ask very little when You consider the talent it takes to create them. Please give them Your support.

A new service is available to GEOWORLD readers. If You have any questions You would like answered by Authors of the monthly columns or feature articles, send them to us, they will be answered and published in a special column.

That's it for this month. Please keep Your ideas and suggestions coming in, I appreciate the kind help Everyone has offered.

Roger Ledbetter - Publisher

Back issues available
$2.00 postpaid
issue # 2, #3, #4

Quad Density Printer Drivers? You Bet!

By Terry R. Mills

A few months ago I wrote about an Epson FX-80 driver that made two passes per line for a darker, denser printout. This month I'd like to share with you how to get an Epson (or compatible) to print using quad density mode.

While GEOS is geared primarily for 80 dot per inch (dpi) output, and, for appropriate printers, sends data to the printer based on this assumption, it is possible to modify a driver to invoke a printer's quad density (240 dpi) mode. Because we will be tripling the number of dots printed per inch, it is necessary to send an extra two bytes of graphic data so that what is printed still resembles what is on the screen. In effect, a quad density driver allows us to pack more dots into the same, or slightly wider, space. What would print as a single dot with the normal Epson FX-80 driver is now actually three slightly offset overlapping dots. Two adjacent dots becomes six dots, each overlapping the one next to it. And a straight horizontal line, formerly made up of equally spaced adjacent dots, is now in fact a straight horizontal line!

One way to implement these changes in a printer driver is to add new code at the end of an existing driver, such as the normal Epson FX-80 driver or the EPSON DBL.PRT driver that prints each line twice, and then overwrite the existing code where necessary to call these new patches. Of course, the code we choose to overwrite must then become part of the patches, but this is no problem. In particular, we must get some extra code to execute during the SetGraphics and SendBuff routines described on pages 358 and 359 of The Official GEOS Programmer's Reference Guide.

A "simplified" description of part of how a printer driver works is useful, perhaps mandatory, to understanding the significance of the modifications made in a quad driver. (All right, everybody get out your Programmer's Reference Guide and follow along!)

The PrintBuffer routine (p. 353) calls the PrintPrintBuffer routine (p. 355) once for each line to be printed. PrintPrintBuffer calls the TestBuffer routine (p. 356), which has the job of determining if the data in the buffer area has any printable data in it. If not, upon returning to PrintPrintBuffer no data is sent and control is transferred back to the PrintBuffer routine, which then calls Greturn (p. 360) to do a carriage return and linefeed. However, if data is found in the buffer, it is the job of TestBuffer to update the value of the variable cardcount. Note that TestBuffer does this by starting at the 80th card (there are eight bytes per card for a possible 640 bytes of graphic data per line) and working backwards until it finds data or has determined that there is no printable data in the buffer for that line. Cardcount is therefore not the number of cards with data to print, but the number of cards on the right side of the buffer without printable data!

Upon returning to the PrintPrintBuffer routine, the SetGraphics routine gets called. Because of the way graphics data is managed on Epsons, at the beginning of a line of graphics the computer must send printer codes telling it what graphics mode to use (normal, double density, quad density, etc.) and the number of bytes of graphic data to expect, followed by the data. To accomplish this task, the SetGraphics routine must convert cardcount into a number of bytes by multiplying it by eight. (This is done witr a series of ASLs and ROLs.) But remember, cardcount is really the number of cards on the right side of the buffer without printable data, so this new number must be subtracted from 640 to obtain the number of printable bytes to send. This two-byte number is then stored in the .byte table labelled "wsdgphtbl." (I'll guess and suggest this means something like WidthofSingleDensityGraPHicsTaBLe.) A pointer is then set to point at

Continued . . .

Quad Density Printer Drivers
Continued . . .

this .byte table, the number of bytes in the table (five) is loaded into the accumulator, and the Strout routine (referrred to but not included in the Programmer's Reference Guide) is called. Strout sends a string of data to the printer, working from the back of the table to the front. SetGraphics is left and control is returned to the PrintPrintBuffer routine.

Back in PrintPrintBuffer, the number of cards (not bytes) to send is transferred to X and a loop is entered, rotating cards and sending them until all the cards with data are printed. It is in SendBuff where, finally, a byte of graphic data is sent to the printer. When all of the data is printed, control is returned from PrintPrintBuffer to PrintBuffer where, as mentioned before, Greturn is called and a carriage return and linefeed are sent. This concludes the highlights of the actions of a printer driver as pertinent to an understanding of modifying one to do quad density printing.

When laid out this way, it becomes easier to see what modifications are required to get a driver to print in quad density. First, we have to change the control code that sets the density mode the printer uses. Second, we have to change the values SetGraphics calculates and stores in the .byte table by multiplying it by three and restoring our new values in the table before Strout sends it to the printer. And third, we have to send each byte of graphic data three times instead of just once in order for printouts to be proportioned the same as if printed with an unmodified driver. My modifications to SetGraphics and SendBuff, as well as my routines Times3 and Print3, at the end of this article, illustrate one way to implement these changes. In the interest of those struggling with the Programmer's Reference Guide, I am also including the five subroutines referred to in the sample Epson driver that for some reason didn't find their way into the book. These are the OpenFile, CloseFile, OpenPrint, ClosePrint, and Strout routines.

Working versions of single and double printing quad density drivers for Epsons and compatibles are located in the GEOS software library-printer driver section of QuantumLink.

Modified SetGraphics Routine

```
SetGraphics:
    lda #$00  ;This is the macro LoadB r3h,#0
    sta r3h   ;The pseudoregister r3h is location $09
    lda cardcount
              ;This is the macro MoveB cardcount,r3l
    sta r3l   ;The pseudoregister r3l is location $08
    asl r3l   ;This series of ASLs and ROLs is how the
    rol r3h   ; value of cardcount is multiplied by 8 to
    asl r3l   ; obtain the number of bytes in those cards
    rol r3h
    asl r3l
    rol r3h
    sec
    lda #$80  ;This is perhaps not readily apparent, but
              we
    sbc r3l   ; are simply subtracting the number of
              bytes
    sta wsdgphtbl+1
              ; from above from 640 to determine
              how many
    lda #$02  ; printable bytes of data to send and then
    sbc r3h   ; storing the values in the .byte table
    sta wsdgphtbl;
              ;Here we overwrite some code to transfer
              control to the routine that will multiply by 3
              the values just
              ; placed in the .byte table wsdgphtbl
              ;
    jsr Times3
    nop       ;
    lda #>wsdgphtbl
              ;High byte of address of .byte table
    sta r3h
    lda #$05  ;Number of codes in .byte table
    jmp Strout;Sends the codes
    wsdgphtbl:
    .byte     $07,$80
              ;These 2 bytes tell how many bytes of data
              ; the printer should expect
              (changes each line)
    .byte $03,$2A,$1B
              ;These bytes set quad density mode
```

Quad Density Printer Drivers
Continued . . .

Modified SendBuff Routine

```
SendBuff:
    ldx #$00
    l0$:
    txa
    pha
    lda prntblcard,x
                    ;get byte to send
                    ;
                    ;Instead of sending byte
                    once, call routine tosend
                    it 3 times
                    ;
  jsr  Print3;
  pla
  tax
  inx
  cpx #$08
  bne l0$
  rts
```

Times3 and Print3 Routines

```
Times3:
    lda wsdgphtbl+1 ;Get low byte
    sta templo
    lda wsdgphtbl
                    ;Get high byte
    sta temphi
    asl templo      ;Double the number
    rol temphi
    clc
    lda wsdgphtbl+1
    adc templo      ;And add original value to product
    sta wsdgphtbl+1 ; (in effect, tripling the original)
    lda wsdgphtbl
    adc temphi
    sta wsdgphtbl
    lda #<wsdgphtbl ;This is the beginning of the
                    macro partially
    sta r3l         ; overwritten by code that jumps
                    here
    rts
    templo:.byte $00
    temphi:.byte $00 ;

Print3:
    jsr  Ciout      ;Kernal routine at
                    $FFA8 to send a
                    byte
    jsr  Ciout      ; over the serial bus
    jsr  Ciout
    rts
```

The "Lost Routines" from the Sample Driver

```
OpenFile:
    lda #$04
    jsr  Listen     ;$FFB1
    lda #$F5
    jsr  Second     ;$FF93
    jsr  Unlsn      ;$FFAE
    rts             ;

CloseFile:
    lda #$04
    jsr  Listen     ;$FFB1
    lda #$E5
    jsr  Second     ;$FF93
    jsr  Unlsn      ;$FFAE
    rts

OpenPrint:
    lda #$04
    jsr  Listen     ;$FFB1
    lda #$65
    jsr  Second     ;$FF93
    rts

ClosePrint:
    jsr  Unlsn      ;$FFAE
    rts

Strout:
    sta scount      ;See p. 366 RAM Storage/Utilities
    dec scount

l0$:
    ldy scount
    lda (r3),y      ;r3 is the pseudoregister at $08-$09
    jsr  Ciout      ;$FFA8
    dec scount
    bpl l0$
    rts
```

HOW TO EXPAND THE 1764 TO 512K

by Scott A. Boydman

The 1764 RAM EXPANSION UNIT (REU) consists of a plug-in cartridge, a deluxe power supply, and a disk containing software for using the REU with both BASIC and GEOS. The power supply is required, since the original C-64 power supply was not designed to meet the power requirements of the great many chips in the REU. The power supply is rated for 2.5 amps on the 5 volt DC side, and 1 amp on the 9 volt AC side, although apparently, it is only the DC current that supplies the REU. The REU itself is useless without appropriate software. The back side of the disk contains a GEOS upgrade which installs GEOS Kernal V1.3 on the system boot disk, as well as other upgraded applications and input/printer drivers. The front side of the disk contains RAMDISK software for use with BASIC. The GEOS and BASIC RAMDISKS are not compatible. The REU test program is designed to test the 256K of ramdom access memory that is supplied in the original 1764.

The 1764 REU itself consists of a printed circuit (PC) board inside a plastic and metal housing, which plugs into the expansion port of the C-64 or 64-C computers. The PC board is the same board that is used in the 1700 (128K) and 1750 (512K) REU's, intended for the C-128, but with the following changes:

1) The 1764 has only **one** bank of eight 256K Dynamic RAM chips installed. Each chip holds 256K **bits** of information; therefore eight chips are required to obtain 256K **bytes** (8 bits make up 1 byte).

2) There is an **empty** bank of eight positions labeled **BANK II** on the PC board. The traces for additional memory chips are present, and the holes for the pins on the chips are presoldered.

3) There is a resistor in the 1700 and 1750 REU's that is **not present** in the 1764, apparently due to a small difference between one of the expansion port lines in the C-128 and the C-64. In spite of this difference, there are C-64 users that have installed **unmodified** 1750 REU's without reporting problems. In these installations, larger power supplies rated at three amps (for the DC circuit) are used to supply the additional power to the REU.

Commodore most likely decided to ship the 1764 with 256K in order to keep the price of the unit lower and make it attractive to as many C-64 owners as possible. Only by selling a significant quantity of REU's would it be possible to entice software writers (other than **GEOS**) to support the unit. The **RAMDISK** software included with the 1764 is a nice option, but it is not compatible with many software packages.This is due to the fact it **only** supports the four standard file types (PRG, USR, SEQ, and REL), **but not** direct track-sector access, which many programs use for loading and other functions. GEOS, however, treats the expansion as an autonomous unit when in the **RAM 1541** mode, and it is used exactly as a second disk drive. There is another option available as well, namely the **shadow drive.** This mode causes GEOS, when reading files, to also load them into the REU, so that further accesses occur directly from RAM. This results in a significant speed-up by reducing multiple disk access. All writes go to the physical drive, which eliminates the **primary disadvantage** of a RAM disk; namely the file has to be saved to a **real drive** before your computer is turned off, or the data will be **lost.**

The 256K REU does not have the amount of memory needed to shadow **two** physical 1541 drives, and for this reason many people wanted the 1764 to be delivered with 512K. It is not too difficult for someone with a knowledge of soldering to install the additional chips in BANK II on the PC board. I recommend soldering 16-pin **DIP (Dual Inline Package)** sockets to the board, which prevents heat damage to the memory chips. The chips are simply pushed carefully into the sockets. Radio Shack stocks the 16-pin DIP sockets (Catalog #276-1998, $.89/2) and 256K RAM chips (Catalog #276-1252, $6.95 each). The chips are also available from mail order houses at a discount.

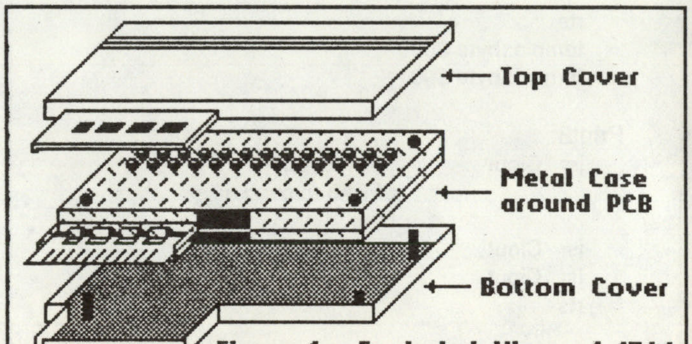


Figure 1: Exploded View of 1764

NOTE: The steps required to install the chips should only be undertaken by someone skilled in handling PC boards, and some mechanical skill is needed to remove and install the PC board in its housing. Opening the REU housing will void Commodore's warranty, and is done at the owner's risk.

Step 1:
The housing of the 1764 must be opened. There are four pins, one in each corner, holding the plastic case together with friction. Starting at the opening which connects the unit to the computer's expansion port, gently but firmly pull the top and bottom halves apart. The plastic case will begin to separate. Continue around the perimeter of the case until the top can be removed from the bottom. **Figure 1** shows an exploded view of the disassembled 1764 REU.

Step 2:
Remove the metal shield from the plastic casing by pulling it out at each corner. The PC board is inside the metal shield. Open the shield, also starting at the connector. You may need a small screwdriver to pry the metal slightly where it is held together. When the shield opens, spread it only as much as needed to remove the PC board. Then slide the cardboard liner off from the expansion connector.

Step 3:
Examine the PC board. **BANK I** is the row of eight chips along the top of the board. The **RAM Expansion Controller (REC)** is the square chip in the lower portion of the board. The empty **BANK II** can be seen just below BANK I. Refer to **Figure 2** for a schematic representation of the 1764 PC board.

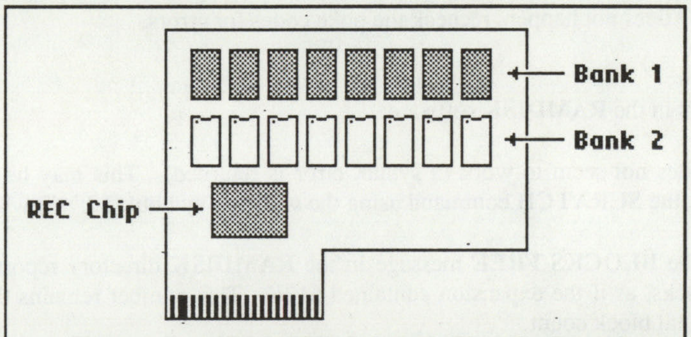


Figure 2: 1764 REU Printed Circuit Board

Step 4:
Solder **eight DIP sockets** into the locations in **BANK II**. This is a difficult step, because it is hard to position all 16 pins at once. I recommend that a small needle or paperclip be used to open the 16 holes on the PC board for each socket position. Using a soldering iron, heat the hole position until the needle can be inserted. By rotating the needle during cooling, it will not bond to the solder, and a hole will be left upon its removal. After doing this for all 16 holes, the DIP socket can be inserted into the holes, and then each pin soldered individually. Do this carefully, as **each connection must be secure on both sides** of the PC board.

Step 5:
Now **insert the chips carefully** into the sockets, making certain to maintain the correct orientation. The notch in the chip faces the top of the PC board. You may have noticed two "jumpers" on the PC board. These have been set properly at the factory, and don't require further attention.

Step 6:
Before putting the REU together again, plug the PC board into the computer (make sure the power switch is off), turn the computer on, and boot GEOS V1.3. Run the **UPGRADE conversion program** on the disk that came with the 1764 REU, or 1351 MOUSE to convert V1.2 to V1.3, but be sure to **keep a V1.2 backup copy** should anything go awry. Then from the deskTop, run the **CONFIGURE** program. The RAM EXPANSION box that appears should indicate **512K** installed. If this is the case, reassemble the REU. If only 256K is displayed, either a trace was not soldered properly, a pin on the chip may not be making good contact, or a chip is defective. If this occurs, recheck all the connections systematically, and run **CONFIGURE** again.

Expanding the 1764
Continued . . .

If you arrive at this point, **CONGRATULATIONS!** GEOS automatically uses the entire 512K memory. The **RAMDISK** software included with the REU also uses the full amount of expansion memory; but the **RAMTEST** program was only designed for 256K. The program, however, can be modified to check the entire 512K in the expanded 1764 as follows:

1) Copy the files **1764 RAMTEST.BAS** and **1764 RAMTEST.BIN** from the demonstration disk onto another formatted disk. You may use the **FILE COPY** program included on the disk to do this, or even copy the files from the GEOS deskTop.

2) From **BASIC**, type the following:

```
NEW <RETURN>
LOAD "1764 RAMTEST.BAS",8 <RETURN>
501 REM LINES 502 & 503 POKE THE CODES TO TEST 512K! <RETURN>
502 POKE 33093,7: POKE 33099,33 <RETURN>
503 POKE 36003,53: POKE 36004,49: POKE 36005,50 <RETURN>
```

3) List lines 500-510 to confirm the changes have been entered. Then, list line 100 and delete the characters **100 REM** and press <RETURN>. This will resave the altered version of 1764 RAMTEST.BAS to the disk, and delete the original version. Relist lines 502 and 503, and recheck for errors. Then, run **1764 RAMTEST.BAS**, and the program should test all 512K in the expansion. If this does not happen, recheck the poke codes for errors.

Sidenotes:
I have identified some **bugs** in the **RAMDISK** software:

1) The **NEW** command does not seem to work (a syntax error is returned). This may be an intentional feature for safety purposes. However, the **SCRATCH** command using the wildcard symbol (S0:"*") will still get the job done.

2) With the 256K REU, the **BLOCKS FREE** message in the **RAMDISK** directory reports about twice the correct number (over 2000) of blocks, as if the expansion contained 512K. This number remains the same after the 512K is installed, indicating the actual block count.

The operation of the 1764 with **GEOS** appears perfect. It would have been a nice feature to exit GEOS and directly utilize the **GEOS RAM 1541** files in the **BASIC** environment, but due to the different format of the Commodore RAMDOS, this isn't possible.

I hope this information will be helpful to all who were disappointed that Commodore did not put 512K in the 1764 REU. If you desire further information, feel free to contact:

Scott A. Boydman
25447 Bryden Road
Beachwood, Ohio 44122

or you may leave E-Mail to **ScottB30** on QuantumLink.

Now GEOS isn't just for adults anymore !

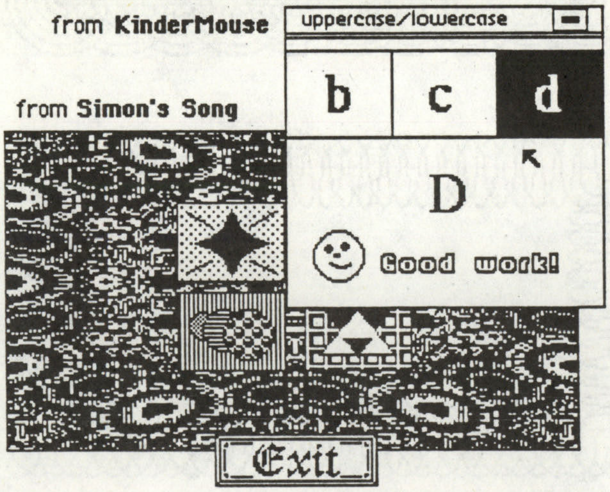
introducing:

geoKids vol. 1

educational/enrichment software for ages 4-7

from **KinderMouse**

from **Simon's Song**



geoKids **vol. 1 contains:**

Kindermouse: Eight activities for practice in some of the skills most important to early learners. These include letter recognition, case and order, numeral recognition and order, matching simple shapes and groups, counting objects, and counting parts of a whole. The child can play any of the activities as long as he/she wants, and switch easily between the activities.

Simon's Song: A memory enhancement game using musical sound and intriguing graphics. Play level is user-selectable.

geoKids **vol. 1 is now available for only**

$12.00 *

User groups inquire about special offer.

*Kansas residents please add 5% sales tax

dataDomain Software
2114 SW 71st Terrace
Topeka, KS 66619

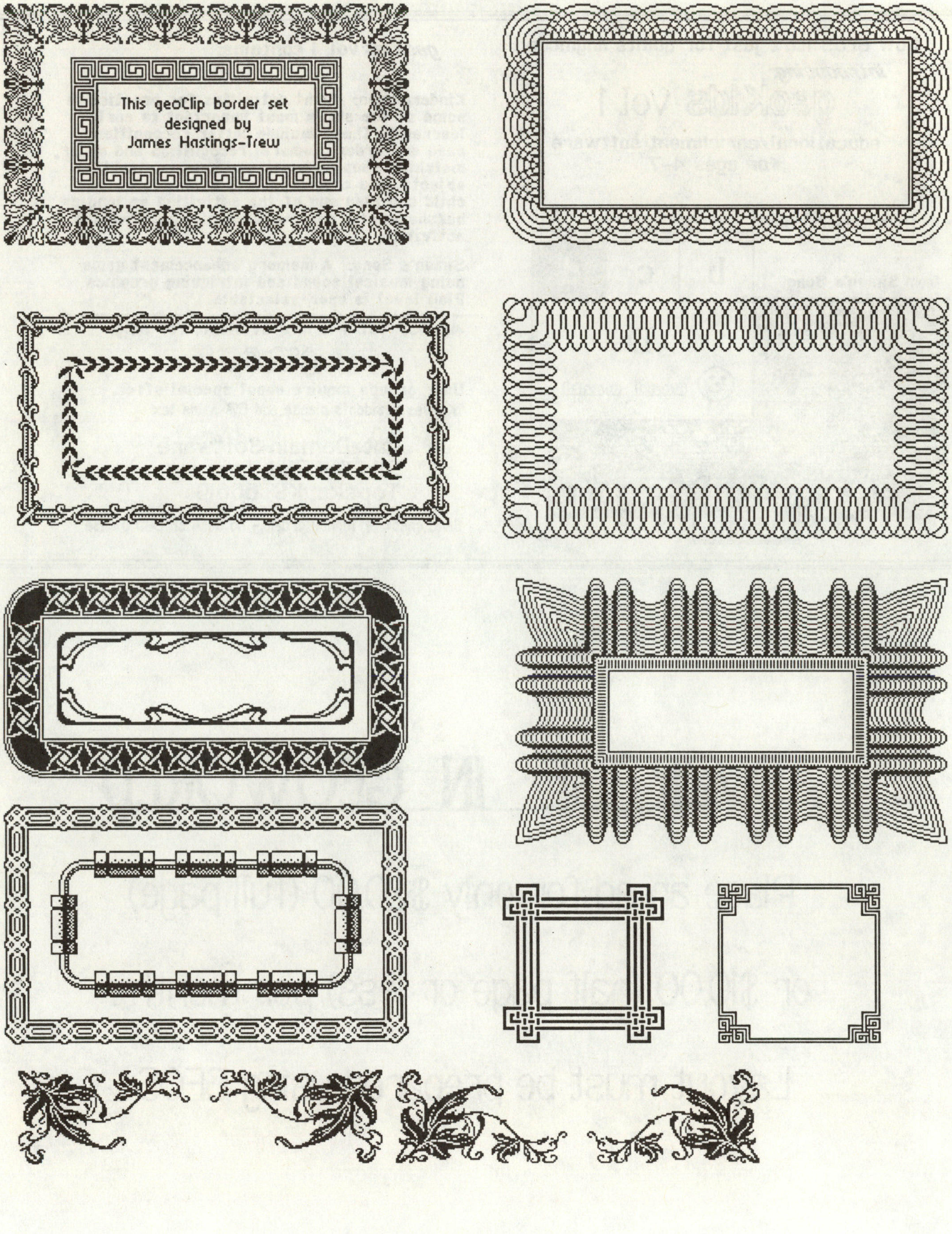
Send check or money order - no cash or COD please

ADVERTISE IN GEOWORLD

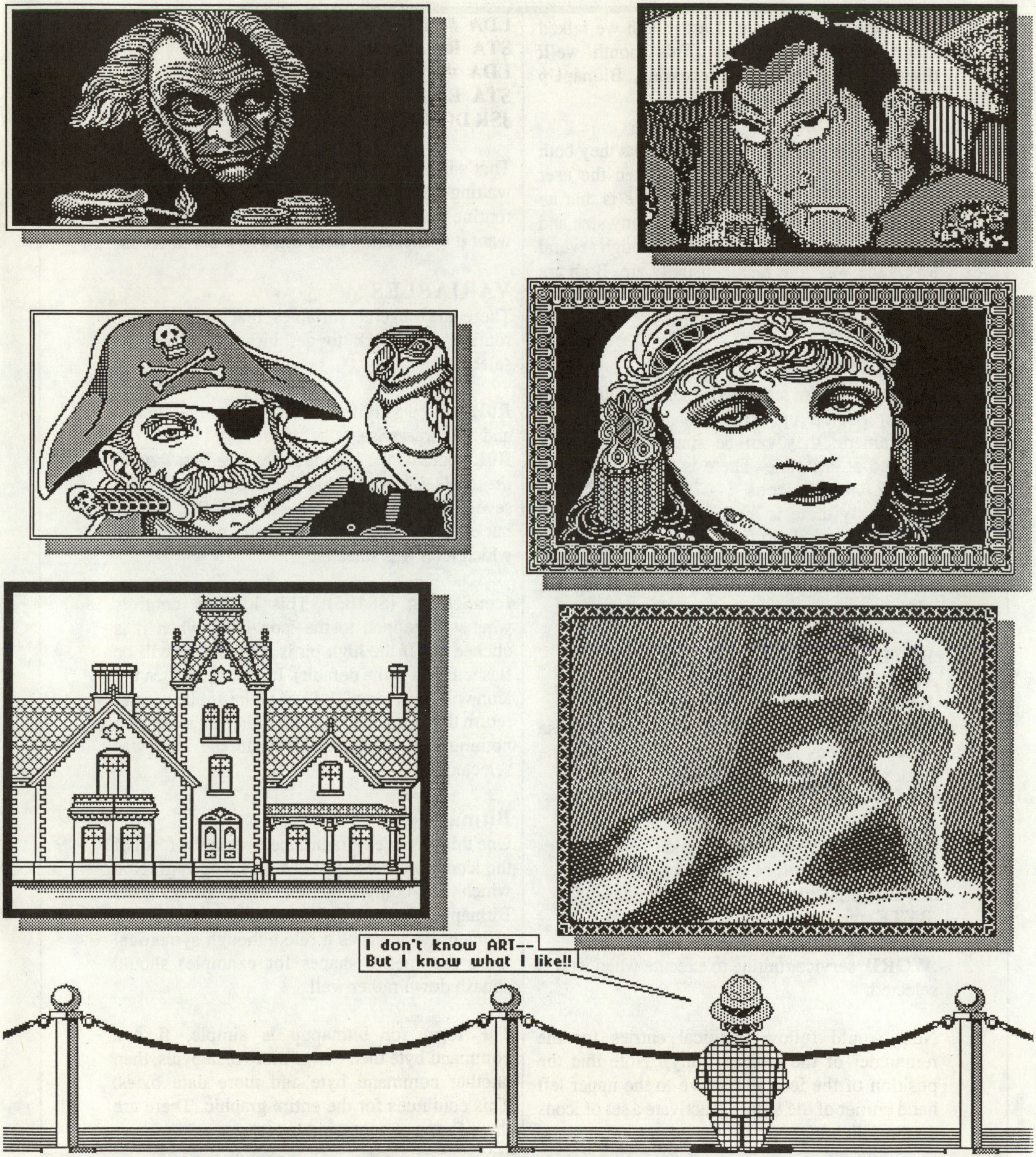
Place an ad for only $20.00 (full page)

or $10.00 (half page or less) per month .

Layout must be prepared using GEOS .



This geoClip border set designed by James Hastings-Trew


I don't know ART--
But I know what I like!!

INSIDE GEOS

MASTER BLASTER

Well here we are again! Last month we talked about events and menus. This month we'll discuss icons and icon processing, BitmapUp format, and a few other goodies.

Icons are very similar to menus in that they both cause a routine to be executed when the user clicks on them. The main difference is that an icon can be designed in practically any size and shape and is normally not nested through several levels the way that menus usually are. Both are set up in basically the same way -- via a table structure.

INITIALIZATION

Usually icons are intialized when the application is first booted up. This, of course, is not a requirement; they can be started, stopped, or changed at any time. There is only one routine dealing with icons -- **DoIcon.($C15A)**. Supposedly there is an inline form but if so I haven't been able to find it. To use this routine simply pass the address of the icon definition table to the routine in R0. The table is set up like this:

.BYTE # of icons.
.WORD X position of mouse (in pixels).
.BYTE Y position of mouse (in scanlines).
The position defines where to place the mouse when the icons are initialized. Next comes the entries for the icons themselves:

ICON #1
.WORD pointer to icon graphic.
.BYTE X position of icon (in bytes).
.BYTE Y position of icon (in scanlines).
.BYTE width in bytes.
.BYTE height in bytes.
.WORD service routine to execute when icon is selected.

Next would follow identical entries for the remainder of the icons (if any). Note that the position of the icon is relative to the upper left hand corner of the icon. To activate a set of icons do the following:

```
LDA #< ICONTABLE
STA R0L
LDA #>ICONTABLE
STA R0H
JSR DOICONS
```

That's it ! Your icons will now be displayed and waiting for the user to click on them. The service routine for an icon can do about anything you want it to including defining a new set of icons.

VARIABLES

There are several variables that your service routine can check to get information on the selected icon:

R0H will be $FF if the icon was double-clicked and $00 otherwise.
R0L contains the icon number (*the first icon in the table is zero*). This is handy if you have several icons that use the same service routine but execute somewhat differently depending on which icon was selected.

IconSelFlag ($84B5): This location controls what will happen to the icon itself when it is clicked on. If the high bit is set the icon will be flashed (this is the default). If bit 6 is set then the icon will be inverted. It is up to the application to return the icon to normal. If neither bit is set then nothing will happen to indicate the icon was selected.

BitmapUp

One thing I haven't mentioned yet is the form of the icon picture itself. It is a bit-mapped graphic which is compacted it what BSW calls BitmapUp format. The compaction is not very efficient for icons as a rule although symetrical icons (geometric shapes for example) should squash down rather well.

The form for bitmapup is simple, first a command byte then one or more data bytes, then another command byte and more data bytes. This continues for the entire graphic. There are three forms of command bytes:

Bytes between 0 and 127 mean to repeat the following byte X times, i.e. 100,255 would become 100 255's in a row.

Bytes between 128 and 220 mean to use the next X-128 bytes once each, i.e. a straight bitmap. Obviously the most bytes that can be used for each command byte is 92 (220-128).

Bytes between 221 and 255 are the most complicated. The form is as follows:

COMMAND BYTE COUNT PATTERN

The command byte (minus 220) is the number of bytes in the pattern (max 92 again). Count is the number of times to repeat the pattern. Note that the pattern must be enocoded in one of the two other formats. This form provides the most compaction but obviously can only be used if the graphic contains a repeating series of bytes. Here's an example. Let's say you need to encode the following series of bytes:

12 56 34 67 12 56 34 67 12 56 34 67

First encode a single group. Hmm, you would have to use 4 12 56 34 67. The pattern contains five bytes and we need to repeat it three times. So the finished command is:

225 3 4 12 56 34 67

Which saves us five bytes. Of course the bigger the pattern or the greater the number of iterations the greater the savings.

Note that graphics are decoded **scanline by scanline** rather then by cards as with C-64 sprites. When you are in a hurry the quickest way to encode a graphic is to place a count byte in the front. If the graphic is bigger then 92 bytes just stick a 92 in front and insert another count byte before the 93rd byte. You can continue this until you've defined the whole graphic.

On and off

If the picture pointer of the icon (the two bytes in the icon table that point to the icon's graphic image) are set to 0,0 the icon will be disabled. One thing to remember is that it won't be erased from the screen, you have to do that yourself either by calling RecoverRectangle or by drawing something over the top of it. If the pointer is set to zero before DoIcons is called then the icon will not even be displayed.

If you need to disable all the icons on the screen the easiest way to do it is to call DoIcon again. If you don't want to have any icons active just set up a table containing a single icon that's one byte wide and one scanline high with a picture pointer of 0,0. The older versions of the Kernal required at least one icon be defined and this was the easiest way to create a dummy icon. If you don't know what version of Geos your application will be run on, always define at least a dummy icon.

ICONS VS MENUS

For the most part icons and menus can be interchanged, at least if converting from icons to menus. Menus with many levels of nesting can't be converted over very easily. The main advantage of icons is a cosmetic one - icons look nicer (or should anyway). If your writing under a deadline you may want to consider menus because they're quicker to code (no graphic data).

Well that's about it for now, next time we'll be discussing Dialog Boxes. Hope to see you there.

Desk-Top Publishing

by Skip Goetzinger

For the infrequent user or non-professional, desk-top publishing can be a confusing or elating experience. But what about the pro? What about the well-adjusted, non-pro user of GEOS?

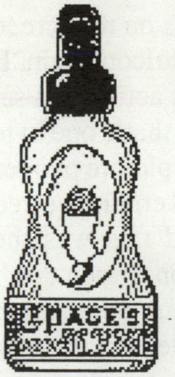
Cut & Paste or Overlay?

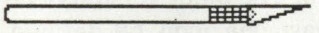
There are times when "pasting-in" your photos, or mechanically creating your page is faster and easier than formatting the page within GEOS. I personally tend to format my doc's within GEOS if I:

1) Want an electronic "master" for future updating.
2) Know that I will be generating multiple copies off of the LaserWriter.

I will create a semi-formatted page, or, galley type within GEOS if I:

1) Need a higher resolution photo to be pasted into the doc.
2) Need quick typesetting.

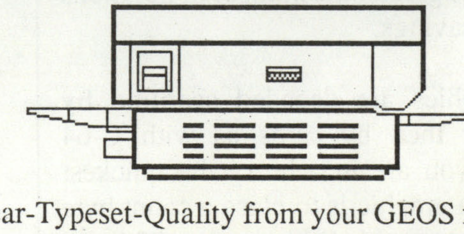




The 1750 & Desk-Top Publishing

The 1750 has greatly enhanced the speed of the GEOS system. What you will find concerning some previously discussed quirks of GEOS & the Commodore:

- * Pic scatter will still occur when you move from one side of the page to the other. (e.g. left to right) Make sure you select "hide pictures" from the Options pull-down menu. This will eliminate you locking-up your system.
- * Your Photo Albums must be on the same disk as your Photo Manager. (The same holds true for the Text Manager.) This application does not have a "drive" select option as does GeoLaser, for example. Knowing this ahead of time will eliminate wasted time once inside of a file.
- * The 1750 allows GEOS to truely act and perform as a professional Desk-Top Publishing system. Added speed, Shadowing (which opens up your disks quite a bit for added files) and simplified file transfers all add-up to a top quality system.
- * I am currently using only one drive (1541), and am experiencing no difficulty at all in transferring files between disks. I simply use the RAM.



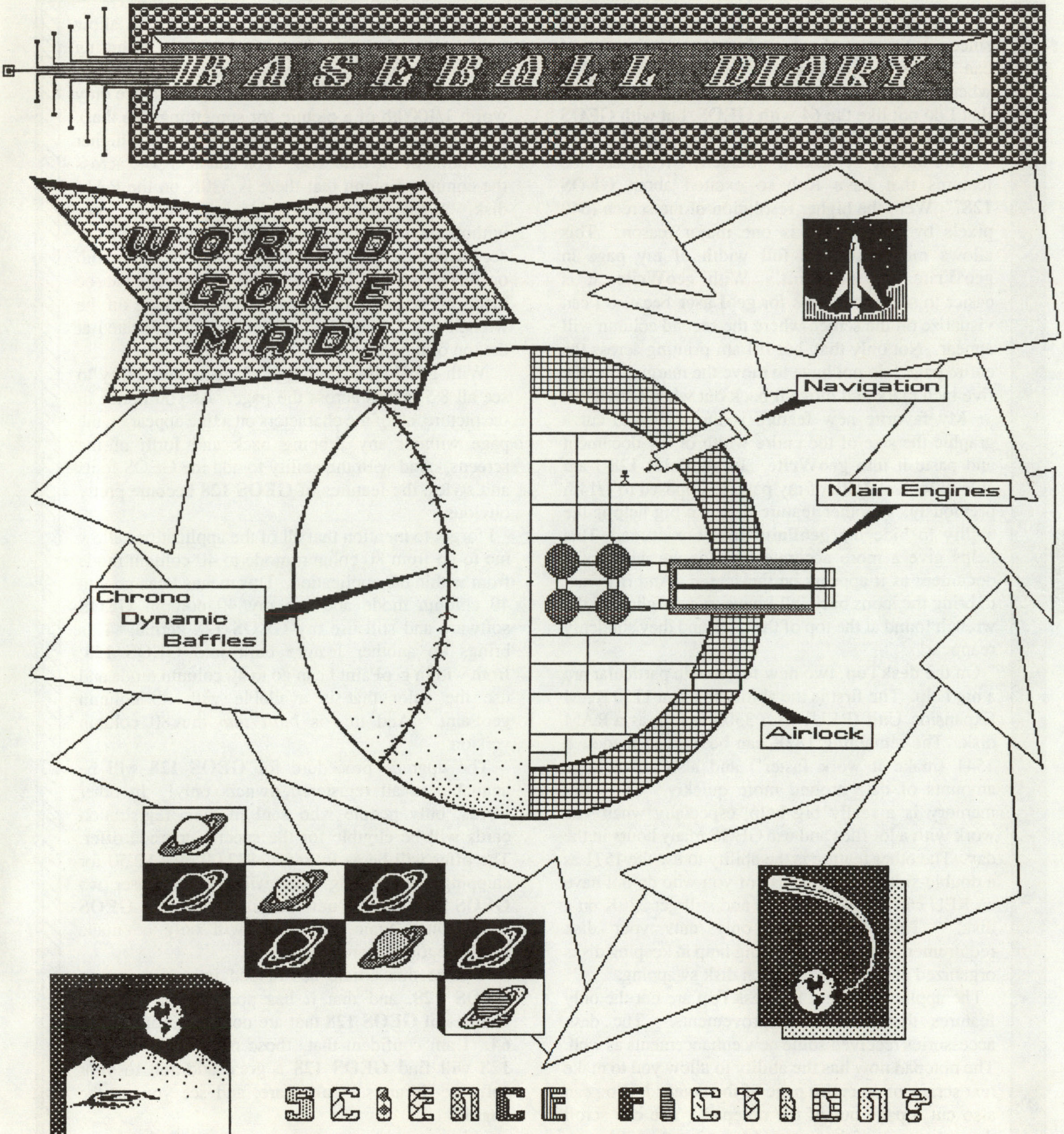
Do I have to have the LaserWriter?

If you want to achieve Near-Typeset-Quality from your GEOS files, YES, you must have, or use an Apple LaserWriter. In order to achieve NTQ, GEOS requires PostScript, a page description language, in order to print out your files with high resolution. Even if a Laser advertises itself as having "resident" fonts, but fails to include PostScript, you will not be able to achieve the type quality most of use are lookig for. The HP LaserJet is a perfect example.

If you're considering the purchase of a LaserWriter, *MAKE SURE* that you have enough business to pay for it in a years time. There is an option now on the new tax laws, that allows you (starting with '86 taxes) to take a business deduction/write-off of up to $5000 in one year. $4000-plus is alot of cash to tie-up in one piece of equipment.

Have to finish-up the rest of GEOWORLD.
Happy Publishing!

Created by John Hastings (Johnny 99)



Berkeley Softworks

This month I would like to discuss **GEOS 128** since it is the next of our products to start shipping. I can honestly say that I will use **GEOS 128** whenever possible for my computer needs. It is not that I do not like the 64 with **GEOS**, but with **GEOS 128** I have a hard time getting off the system!

You might be asking yourself, "What are the features that have Rob so excited about **GEOS 128**?" Well, the higher resolution of the screen (640 pixels by 200 pixels) is one major reason. This allows me to see the full width of my page in **geoWrite** and **geoPaint**. With **geoWrite**, it is easier to set up columns for **geoLaser** because I can visualize on the screen where the second column will appear. Not only that, but if I am printing across the entire page I do not have to move the margin in to the five inch mark and move it back out when I am done.

My favorite new feature is the ability to cut a graphic the size of the entire width of my document and paste it into **geoWrite**. In **geoPaint 128** I am able to look at 1/5th of my page as opposed to 1/14th previously. Another feature that is a big help is the ability to hide the **geoPaint** tool box icons. This helps give a more accurate feel for the width of a document as it appears on the screen. And if I want to bring the icons back, all I have to do is click on the wrench found at the top of the page and they suddenly reappear!

On the **deskTop**, two new features in particular are a big help. The first is the ability to use a 1750 RAM Expansion Unit (REU) with 330K set up as a RAM disk. The remaining 182K can be used to shadow a 1541 (make it work faster!) and also move large amounts of data around more quickly. The extra memory is a really big help, especially when you work with a lot files and use GEOS many hours in the day. The other feature is the ability to use the 1571 as a double-sided drive. Those of you who do not have an REU can use your 1571's and still get 330K on a disk. This feature not only cuts your disk requirements in half, but is a big help in keeping disks organized and cutting down on disk swapping.

The applications and the **deskTop** are not the only features that received improvements. The desk accessories received some new enhancements as well. The notepad now has the ability to allow you to make text scraps of an entire page of the notepad. You can also cut a page out of the notepad. You can scroll through pages of the notepad by using the keyboard, as well as using the Commodore key and the letter Q to quit. The calculator, like the notepad, will allow you to cut and copy to a textscrap simply by hitting the Commodore key and the letter C.

Now you know the differences, but words are only worth 1/1000th of a picture (or something like that). On the page opposite this you can see the higher resolution of the **deskTop**. You can also note across the command menu that there is 330K on the RAM disk. What this does for file manipulation goes without saying. On the **geoPaint** screen, you can see the before and after difference with the icons both on the screen and later removed. The icons can be made to appear or disappear just by clicking on the wrench that I mentioned earlier, which is located at the top of the screen.

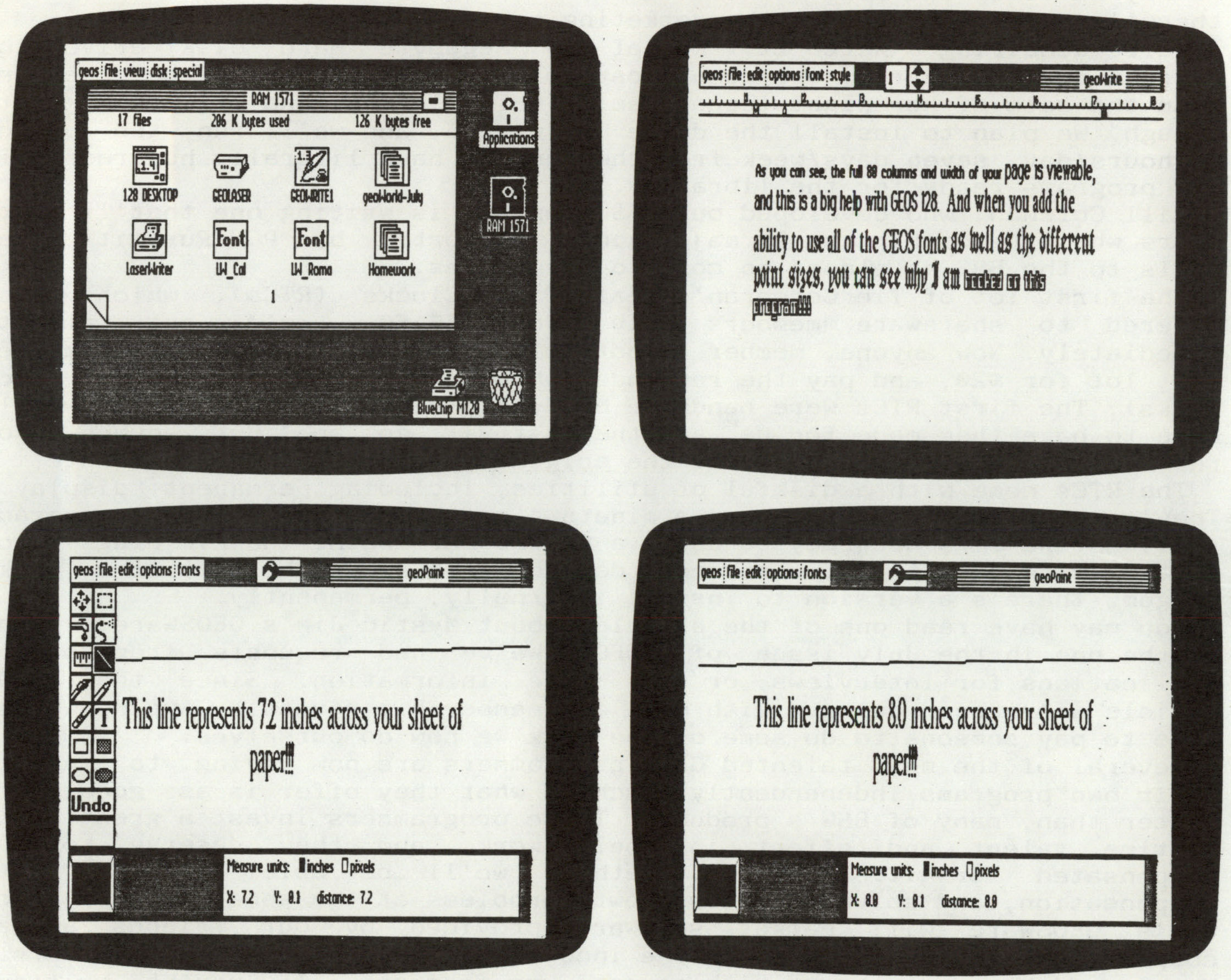
With **geoWrite**, the biggest help is the ability to see all 8.5 inches across the page. As you notice in the picture, all of the characters on a line appear on the page without any flipping back and forth of the screens. And with the ability to add the GEOS fonts and styles, the features of **GEOS 128** become pretty obvious.

I forgot to mention that all of the applications allow me to go from 80 column mode to 40 column mode from within the application. This means I can go into 40 column mode and use my 40 column **GEOS** software and still use my **GEOS 128** kernal. This brings up another feature built into **GEOS 128**: from within geoPaint I can go to 40 column mode and use the color that is available with 40 column geoPaint. And this is *built into* the 80 column version.

The upgrade procedure for **GEOS 128** will be available to all registered owners only! In other words, only people who sent in their registration cards will be eligible for the special upgrade offer. The offer will be as follows: $22.00 plus $2.50 for shipping and handling. This will give each user two **GEOS** boot disks, a new manual, and a new **GEOS** box. Once again, this offer will only be made available to registered owners.

I hope this article has helped introduce you to **GEOS 128**, and that it has pointed out the new features of **GEOS 128** that are not present in **GEOS 64.** I am confident that those of you who have a 128 will find **GEOS 128** a great addition to your software libraries. Take care, and see you on the boards!

Rob Siegel (A.K.A. GEOS ROB)



As you can see, the full 80 columns and width of your page is viewable, and this is a big help with GEOS 128. And when you add the ability to use all of the GEOS fonts as well as the different point sizes, you can see why I am hooked on this program!!!

This line represents 7.2 inches across your sheet of paper!!!

Measure units: ■ inches □ pixels

X: 7.2 Y: 0.1 distance: 7.2

This line represents 8.0 inches across your sheet of paper!!!

Measure units: ■ inches □ pixels

X: 8.0 Y: 0.1 distance: 8.0

GEOS 128 SCREEN DUMPS

Mystic Jim
2388 Grape
Denver CO
80207

Lots of news this month: we expect to have the BBS on a twenty-meg hard drive by the middle of July; members will soon be able to call the BBS at nearly no telephone cost to themselves; Tim's Timers are sold out, but there'll be more; and we'll be marketing commercial programming soon.

We're acquiring a Xetec Lt. Kernal 20 megabyte hard disk drive to operate our BBS. Barky, our mama frost-point cat, has donated the proceeds from the sale of her eight (count 'em, EIGHT) kitties, and it'll be just enough. We plan to install the drive by July 15, and we'll run the board 24 hours/day, seven days/week from then on. We have literally hundreds of new programs ready for the libraries.

Bill Coleman, who developed our BBS program, is writing one that'll let users who live within the 25 major zones supported by PC Pursuit make calls to the BBS at almost no cost to themselves.

The first lot of Tim Corcoran's Real Time Clocks (RTCs), which were offered to shareware members only, for $37.50, sold out almost immediately. Now anyone, member or not, can reserve an RTC or kit from the next lot for $20, and pay the remainder when RTCs are ready (about two weeks). The first RTCs were handmade by Mystic Jim, but from now on, we'll have to have them made for us, and the cost will go up. But anyone who reserves one now can have it for the original price: don't procastinate!

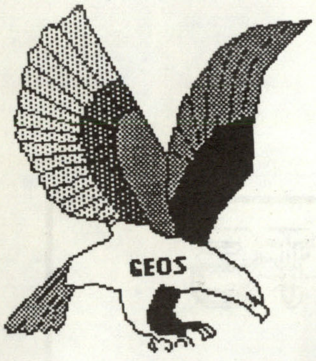
The RTCs come with a diskful of utilities, including permanent display. They're accurate to better than a minute a month and there'll be a program to fine-tune it's accuracy to about a SECOND per month. The RTC plugs into port#2, with pass-thru for an input device. It's about the size of a pack of gum. There's a version to install internally, permanently.

You may have read one of the articles about Mystic Jim's GEOSWare, such as the one in the July issue of INFO. We've had requests from other publications for interviews, or for more information. Since the INFO article we've been deluged with disk and membership requests. Again, we'll have to pay someone to do some of the work we now do ourselves.

Several of the most talented GEOS programmers are now trying to market their own programs independently. Much of what they offer is as good, or better than, many of BSW's products. These programmers invest a great deal of time, talent and effort in their work, and they deserve to be compensated for it. Working together, we'll be able to get that compensation, and solve some our growth problems at the same time. Mystic Jim's GEOSWare will market software provided by our friends as a cooperative effort, and most of the income will belong to the programmers. We already have agreements with several programmers. If you've written GEOS programs, plan to, or would like to, please contact us through our BBS, or by mail.

Of course, we'll also continue to offer shareware as always, and do a much better job. Our shareware members will find that their membership fee is the best forty dollars they've ever spent. We'll be able to acquire more and better shareware programs, and deliver them more efficiently. Members will get substantial discounts on our commercial products, and have access to a greatly expanded BBS.

There's still time to join our shareware group at forty dollars, for which you get a subscription to GEOWORLD, all of our disks, access to our BBS, and more. Please make checks payable to Mystic Jim. You can get our introductory disk, as shareware, by just sending us your name and address.
Best.



MYSTIC JIM'S SHAREWARE UPDATE

DISK#1: New Program Updates
GEONumber: serial# editor (7)
Bootstrap1.3d: distribution (7)
BootstrapG: for all GEOS prgs(7)
GEOMouse: use that other button! Four kinds from Cosmac.
AutoPreference1.3 (3)
PS-GEOS: PrintShop convert.(7)

DISK#2: Introductory
Bootstraps: bootdisk makers(7) for 1.2, 1.3Upgrade

Input Drivers:
Sketchpad (1)
Lightpen: for pens w/switches
PenJoy: hybrid lpen/jstik (7)
Anypen: switchless pens (7)
Inkwell: Flexidraw litepens
KoalaPad: 8 kinds (3/7)
Mouse1351
Mouse2: for port#2 (7)
Mouse1350

ACCESSORIES:
GEODump: print any screen (3)
Change Input: from appli. (3)
Change Printer: (3)
GEOFormat1.3: fast formats (3)
Note Printer: notepad. (3)
Auto Preference: (3)
Watchset: fast time set. (2)
Quick Dateset: (4)
Icon Maker/Sprite Magic (7/x)
Icon Edit2.1: best around (11)
O'Clock: display time. (7)
Convert1.4: for ALL types (9)
GEOTab: tabs for GEOWrite (1)
GrafGrab, Dex/Merge patches.
TextGrab1.3: grab seq text. (6)

GRAPHICS
Imagecon: hi-res/multi/GEOS graphics convertor. (7)
Piconvert: Doodle to GEOS. (PD)
ComputerEyes: GEOS software.
GEOPaint Help Files: (7)

() PROGRAMMERS (semi-alphabetic)
1. Alex Boyce
2. Bill Coleman
3. Art Dahm
4. Jeff Fox
5. Don Mosedale
6. R. Winchester
7. Mystic Jim
8. W. Bruce Moore
9. Ben Taylor
10. Tony Reynolds
11 Terry Mullett
12. Mitch Romm
13. Tim Corcoran
14. Terry Mills
15 Lynn Kerby
16 Andrea Needham

DISK#3: Fonts, Font printouts.
Balloon: 24p
Banner: 36pt.
Dvorak: 18pt.
Esperanto:6,10,12,14,18,24pt.
Greek 12pt.
IBM 10pt.
Fill-48: 48pt.
MegaBanner: 280pt horizontal
Mystic: 18pt.
Oxford: 12,24pt.
Paintbrush: 36pt.
Zapf: 14pt.
Andrea's Calligraphy (16)

* Additional fonts added regularly. Fonts designed using Cosmac's GEOFont, by Cosmac, P. Hughes, Mystic Jim, Andrea Needham, and others. GEOFont is available from Myth Conceptions, 111 New Canaan Ave, Norwalk Ct 06850.

DISK#4: Programming Utilities.
Convert.ml: ml to GEOS. (5)
GEOStart: program to GEOS
prgtogeos: from tech manual.
DISPMEM: disassemble from mem (15)
GEOThreader: single step dis. (7)
GEOLinker: link header block. (2)
GEOSplitter: split header out. (2)
GEOIcon: icons for programmers(2)
Disk Editors, 64 and 128.
Micromon64
GEOCross: cross-ref tech man.
Spelunker, with src. (10)
Tech Manual updates.
Alex Boyce's Tech Manual.

DISK#5: BBS Utilities and Terms.
Fasterm, Multi-Term, CG Term
Library Utilities.

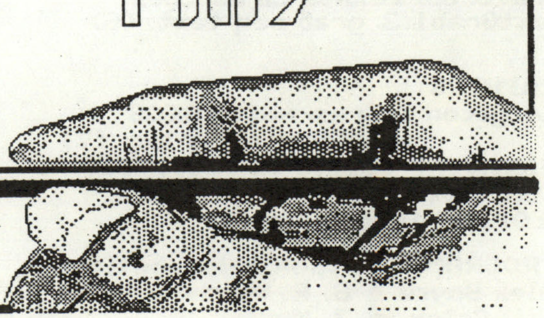
DISK#6. Printer Drivers.
Mostly for user groups.
All BSW drivers,including Laserwriters and GEOCable.
Mills double/quad drivers.
(14)

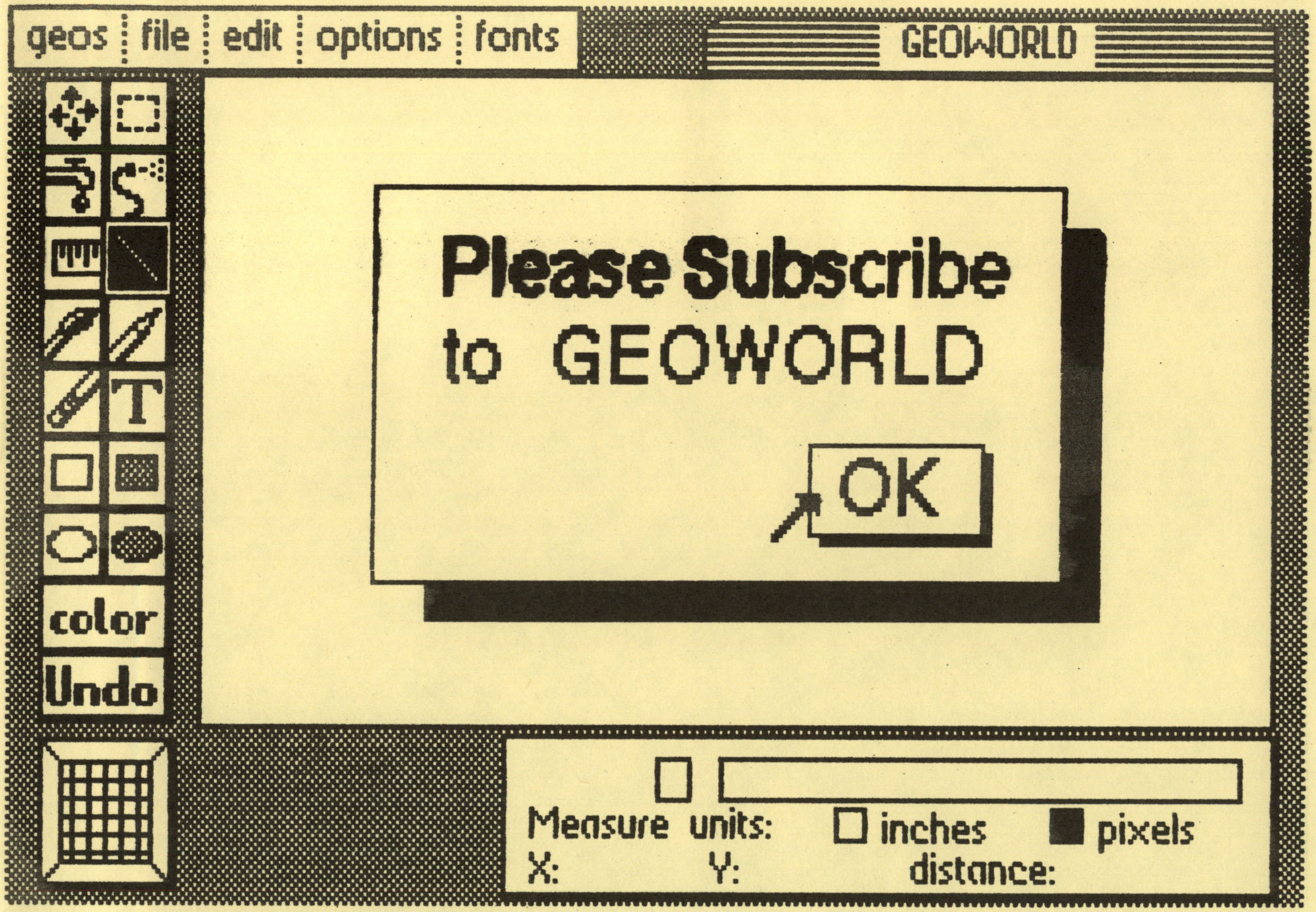
DISK#7. Graphics Convertors.
PS-GEOS. all PrintShop. (7)
PS/NewsRoom/PrintMaster
ImageCon: Hi-res/multi-GEOS
Piconvert: Doodle-GEOS (PD)
GEOS-HiRes: convertor. (7)

**DISK#8: Mystic Jim's BBS.
Run your own MJ-BBS!

Tim's Timer: Real Time Clock
Super battery-backup clock. and programs to run it.
Cost: Chip-$20, Kit-$28, Tested Clock-$37.50
Reserve a clock for $20: they are sold out fast!!
Designed and programmed by Tim Corcoran.

MYSTIC JIM ENCHANTS THE EAGLE





Please Subscribe
to GEOWORLD

Measure units: □ inches ■ pixels
X: Y: distance:

Send Check or
Money Order to:

GEOWORLD
38 Santa Ynez St.
Santa Barbara, Ca
93103

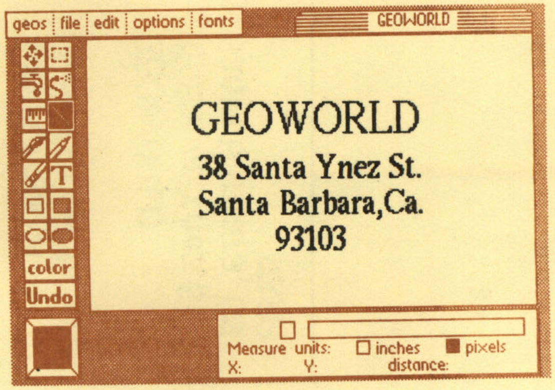
Try **GEOWORLD**
For a special charter subscription
price of $10.00 for 6 months.

Name________________________________

Street________________________________

City____________________ **State**__________

Zip Code____________



GEOWORLD
38 Santa Ynez St.
Santa Barbara,Ca.
93103