PAiA INTERACTIVE EDITOR DEBUGGER (PIEBUG)

Monitor Listing

```
0200          ;
0200          ;
0200          ;
0200          ;
0200          ;
0200          ;
0200          ;     **************************************
0200          ;     *                                    *
0200          ;     *   PIEBUG             VERSION 1.0   *
0200          ;     *   PAIA INTERACTIVE EDITOR-DEBUGGER *
0200          ;     *   WRITTEN BY ROGER WALTON          *
0200          ;     *   COPYRIGHT 1977 BY PAIA           *
0200          ;     *      ELECTRONICS, INC.             *
0200          ;     *                                    *
0200          ;     **************************************
0200          ;
0200          ;
0200          ;
0200                   *=$0F00
0F00          ;
0F00          KEY      =$0800          ;BASE ADDR OF KEY PORTS
0F00          TEMP     =$EE            ;TEMPORARY STORAGE
0F00          LASTKE   =$F8            ;PREVIOUS KEY DECODED
0F00          BUFFER   =$F0            ;KEY ENTRY BUFFER
0F00          DISP     =$0820          ;LED DISPLAY
0F00          MSTACK   =$ED            ;MONITOR STACK POINTER
0F00          PNTER    =$F6            ;16 BIT ADDR POINTER
0F00          TAPE1    =$0E00          ;START OF TAPE SYSTEM
0F00          CASS     =$0900          ;CASSETTE PORT
0F00          ;
0F00          ;
0F00          ACC      =$F9            ;REG STORAGE
0F00          YREG     =$FA            ;      "
0F00          XREG     =$FB            ;      "
0F00          PC       =$FC            ;      "
0F00          STACKP   =$FE            ;      "
0F00          PREG     =$FF            ;REG STORAGE
0F00          ;
0F00          ;
```

```
0490  0F00           ;       DECODE KEY SUBROUTINE
0500  0F00           ;       THIS SUB SCANS THE ENTIRE KEYBOARD AND
0510  0F00           ;       RETURNS WITH DECODED KEY VALUE IN A AND Y.
0520  0F00           ;       CARRY IS CLEAR IF NEW KEY.  X IS
0530  0F00           ;       DESTROYED.  $18 IS "NO KEY" CODE.
0540  0F00           ;
0550  0F00  A0 00    DECODE LDY #0           ;CLEAR RESULT REG
0560  0F02  A2 21           LDX #$21         ;X IS PORT REG
0570  0F04  A9 01    LOOP   LDA #1
0580  0F06  85 EE           STA TEMP         ;SET UP MASK
0590  0F08  BD 00 08 NEXT   LDA KEY,X        ;READ CURRENT KEY PORT
0600  0F0B  25 EE           AND TEMP         ;USE MASK TO SELECT KEY
0610  0F0D  D0 0A           BNE RESULT       ;BRANCH IF KEY DOWN
0620  0F0F  C8              INY              ;SET RESULT TO NEXT KEY
0630  0F10  06 EE           ASL TEMP         ;SHIFT MASK TO NEXT KEY
0640  0F12  90 F4           BCC NEXT         ;BR IF MORE KEYS ON PORT
0650  0F14  8A              TXA
0660  0F15  0A              ASL A            ;SELECT NEXT PORT
0670  0F16  AA              TAX
0680  0F17  90 EB           BCC LOOP         ;BRANCH IF NOT LAST PORT
0690  0F19  C4 F8    RESULT CPY LASTKE       ;CLEAR CARRY IF NEW KEY
0700  0F1B  84 F8           STY LASTKE       ;UPDATE LASTKEY
0710  0F1D  98              TYA              ;MOVE KEY TO ACC
0720  0F1E  60              RTS              ;RETURN
0730  0F1F           ;
0740  0F1F           ;
0750  0F1F           ;
0760  0F1F           ;       GETKEY SUBROUTINE
0770  0F1F           ;       THIS SUB WAITS FOR A NEW KEY TO BE
0780  0F1F           ;       TOUCHED AND THEN RETURNS WITH THE
0790  0F1F           ;       KEY VALUE IN THE ACCUMULATOR.
0800  0F1F           ;       X AND Y ARE CLEARED.
0810  0F1F           ;
0820  0F1F           ;       BEEP SUBROUTINE (EMBEDDED IN GETKEY SUB)
0830  0F1F           ;       THIS SUB PRODUCES A SHORT BEEP AT
0840  0F1F           ;       THE CASSETTE PORT.  CARRY MUST BE
0850  0F1F           ;       CLEAR BEFORE ENTERING.  X AND Y
0860  0F1F           ;       ARE CLEARED.
0870  0F1F           ;
0880  0F1F  20 00 0F GETKEY JSR DECODE       ;GET A KEY
0890  0F22  A2 14    BEEP   LDX #20          ;ENTER HERE FOR BEEP SUB
0900  0F24  A0 3F    NXTX   LDY #$3F
0910  0F26  B0 03    DELAY  BCS DLY          ;SKIP TONE IF CARRY SET
0920  0F28  8C 00 09        STY CASS         ;GENERATE TONE
0930  0F2B  88       DLY    DEY              ;DELAY
0940  0F2C  D0 F8           BNE DELAY
0950  0F2E  CA              DEX              ;DELAY SOME MORE
0960  0F2F  D0 F3           BNE NXTX         ;NEXT X
0970  0F31  B0 EC           BCS GETKEY       ;BRANCH IF NOT NEW KEY
0980  0F33  60              RTS              ;RETURN
0990  0F34           ;
1000  0F34           ;
1010  0F34           ;
1020  0F34           ;
```

```
1030  0F34             ;      SHIFT BUFFER SUBROUTINE
1040  0F34             ;      THIS SUB SHIFTS THE LOWER 4 BITS OF
1050  0F34             ;      THE ACCUMULATOR INTO THE LEAST
1060  0F34             ;      SIGNIFICANT POSITION OF BUFFER.  THE
1070  0F34             ;      ENTIRE BUFFER IS SHIFTED 4 TIMES AND
1080  0F34             ;      THE MOST SIGNIFICANT 4 BITS ARE LOST.
1090  0F34             ;      X AND Y ARE CLEARED.  IF ON RETURN,
1100  0F34             ;      A SINGLE "ROL A" IS PERFORMED,
1110  0F34             ;      THE LOWER 4 BITS OF THE ACCUMULATOR
1120  0F34             ;      WILL CONTAIN THE 4 BITS THAT WERE
1130  0F34             ;      SHIFTED OUT OF BUFFER.
1140  0F34             ;
1150  0F34  0A         SHIFT  ASL A           ;SHIFT KEY INFORMATION
1160  0F35  0A                ASL A           ;TO UPPER 4 BITS OF ACC
1170  0F36  0A                ASL A
1180  0F37  0A                ASL A
1190  0F38  A0 04             LDY #4
1200  0F3A  2A         ROTATE ROL A           ;SHIFT BIT TO CARRY
1210  0F3B  A2 FA             LDX #$FA        ;WRAP AROUND TO $F0
1220  0F3D  36 F6      ROTNXT ROL BUFFER+6,X  ;CARRY TO BUFFER TO CARRY
1230  0F3F  E8                INX             ;AND SO ON
1240  0F40  D0 FB             BNE ROTNXT      ;UNTIL END OF BUFFER
1250  0F42  88                DEY             ;DONE 4 BITS?
1260  0F43  D0 F5             BNE ROTATE      ;BRANCH IF NOT
1270  0F45  60                RTS             ;RETURN
1280  0F46             ;
1290  0F46             ;
1300  0F46             ;      RESET ENTRY POINT
1310  0F46             ;
1320  0F46  A9 00      RESET  LDA #0
1330  0F48  8D E0 08          STA $08E0       ;CLEAR DISPLAY AND PORTS
1340  0F4B  F0 08             BEQ COMAND      ;BRANCH ALWAYS
1350  0F4D             ;
1360  0F4D             ;
1370  0F4D             ;
1380  0F4D  20 34 0F   SHFTD  JSR SHIFT       ;SHIFT KEY INTO BUFFER
1390  0F50  A5 F0      DSPBUF LDA BUFFER      ;GET BUFFER
1400  0F52  8D 20 08   SEE    STA DISP        ;UPDATE DISPLAY
1410  0F55             ;
1420  0F55  A6 ED      COMAND LDX MSTACK
1430  0F57  9A                TXS             ;SET MONITOR STACK
1440  0F58  20 1F 0F          JSR GETKEY      ;WAIT FOR KEY
1450  0F5B  C9 10             CMP #$10        ;IS IT CONTROL KEY
1460  0F5D  90 EE             BCC SHFTD       ;BRANCH IF NOT
1470  0F5F  A8                TAY             ;CONTROL KEY INTO Y
1480  0F60  BE E2 0F          LDX TABLE-16,Y  ;GET COMMAND ADDR LOW
1490  0F63  86 EE             STX TEMP        ;SAVE IT
1500  0F65  A2 FF             LDX #$FF        ;GET COMMAND ADDR HIGH
1510  0F67  86 EF             STX TEMP+1      ;ASSEMBLE COMMAND ADDR
1520  0F69  E8                INX             ;CLR X
1530  0F6A  6C EE 00          JMP (TEMP)      ;EXECUTE COMMAND
1540  0F6D             ;
1550  0F6D             ;
```

```
 1560  0F6D  18          PHIGH  CLC
 1570  0F6E  A5 F6       PLOW   LDA PNTER        ;MOVE POINTER TO BUFFER
 1580  0F70  85 F0              STA BUFFER
 1590  0F72  A5 F7              LDA PNTER+1
 1600  0F74  85 F1              STA BUFFER+1
 1610  0F76  B0 D8              BCS DSPBUF       ;BRANCH IF POINTER LOW
 1620  0F78  90 D8              BCC SEE          ;BRANCH IF POINTER HIGH
 1630  0F7A              ;
 1640  0F7A              ;
 1650  0F7A  A5 F0       DISPLA LDA BUFFER       ;MOVE BUFFER TO POINTER
 1660  0F7C  85 F6              STA PNTER
 1670  0F7E  A5 F1              LDA BUFFER+1
 1680  0F80  85 F7              STA PNTER+1
 1690  0F82  B0 14              BCS LOAD         ;BRANCH ALWAYS
 1700  0F84              ;
 1710  0F84              ;
 1720  0F84  A5 F6       BACKSP LDA PNTER        ;DEC 16 BIT POINTER
 1730  0F86  D0 02              BNE SKIP         ;BRANCH IF NO BORROW
 1740  0F88  C6 F7              DEC PNTER+1
 1750  0F8A  C6 F6       SKIP   DEC PNTER
 1760  0F8C  B0 0A              BCS LOAD         ;BRANCH ALWAYS
 1770  0F8E              ;
 1780  0F8E              ;
 1790  0F8E  A5 F0       ENTER  LDA BUFFER       ;GET BYTE IN BUFFER
 1800  0F90  81 F6              STA (PNTER,X)    ;STORE IT IN ACTIVE CELL
 1810  0F92  E6 F6              INC PNTER        ;INC 16 BIT POINTER
 1820  0F94  D0 02              BNE LOAD         ;BRANCH IF NO CARRY
 1830  0F96  E6 F7              INC PNTER+1
 1840  0F98  A1 F6       LOAD   LDA (PNTER,X)    ;GET BYTE IN ACTIVE CELL
1850  0F9A  85 F0       STABUF STA BUFFER       ;STORE IT IN BUFFER
 1860  0F9C  B0 B2              BCS DSPBUF       ;BRANCH ALWAYS
 1870  0F9E              ;
 1880  0F9E              ;
 1890  0F9E  D8          RELADR CLD
 1900  0F9F  18                 CLC              ;THIS ADDS 1 TO POINTER
 1910  0FA0  A5 F0              LDA BUFFER       ;GET BUFFER LOW
 1920  0FA2  E5 F6              SBC PNTER        ;SUBTRACT POINTER LOW + 1
 1930  0FA4  85 F0              STA BUFFER       ;SAVE RESULTS
 1940  0FA6  A5 F1              LDA BUFFER+1     ;GET BUFFER HIGH
 1950  0FA8  E5 F7              SBC PNTER+1      ;SUBTRACT POINTER HIGH
 1960  0FAA  A8                 TAY              ;SAVE RESULTS IN Y
 1970  0FAB  A5 F0              LDA BUFFER       ;GET RESULTS LOW
 1980  0FAD  B0 08              BCS POS          ;BR IF TOTAL RESULT POS
 1990  0FAF  10 0A              BPL BAD          ;BR IF RESULT LOW POS
 2000  0FB1  C8                 INY              ;INC RESULT HIGH
2010  0FB2  98          CHK    TYA              ;CHECK RESULT HIGH
 2020  0FB3  D0 06              BNE BAD          ;BR IF NOT ZERO
 2030  0FB5  F0 99              BEQ DSPBUF       ;BR ALWAYS, DISP REL ADDR
 2040  0FB7  30 02       POS    BMI BAD          ;BR IF RESULT LOW NEG
 2050  0FB9  10 F7              BPL CHK          ;BR ALWAYS
 2060  0FBB  8A          BAD    TXA              ;CLEAR ACC
 2070  0FBC  38                 SEC
 2080  0FBD  B0 DB              BCS STABUF       ;BRANCH ALWAYS
 2090  0FBF              ;
 2100  0FBF              ;
 2110  0FBF  EA                 NOP
 2120  0FC0              ;
```

```
2130  0FC0                ;
2140  0FC0                ;
2150  0FC0                ;      BREAK ROUTINE ENTRY POINT
2160  0FC0                ;
2170  0FC0  85 F9         BREAK  STA ACC          ;SAVE ACCUMULATOR
2180  0FC2  84 FA                STY YREG         ;SAVE Y
2190  0FC4  86 FB                STX XREG         ;SAVE X
2200  0FC6  68                   PLA              ;GET STATUS REG
2210  0FC7  85 FF                STA PREG         ;SAVE IT
2220  0FC9  68                   PLA              ;GET PC LOW
2230  0FCA  D8                   CLD
2240  0FCB  38                   SEC
2250  0FCC  E9 02                SBC #2           ;CORRECT PC LOW
2260  0FCE  85 FC                STA PC           ;SAVE IT
2270  0FD0  68                   PLA              ;GET PC HIGH
2280  0FD1  E9 00                SBC #0           ;SUBTRACT CARRY
2290  0FD3  85 FD                STA PC+1         ;SAVE IT
2300  0FD5  BA                   TSX              ;GET USER STACK POINTER
2310  0FD6  86 FE                STX STACKP       ;SAVE IT
2320  0FD8  A9 BB                LDA #$BB         ;BREAK INDICATION
2330  0FDA  B0 BE                BCS STABUF       ;BRANCH ALWAYS
2340  0FDC                ;
2350  0FDC                ;
2360  0FDC  A6 FE         RUN    LDX STACKP       ;GET USER STACK POINTER
2370  0FDE  9A                   TXS              ;INIT STACK
2380  0FDF  A5 F1                LDA BUFFER+1     ;GET PC HIGH
2390  0FE1  48                   PHA              ;PUT IT ON STACK
2400  0FE2  A5 F0                LDA BUFFER       ;GET PC LOW
2410  0FE4  48                   PHA              ;PUT IT ON STACK
2420  0FE5  A5 FF                LDA PREG         ;GET STATUS REG
2430  0FE7  48                   PHA              ;PUT IT ON STACK
2440  0FE8  A6 FB                LDX XREG         ;RESTORE X
2450  0FEA  A4 FA                LDY YREG         ;RESTORE Y
2460  0FEC  A5 F9                LDA ACC          ;RESTORE ACCUMULATOR
2470  0FEE  40                   RTI              ;RESTORE PC & STATUS REG
2480  0FEF                ;                         FROM STACK AND EXECUTE
2490  0FEF                ;                         USER'S PROGRAM
2500  0FEF                ;
2510  0FEF                ;
2520  0FEF  4C 00 0E      TAPE   JMP TAPE1        ;EXECUTE TAPE OPTION
2530  0FF2                ;
2540  0FF2                ;
```

```
2550  0FF2                ;      COMMAND ADDRESS TABLE
2560  0FF2                ;      STORES LOW BYTE ONLY OF ENTRY
2570  0FF2                ;      ADDRESS FOR EACH COMMAND
2580  0FF2                ;
2590  0FF2  DC 0F         TABLE   .WORD RUN
2600  0FF4                        *=*-1
2610  0FF3  7A 0F                 .WORD DISPLA
2620  0FF5                        *=*-1
2630  0FF4  84 0F                 .WORD BACKSP
2640  0FF6                        *=*-1
2650  0FF5  8E 0F                 .WORD ENTER
2660  0FF7                        *=*-1
2670  0FF6  6D 0F                 .WORD PHIGH
2680  0FF8                        *=*-1
2690  0FF7  6E 0F                 .WORD PLOW
2700  0FF9                        *=*-1
2710  0FF8  EF 0F                 .WORD TAPE
2720  0FFA                        *=*-1
2730  0FF9  9E 0F                 .WORD RELADR
2740  0FFB                        *=*-1
2750  0FFA                ;
2760  0FFA                ;
2770  0FFA  03 00                 .WORD $0003      ;NMI VECTOR
2780  0FFC  46 0F                 .WORD RESET      ;RESET VECTOR
2790  0FFE  00 00                 .WORD $0000      ;IRQ VECTOR
2800  1000                ;
2810  1000                ;
2820  1000                        .END


ERRORS = 0000


SYMBOL TABLE

 RESULT  0F19     DLY     0F2B     COMAND  0F55     LOAD    0F98
 SKIP    0F8A     POS     0FB7     BAD     0FBB     TABLE   0FF2
 KEY     0800     TEMP    00EE     LASTKE  00F8     BUFFER  00F0
 DISP    0820     MSTACK  00ED     PNTER   00F6     TAPE1   0E00
 CASS    0900     ACC     00F9     YREG    00FA     XREG    00FB
 PC      00FC     STACKP  00FE     PREG    00FF     DECODE  0F00
 LOOP    0F04     NEXT    0F08     GETKEY  0F1F     BEEP    0F22
 NXTX    0F24     DELAY   0F26     SHIFT   0F34     ROTATE  0F3A
 ROTNXT  0F3D     RESET   0F46     SHFTD   0F4D     DSPBUF  0F50
 SEE     0F52     PHIGH   0F6D     PLOW    0F6E     DISPLA  0F7A
 BACKSP  0F84     ENTER   0F8E     STABUF  0F9A     RELADR  0F9F
 CHK     0FB2     BREAK   0FC0     RUN     0FDC     TAPE    0FEF
```