AMPOS

Use this quick and easy routine to place the cursor on both the 40- and 80-column screens, and fix an 80-column bug while you're at it!

This short program can change forever the arcane way you specify cursor positions on the screen. With Ampos, a three-parameter Applesoft ampersand command takes the place of the HTAB, VTAB, and PRINT commands, enabling you to program cursor movement speedily and with a minimum of fuss. As an added bonus, it also fixes that pesky 40-column wraparound problem that plagues some 80-column IIe cards.

USING THE PROGRAM

The syntax of Ampos is simple to use:

```
&CV,CH,PRINT
```

where CV is any legal VTAB value, from 1-24; CH is any HTAB value from 1-80; and PRINT is any legal PRINT parameter. You can use normal variables, literals, and functions with all three parameters, provided they are of the proper Applesoft syntax.

You can also skip a parameter. If you want to print "ERNIE" at HTAB 10, then you would be able to skip entering the VTAB parameter by typing:

```
&,10,"ERNIE"
```

To install Ampos, type BRUN AMPOS. You may do this in immediate mode, or in your Applesoft programs with the statement:

```
PRINT CHR$(4)"BRUN AMPOS"
```

Limitations

When you are using Ampos, keep four things in mind:

1. The order of the parameters must be CV,CH,PRINT. If you specify a different order (e.g., CH,CV,PRINT or PRINT,CH,CV), then you may generate an error.
2. If you decide to skip a parameter, you still must include a comma to let the program know which expressions go with which parameter. If you wish to print "ERNIE" at the current cursor position, the proper syntax is &,,"ERNIE", not &"ERNIE". Or if you wish to position the cursor at HTAB 10, then the proper syntax is &,10.
3. Every Ampos call must have at least one comma, no matter how many parameters you are using. If you want to position the cursor at VTAB 10, the proper syntax for Ampos would be & 10, and not & 10. Even though there is just one parameter, you must include a comma.
4. Ampos will not work with the Videx and compatible 80-column cards on the Apple II Plus.

AMPOS.DEMO demonstrates how Ampos might be used in your programs. **Lines 100-120** print the opening message and test your machine for 80-column capability. **Line 140** then installs Ampos. After waiting for a keypress, **line 160** shuts off your 80-column display, if it's on. The program then performs a demo, first in 40 columns (**lines 180 and 190**) and then in 80 columns (**lines 200 and 210**). The 80-column demo will be executed only if you have 80-column capability. Both the 40- and 80-column demos use a subroutine at **lines 230-360**, where the new ampersand command is used.

ENTERING THE PROGRAM

To enter Ampos, you may either use an assembler or enter the hex code directly via the monitor. If you use the monitor, enter the hex code from **Listing 1** and save it with the command:

BSAVE AMPOS,A$300,L$49

If you have an assembler, enter the source code from **Listing 2** and assemble it, saving the object code in a file called AMPOS. Next, enter the Applesoft demo program (**Listing 3**) and save it with the command:

SAVE AMPOS.DEMO

For help in entering the listings, see the Typing Tips section of this magazine.

HOW IT WORKS

The machine language portion of Ampos doesn't actually begin in **Listing 1** until **line 41**, where the ampersand hook is installed at locations $3F6-$3F7. When an ampersand is encountered during the execution of an Applesoft program, control passes to **line 47**. The accumulator is loaded with the current character pointed to by TXTPTR ($B8), and the accumulator is checked for a comma. If a comma is found, then the VTAB parameter has been skipped and control passes to **line 52**. If there is no comma, then the Applesoft HTAB routine will attempt to interpret the parameter as a VTAB (**line 50**).

Lines 52-54 increment TXTPTR past the comma delimiter separating the VTAB and HTAB parameters and get the next character. Again, a parameter skip is checked, this time by looking for a zero (**line 55**), a comma (**line 56**), or a full colon (**line 58**). If none of these is found, then Ampos attempts to evaluate the expression as an HTAB and checks it for legal range (0-79) in **lines 62-65**. If the HTAB is not in range, then **lines 67-68** print an ILLEGAL QUANTITY ERROR message and halt the program. **Lines 71-72** update main memory and the 80-column cards, horizontal cursor position, CH.

Lines 73-78 interpret the PRINT parameter. The current character is loaded and checked for a zero (**line 74**) or a full colon (**line 75**). If neither is found, then the Applesoft PRINT routine will try to evaluate the expression (**line 78**), followed by a return to Applesoft at **line 80**.

LISTING 1: AMPOS

```
START:300      LENGTH:49

C2 0300:A9 0B 8D F6 03 A9 03 8D
27 0308:F7 03 60 20 B7 00 C9 2C
6A 0310:F0 03 20 56 F2 20 B1 00
D1 0318:20 B7 00 F0 2B C9 2C F0
4E 0320:18 C9 3A F0 23 20 F8 E6
9C 0328:CA 30 04 E0 50 90 05 A2
C9 0330:35 4C 12 D4 86 24 8E 7B
4E 0338:05 20 B7 00 F0 0A C9 3A
D4 0340:F0 06 20 B1 00 20 D5 DA
48 0348:60

TOTAL: D00F
```

END OF LISTING 1

```
              KEY PERFECT 5.0
                  RUN ON
                  AMPOS
=====================================
  CODE-5.0    ADDR# - ADDR#   CODE-4.0
  --------    -------------   --------
  7DF1A91A     0300 - 0348      2199
  7DF1A91A = PROGRAM TOTAL =      49
```

LISTING 2: AMPOS.S

```
 1            ORG  $300
 2            OBJ  $300
 3 *
 4 *
 5 *
 6 *
 7 *************************
 8 *    AMPOS.S            *
 9 *    BY ED DOXTATOR     *
10 *    COPYRIGHT (C) 1988 *
11 *    MICROSPARC, INC    *
12 *    CONCORD, MA 01742  *
13 *    APPLESOFT TOOLKIT  *
14 *      ASSEMBLER        *
15 *************************
16 *
17 *               ZPAGE          [APPLESOFT]
18 *
19 CH        EQU  $24          ;MAIN MEM HORIZ. POS
20 CV        EQU  $25          ;VERTICAL POS
21 CHRGET    EQU  $B1          ;INC TXTPTR, GET CHAR
22 CHRGOT    EQU  $B7          ;GET A CHAR, DON'T INC TXTPTR
23 *
24 *          PAGE THREE
25 *
26 AMPERV    EQU  $3F5         ;AMPERSAND HOOK
27 *
28 *          CARD SCRATCHPAD
29 *
30 OURCH     EQU  $57B         ;"HTAB" FOR 80-COLUMN CARD
31 *
32 *          ROM  ROUTINES
33 *
34 GETBYT    EQU  $E6F8        ;GET A BYTE IN X-REG
35 ERROR     EQU  $D412        ;HANDLE ERROR IN X-REG
36 PRINT     EQU  $DAD5        ;APPLESOFT PRINT ROUTINE
37 CHKCOM    EQU  $DEBE        ;PARSE COMMA
38 VTAB      EQU  $F256        ;INTERP VTAB PARM
39 *
40 *          START THIS       MESS HERE...
41 *
42           MSB  OFF
43 AMPER     LDA  #>START      ;SET UP AMPERSAND HOOK
44           STA  AMPERV+1
45           LDA  #<START
46           STA  AMPERV+2
47           RTS               ;OUT TO DOS/BASIC
48 *
49 START     JSR  CHRGOT       ;CHECK FOR PARM SKIP
50           CMP  #',          ;COMMA?
51           BEQ  SKIPCV       ;YES, KEEP CURRENT CV
52           JSR  VTAB
53 *
54 SKIPCV    JSR  CHRGET
55 *
56 DOCH      JSR  CHRGOT
57           BEQ  BYE          ;END OF LINE
58           CMP  #',
59           BEQ  MYPRINT      ;PARM SKIP, HEAD FOR PRINT
60           CMP  #':
61           BEQ  BYE          ;END OF STATEMENT
62 GETCH     JSR  GETBYT
63           DEX
64           BMI  ERR
65           CPX  #80
66           BCC  OKSTORE      ;IN RANGE, STORE IT.
67 *
68 ERR       LDX  #53          ;ERROR, ILLEGAL QUANTITY
69           JMP  ERROR
70 *
71 OKSTORE   STX  CH           ;FIX COLUMN WIDTH
72           STX  OURCH
73 *
74 MYPRINT   JSR  CHRGOT       ;PRINT ANYTHING?
75           BEQ  BYE          ;ZERO, EOL
76           CMP  #':
77           BEQ  BYE          ;NEXT STATEMENT
78           JSR  CHRGET       ;GOBBLE COMMA
79           JSR  PRINT        ;AND PRINT STRING
80 *
81 BYE       RTS               ;EXIT
82 END       EQU  *
83 LENGTH    EQU  END-AMPER
```

END OF LISTING 2

LISTING 3: AMPOS.DEMO

```
37  10  REM ***********************
C0  20  REM * AMPOS.DEMO          *
B9  30  REM * BY ED DOXTATOR      *
AE  40  REM * COPYRIGHT (C) 1988  *
CB  50  REM * BY MICROSPARC, INC. *
24  60  REM * CONCORD, MA   01742 *
45  70  REM ***********************
3A  80  REM
3C  90  HOME
56  100  VTAB 12: PRINT "AMPOS.DEMO BY ED DOXTATOR"
        : PRINT "COPYRIGHT (C) 1988 BY MICROSPARC,
        INC."
10  110 EIGHTY = 1: IF  PEEK (64435) <  > 6 THEN EI
        GHTY = 0
02  120  POKE 49153,0: POKE 49237,0: POKE 1024,123:
        A =  PEEK (1024): POKE 49236,0: POKE 49152,
        0: IF A <  > 123 THEN EIGHTY = 0
50  130  ONERR  GOTO 400
CE  140  PRINT  CHR$ (4);"BRUN AMPOS": POKE 216,0
9B  150  GOSUB 380: HOME
61  160  PRINT  CHR$ (27); CHR$ (17)
1C  170 A$ = "PLACING TEXT IS":B$ = "EASY WITH AMPO
        S":C$ = "                ": REM  16 SPACES
EB  180  & 1,9,"40 COLUMN AMPOS DEMO"
14  190 YT = 20: GOSUB 230
CC  200  IF EIGHTY THEN  PRINT  CHR$ (4)"PR#3":YT =
        60: & 1,9,"80 COLUMN AMPOS DEMO": GOSUB 23
        0
BC  210  IF  NOT EIGHTY THEN  PRINT "CANNOT BE USED
        WITH THIS MACHINE."
F7  220  PRINT  CHR$ (27); CHR$ (17): HOME : END
36  230  & 2,1,"-----------------------------
        ----": & 22,1,"--------------------------
        -----------": REM  39 -
F4  240  FOR X = 4 TO 20
2C  250  & X + 1,1,B$: & X,1,A$
FD  260  FOR T = 1 TO 100: NEXT T: REM  DELAY
DF  270  & X,1,C$: NEXT X
81  280  FOR X = 1 TO YT
7E  290  & 20,X,A$: & 21,X,B$
F6  300  FOR T = 1 TO 100: NEXT T: REM  DELAY
FA  310  IF X < YT THEN  & 20,X," ": & 21,X," "
F6  320  NEXT X
99  330  IF YT = 60 THEN  & 2,40,"------------------
        ----------------------": & 22,40,"---------
        ------------------------------": REM  39 -
3C  340  IF YT = 60 THEN  & 1,9,"
          ": & 1,29,"80 COLUMN AMPOS DEMO": REM   2
        0 SPACES
2C  350  GOSUB 370
B2  360  RETURN
45  370  IF YT = 60 THEN  & 23,28,"PRESS RETURN TO
        CONTINUE": GOTO 390
26  380  & 23,8,"PRESS RETURN TO CONTINUE"
9F  390  POKE 49168,0: WAIT  - 16384,128: RETURN
18  400  POKE 216,0: HOME : VTAB 12: PRINT "UNABLE
        TO LOAD AMPOS.": END

TOTAL: 3BFA
```

END OF LISTING 3

```
              KEY PERFECT 5.0
                  RUN ON
                AMPOS.DEMO
=====================================
  CODE-5.0    LINE# - LINE#   CODE-4.0
  --------    -------------   --------
  B78A7BB8      10 -  100       7032
  875BACD4     110 -  200       97ED
  AAA7F6D2     210 -  300       815A
  4BD800E4     310 -  400       A2D6
  D9E967AB = PROGRAM TOTAL =    04C6
```