APPLEVISION HI-RES TEXT

TIPS 'N TECHNIQUES

There are many ways to *put characters on the Hi-Res screen. Some are harder than others. This program allows you to use simple PRINT statements.*

One of the first things I did after getting my new Apple computer was to investigate the programs on the system master. I ran the programs, played Little Brick Out, noted the future use of Renumber and Copy A, and then I ran Applevision. Applevision was written in Integer BASIC, which has since been replaced by the current Applesoft BASIC. It doesn't come with the new Apple //'s.

The program drew a room with a television, and a little man did a song and dance routine. After watching the little man's dance, I filed Applevision away with my cute but useless programs. That was a mistake. Applevision is far from useless. It is an example of excellent, complex programming.

There is one moment, right before the little man does his dance, when the words HOME SWEET HOME appear on the screen. Examining the program, you find the following statement:

```
PRINT "HOME SWEET HOME"
```

It seems logical, until you realize that the words appear on the Hi-Res screen! How could Applevision put letters on the Hi-Res screen with a simple PRINT statement?

After my initial exploratory efforts failed, I ran across a book called *The Apple Software Bank*, which explained some of the better Apple programs. The book flatly stated that Applevision was too complicated to explain. When someone tells me I can't understand something, it makes me want to prove that I can. I tore into Applevision.

After quite a bit of work, and a lot of disassembling, I finally understood how Applevision works. Using my newfound knowledge, I created an updated version of the machine language routines that Applevision used. The PRINTER program (**Listing 1**) and the associated character set (**Listing 2**) will work with Applesoft programs running under DOS 3.3 or ProDOS.

USING THE PROGRAM

To use the PRINTER routines in your own program, just add the following lines at the beginning:

```
10 IF PEEK(104) <> 16 THEN POKE
   104,16:POKE 103,1:POKE 4096,0:
   PRINT CHR$(4)"RUN YOUR
   PROGRAM"
20 PRINT CHR$(4)"BLOAD PRINTER":
   PRINT CHR$(4)"BLOAD PTABLE":IF
   PEEK(48896) = 76 THEN POKE
   2050,216:PRINT CHR$(4)"PR#
   A$802":GOTO 40
30 POKE 54,3:POKE 55,8
40 REM *** THE REST OF YOUR
   PROGRAM GOES HERE ***
```

Line 10 above should be modified so that the actual name of your program appears in place of YOUR PROGRAM. What it does is to check the loading location of the program, and if it is not loaded above the Hi-Res screen, it changes the necessary pointers and reruns the program. Line 20 loads the PRINTER routines and the Hi-Res character set. It then checks to see if ProDOS is active, and if so, it sets up the BASIC interpreter to use the PRINTER routines for screen display. Line 30 does the same thing for DOS 3.3.

Of course, you don't have to use these exact line numbers. Just make sure that they appear at the very beginning of your program, and you'll have no problems.

Listing 3 is a short program that demonstrates the use of the PRINTER routines. For another excellent example of how to use the PRINTER routines, see the program Word Elevator, written by my father Gary, elsewhere in this issue.

There are several POKEs that control the features of PRINTER: POKE 8,1 enables lower-case letters; POKE 8,0 returns to upper-case letters; POKE 7,1 causes numbers to be printed as subscripts; POKE 7,2 causes numbers to be printed as superscripts; and POKE 7,0 causes numbers to be printed normally. Note that actual lower-case characters should not be included in strings. Instead, use upper-case and POKE 8,1. Other than that, just use the PRINT statement normally. VTAB, HTAB and HOME all work just as you would expect, except that they function on the Hi-Res screen.

THE HI-RES/TEXT CORRESPONDENCE

The Hi-Res screen contains 280 vertical lines (numbered 0-279), and 192 horizontal lines (numbered 0-191). The text screen contains 40 vertical lines (numbered 0-39), and 24 horizontal lines (numbered 0-23). From this it can be determined that there are seven Hi-Res columns to one text column, and eight Hi-Res rows to one text row, meaning that each coordinate on the text page takes up a 7 × 8 Hi-Res plotting area. That makes calculating the corresponding Hi-Res blocks of a text coordinate very simple.

After quite a bit of work, and a lot of disassembling, I finally understood how Applevision works.

Now the correct data must be put into that area. Each Hi-Res byte contains information for seven pixels. That means that eight bytes will be required for each character — one for each of the eight rows of the seven-pixel wide characters. The characters are arranged in the data table according to ascending ASCII code. The first byte is always zero so that the top row will remain blank, creating a space between the rows of characters.

LISTING 1: PRINTER

```
                        1   ********************************
                        2   *PRINTER                       *
                        3   *BY KEVIN KNOX                 *
                        4   *COPYRIGHT 1986 BY             *
                        5   *MICROSPARC, INC.              *
                        6   *CONCORD, MA  01742            *
                        7   *                              *
                        8   *MERLIN ASSEMBLER             *
                        9   ********************************
                       10   COUT1    =      $FDF0        ;CHARACTER OUTPUT ROUTINE
                       11   ********************************
                       12            ORG    $803
                       13   *
0803: C9 8D            14            CMP    #$8D         ;END OF LINE?
0805: D0 03            15            BNE    TRANSF       ;NO, PUT IT ON THE SCREEN
0807: 4C AB 08         16            JMP    PRRTN        ;RETURN TO CALLER
080A: 85 FD            17   TRANSF   STA    $FD          ;STORE CHARACTER IN ZERO PAGE
080C: 48               18            PHA                 ;AND ON THE STACK
080D: 98               19            TYA
080E: 48               20            PHA                 ;SAVE Y-REG ON STACK
080F: 8A               21            TXA
0810: 48               22            PHA                 ;SAVE X-REG ON STACK
0811: A5 FD            23            LDA    $FD          ;RETRIEVE CHARACTER
0813: C9 C1            24            CMP    #$C1
0815: 30 0F            25            BMI    CHECK        ;NOT A LETTER, CHECK NUMBERS
0817: C9 DB            26            CMP    #$DB
0819: 10 0B            27            BPL    CHECK        ;NOT A LETTER, CHECK NUMBERS
081B: A4 08            28            LDY    $08          ;IS LOWERCASE ENABLED?
081D: C0 01            29            CPY    #$01
081F: D0 21            30            BNE    CHCK2        ;NO
0821: A2 00            31            LDX    #$00         ;NO OFFSET
0823: 4C 44 08         32            JMP    LOFFS        ;GO PRINT IT
0826: C9 B0            33   CHECK    CMP    #$B0         ;IS IT A NUMBER?
0828: 30 18            34            BMI    CHCK2        ;NO
082A: C9 BA            35            CMP    #$BA         ;A NUMBER?
082C: 10 14            36            BPL    CHCK2        ;NO
082E: A4 07            37            LDY    $07          ;IS SCRIPTING ENABLED?
0830: C0 01            38            CPY    #$01
0832: F0 09            39            BEQ    CHCK1        ;YES, SUBSCRIPTS
0834: C0 02            40            CPY    #$02
0836: D0 0A            41            BNE    CHCK2        ;YES, SUPERSCRIPTS
0838: A2 02            42            LDX    #$02         ;OFFSET FOR NUMBERS
083A: 4C 44 08         43            JMP    LOFFS        ;GO PRINT IT
083D: A2 04            44   CHCK1    LDX    #$04         ;OFFSET FOR SUBSCRIPTS
083F: 4C 44 08         45            JMP    LOFFS        ;GO PRINT IT
0842: A2 06            46   CHCK2    LDX    #$06         ;OFFSET FOR SUPERSCRIPTS
0844: BD FB 0B         47   LOFFS    LDA    TDATA,X      ;GET ADDRESS OF CHARACTER TABLE
0847: 8D 63 08         48            STA    ADD1+1       ;AND SET UP DRAWING ROUTINE
084A: BD FC 0B         49            LDA    TDATA+1,X
084D: 8D 69 08         50            STA    ADD2+1
0850: A5 FD            51            LDA    $FD          ;RETRIEVE THE CHARACTER
0852: 29 3F            52            AND    #$3F         ;STRIP OFF BITS 7 AND 8
0854: 0A               53            ASL                 ;AND SHIFT LEFT THREE TIMES
0855: 0A               54            ASL                 ;TO CREATE INDEX TO PROPER
0856: 0A               55            ASL                 ;CHARACTER
0857: 85 19            56            STA    $19          ;STORE INDEX
0859: A9 00            57            LDA    #$00
085B: 69 00            58            ADC    #$00
085D: 85 1A            59            STA    $1A          ;STARTING ADDRESS
085F: A5 19            60            LDA    $19
0861: 18               61            CLC
0862: 69 D8            62   ADD1     ADC    #$D8         ;THIS OFFSET GETS CHANGED BY LINE 48
0864: 85 19            63            STA    $19
0866: A5 1A            64            LDA    $1A          ;THIS OFFSET GETS CHANGED BY LINE 50
0868: 69 08            65   ADD2     ADC    #$08
086A: 85 1A            66            STA    $1A
086C: A5 25            67            LDA    $25          ;GET VERTICAL CURSOR POSITION
086E: 0A               68            ASL
086F: 0A               69            ASL
0870: 0A               70            ASL                 ;SHIFT LEFT THREE TIMES
0871: 85 1B            71            STA    $1B          ;AND STORE IN ZERO PAGE
0873: A2 08            72            LDX    #$08
0875: A0 00            73            LDY    #$00
0877: 20 AE 08         74   SETHR    JSR    ADJMEM       ;POINT TO HIRES PAGE 2
087A: 18               75            CLC
087B: A5 1C            76            LDA    $1C
087D: 65 24            77            ADC    $24          ;GET CURSOR HORIZONTAL POSITION
```

ENTERING THE PROGRAMS

If you don't have an assembler, enter the Monitor with CALL −151, type in the hexadecimal code from Listing 1 and save it with the command:

BSAVE PRINTER,A$803,L$D5

If you have an assembler, type the entire source code and assemble it instead. Save the object file using the name PRINTER. Next, enter the Monitor with CALL −151, type in the hexadecimal code in Listing 2 and save it with:

BSAVE PTABLE,A$8D8,L$32B

Finally, type in the Applesoft program in Listing 3 and save it with:

SAVE PRINTER.DEMO

For help with entering *Nibble* programs, see "A Welcome to New *Nibble* Readers" at the beginning of this issue.

FEATURES AND LIMITATIONS

PRINTER can utilize every Applesoft command used to control text output except for the FLASH command. Three separate tables can be accessed: the normal table, which contains all possible text characters; the lower-case table, which contains lower-case alphabetic characters; and the number scripting table, which contains numbers for subscripting and superscripting.

If a special table is activated and you try to PRINT a character not in that table, the program will default to the normal table to print the character without deactivating the special table. Actual lower-case characters should not be included in PRINT statements.

PRINTER can detect whether Hi-Res page 1 or 2 is being used for printing.

When PRINTER is engaged, DOS 3.3 is disconnected. In order to reconnect DOS and disconnect the program, type:

POKE 54,189:POKE 55,158

Reconnect the program with:

POKE 54,3:POKE 55,8

This is only necessary if you are going to do some file access, and need to have DOS properly connected. ProDOS works differently, and it is not necessary to disconnect PRINTER before file access.

KEY PERFECT 5.0
RUN ON
PRINTER

```
CODE-5.0    ADDR# - ADDR#    CODE-4.0
--------    -------------    --------
8430C7E0     0803 - 0852       2C6E
7B11AF72     0853 - 08A2       2DDE
3C9947E7     08A3 - 08D7       1A6E
AA6098E7 = PROGRAM TOTAL =       D5
```

LISTING 1: PRINTER *(continued)*

```
087F: 85 1C     78             STA   $1C        ;STORE IN ZERO PAGE
0881: A5 1D     79             LDA   $1D
0883: 69 00     80             ADC   #$00
0885: 85 1D     81             STA   $1D
0887: A5 E6     82             LDA   $E6        ;ADDRESS OF HIRES PAGE (HI BYTE)
0889: 38        83             SEC
088A: E9 20     84             SBC   #$20
088C: 18        85             CLC
088D: 65 1D     86             ADC   $1D        ;CALCULATE ADDRESS FOR DRAWING
088F: 85 1D     87             STA   $1D
0891: B1 19     88             LDA   ($19),Y    ;LOAD SHAPE DATA FOR CHARACTER
0893: A4 32     89             LDY   $32
0895: C0 3F     90             CPY   #$3F       ;DOES IT NEED TO BE INVERSE?
0897: D0 02     91             BNE   STORE      ;NO, DRAW IT AS IS
0899: 49 7F     92             EOR   #$7F       ;MAKE IT INVERSE
089B: A0 00     93  STORE      LDY   #$00
089D: 91 1C     94             STA   ($1C),Y    ;STORE DATA ON SCREEN
089F: E6 1B     95             INC   $1B
08A1: E6 19     96             INC   $19
08A3: CA        97             DEX
08A4: D0 D1     98             BNE   SETHR      ;UNTIL WHOLE CHARACTER IS DONE
08A6: 68        99             PLA              ;RESTORE REGISTERS
08A7: AA        100            TAX
08A8: 68        101            PLA
08A9: A8        102            TAY
08AA: 68        103            PLA
08AB: 4C F0 FD  104 PRRTN      JMP   COUT1      ;AND EXIT THROUGH CHARACTER OUTPUT
08AE: A5 1B     105 ADJMEM     LDA   $1B        ;CALCULATE HI RES ADDRESSES
08B0: 0A        106            ASL
08B1: 0A        107            ASL
08B2: 29 1C     108            AND   #$1C
08B4: 85 1D     109            STA   $1D
08B6: A5 1B     110            LDA   $1B
08B8: 6A        111            ROR
08B9: 6A        112            ROR
08BA: 6A        113            ROR
08BB: 6A        114            ROR
08BC: 29 03     115            AND   #$03
08BE: 05 1D     116            ORA   $1D
08C0: 09 20     117            ORA   #$20
08C2: 85 1D     118            STA   $1D
08C4: A5 1B     119            LDA   $1B
08C6: 6A        120            ROR
08C7: 29 E0     121            AND   #$E0
08C9: 85 1C     122            STA   $1C
08CB: 6A        123            ROR
08CC: 6A        124            ROR
08CD: 29 18     125            AND   #$18
08CF: 05 1C     126            ORA   $1C
08D1: 85 1C     127            STA   $1C
08D3: 60        128            RTS
08D4: EA        129            NOP              ;USE NOPS TO MAKE SOME
08D5: EA        130            NOP              ;ROOM, SO TABLE CAN BE
08D6: EA        131            NOP              ;LOADED IN CORRECT LOCATION
08D7: EA        132            NOP
                133 TABLE      =     *          ;CHARACTER SHAPE TABLE GOES HERE
                134 TDATA      =     TABLE+$323 ;OFFSET DATA LOCATION

--End assembly, 213 bytes, Errors: 0
END OF LISTING 1
```

LISTING 2: PTABLE

```
08D8- 00 1C 22 2A 3A 1A 02 3C
08E0- 00 08 14 22 22 3E 22 22
08E8- 00 1E 22 22 1E 22 22 1E
08F0- 00 1C 22 02 02 02 22 1C
08F8- 00 1E 22 22 22 22 22 1E
0900- 00 3E 02 02 1E 02 02 3E
0908- 00 3E 02 02 1E 02 02 02
0910- 00 3C 02 02 02 32 22 3C
0918- 00 22 22 22 3E 22 22 22
0920- 00 1C 08 08 08 08 08 1C
0928- 00 20 20 20 20 20 22 1C
0930- 00 22 12 0A 06 0A 12 22
0938- 00 02 02 02 02 02 02 3E
0940- 00 22 36 2A 2A 22 22 22
0948- 00 22 22 26 2A 32 22 22
0950- 00 1C 22 22 22 22 22 1C
0958- 00 1E 22 22 1E 02 02 02
0960- 00 1C 22 22 22 2A 12 2C
0968- 00 1E 22 22 1E 0A 12 22
0970- 00 1C 22 02 1C 20 22 1C
0978- 00 3E 08 08 08 08 08 08
0980- 00 22 22 22 22 22 22 1C
0988- 00 22 22 22 22 22 14 08
0990- 00 22 22 22 22 2A 2A 36 22
0998- 00 22 22 14 08 14 22 22
09A0- 00 22 22 14 08 08 08 08
09A8- 00 3E 20 10 08 04 02 3E
09B0- 00 3E 06 06 06 06 06 3E
09B8- 00 00 02 04 08 10 20 00
09C0- 00 3E 30 30 30 30 30 3E
09C8- 00 00 00 08 14 22 00 00
09D0- 00 00 00 00 00 00 00 3E
09D8- 00 00 00 00 00 00 00 00
09E0- 00 08 08 08 08 08 00 08
09E8- 00 14 14 14 00 00 00 00
09F0- 00 14 14 3E 14 3E 14 14
09F8- 00 08 3C 0A 1C 28 1E 08
0A00- 00 06 26 10 08 04 32 30
0A08- 00 04 0A 0A 04 2A 12 2C
0A10- 00 08 08 08 00 00 00 00
0A18- 00 08 04 02 02 02 04 08
0A20- 00 08 10 20 20 20 10 08
0A28- 00 08 2A 1C 08 1C 2A 08
```

```
0A30- 00 00 08 08 3E 08 08 00
0A38- 00 00 00 00 00 08 08 04
0A40- 00 00 00 00 3E 00 00 00
0A48- 00 00 00 00 00 00 00 08
0A50- 00 00 20 10 08 04 02 00
0A58- 00 1C 22 32 2A 26 22 1C
0A60- 00 08 0C 08 08 08 08 1C
0A68- 00 1C 22 20 18 04 02 3E
0A70- 00 3E 20 10 18 20 22 1C
0A78- 00 10 18 14 12 3E 10 10
0A80- 00 3E 02 1E 20 20 22 1C
0A88- 00 38 04 02 1E 22 22 1C
0A90- 00 3E 20 10 08 04 04 04
0A98- 00 1C 22 22 1C 22 22 1C
0AA0- 00 1C 22 22 3C 20 10 0E
0AA8- 00 00 00 08 00 08 00 00
0AB0- 00 00 00 08 00 08 08 04
0AB8- 00 10 08 04 02 04 08 10
0AC0- 00 00 00 3E 00 3E 00 00
0AC8- 00 04 08 10 20 10 08 04
0AD0- 00 1C 22 10 08 08 00 08
0AD8- 00 00 00 0E 10 1E 11 2E
0AE0- 00 01 01 0F 11 11 11 0F
0AE8- 00 00 00 1E 01 01 01 1E
0AF0- 00 10 10 1E 11 11 11 1E
0AF8- 00 00 00 1E 21 3F 01 1E
0B00- 00 0C 02 02 0F 02 02 02
0B08- 00 00 1E 11 11 1E 10 0E
0B10- 00 01 01 0F 11 11 11 11
0B18- 00 02 00 03 02 02 02 07
0B20- 00 08 00 0C 08 08 08 07
0B28- 00 01 01 09 05 03 05 09
0B30- 00 03 02 02 02 02 02 07
0B38- 00 00 00 1E 25 25 25 25
0B40- 00 00 00 0F 11 11 11 11
0B48- 00 00 00 0E 11 11 11 0E
0B50- 00 00 00 07 09 07 01 01
0B58- 00 00 00 0E 09 0E 08 08
0B60- 00 00 00 1D 03 01 01 01
0B68- 00 00 00 1E 01 1E 20 1E
0B70- 00 02 02 0F 02 02 02 1C
0B78- 00 00 00 11 11 11 11 2E
0B80- 00 00 00 11 11 11 0A 04
0B88- 00 00 00 21 25 25 2D 12
0B90- 00 00 00 11 0A 04 0A 11
0B98- 00 00 00 11 0A 04 02 01
0BA0- 00 00 00 1F 08 04 02 1F
0BA8- 00 00 00 1C 14 14 14 1C
0BB0- 00 00 00 08 08 08 08 08
0BB8- 00 00 00 1C 10 1C 04 1C
0BC0- 00 00 00 1C 10 1C 10 1C
0BC8- 00 00 00 14 14 1C 10 10
0BD0- 00 00 00 1C 04 1C 10 1C
0BD8- 00 00 00 1C 04 1C 14 1C
0BE0- 00 00 00 1C 10 10 10 10
0BE8- 00 00 00 1C 14 1C 14 1C
0BF0- 00 00 00 1C 14 1C 10 10
0BF8- 00 00 00 D0 0A 2B 0A 28
0C00- 0A D8 08
```

END OF LISTING 2

```
        KEY PERFECT 5.0
            RUN ON
            PTABLE

====================================
CODE-5.0    ADDR# - ADDR#    CODE-4.0
--------    -------------    --------
6D2890C7      08D8 - 0927        2AFE
BE8F9E5B      0928 - 0977        2423
550BC6D3      0978 - 09C7        2C7D
4344CDE4      09C8 - 0A17        269F
D354BDF5      0A18 - 0A67        24BC
0DD87596      0A68 - 0AB7        22C8
E3527CF2      0AB8 - 0B07        2354
4B6EE32C      0B08 - 0B57        26CE
D2EC21DA      0B58 - 0BA7        28DE
274FF11D      0BA8 - 0BF7        1BB2
374FE53E      0BF8 - 0C02        034A
31DE8B75 = PROGRAM TOTAL =       032B
```

LISTING 3: PRINTER.DEMO

```
10  REM  ************************
20  REM  * PRINTER.DEMO         *
30  REM  * BY KEVIN KNOX        *
40  REM  *                      *
50  REM  * COPYRIGHT 1986 BY    *
60  REM  * MICROSPARC, INC.     *
70  REM  * CONCORD, MA  01742   *
80  REM  ************************
90  IF  PEEK (104) <  > 16 THEN  POKE 104,16:
      POKE 103,1: POKE 4096,0: PRINT  CHR$ (4
    )"RUN PRINTER.DEMO"
100 PRINT  CHR$ (4)"BLOAD PRINTER": PRINT  CHR$
    (4)"BLOAD PTABLE": IF  PEEK (48896) = 76
     THEN  POKE 2050,216: PRINT  CHR$ (4)"PR
    # A$802": GOTO 120
110 POKE 54,3: POKE 55,8
120 HGR : HOME
130 PRINT "THE PRINTER ROUTINES WILL---"
140 VTAB 3: PRINT "PRINT IN CAPITALS"
150 POKE 8,1: PRINT "PRINT IN LOWERCASE"
160 POKE 8,0: INVERSE : VTAB 7: PRINT " PRIN
    T IN INVERSE": NORMAL
170 POKE 7,1: VTAB 9: PRINT "PRINT NUMBERS A
    S SUBSCRIPTS -->A5"
180 POKE 7,2: VTAB 11: PRINT "PRINT NUMBERS
    AS SUPERSCRIPTS -->A6"
190 POKE 7,0: VTAB 13: PRINT "USE CHR$ --> C
    HR$(91) = "; CHR$ (91)
200 P$ = "A CHARACTER STRING":V = 10
210 VTAB 15: PRINT "AND PRINT VARIABLES:"
220 VTAB 17: PRINT "P$ = ";P$: PRINT "AND V
    = "V
230 PRINT
240 PRINT "PRESS <RETURN> FOR TEXT SCREEN"::
     GET Z$: PRINT : TEXT
```

END OF LISTING 3