```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; $Id: DOS HOSS FOR IIe V12 2/8/2026
;
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; RESET ROUTINE FOR BRAIN BOARD running in IIe
; Processor: 6502
;
; THIS CODE ASSUMES DOS HOSS IN SLOT 4
; load this code into D0 - f8 ROM
;
;
; HOT KEYS-
; CONTROL + B = LOAD DOS FROM BRAIN BOARD
; CONTROL + C = LOAD DOS AND CATALOG DISK
; CONTROL + M = ENTER MONITOR. DOS NOT LOADED. 3D0G TO EXIT MONITOR STARTS DOS
; CONTROL + OPEN APPLE = NORMAL RESET - FORCES REBOOT
; CONTROL + CLOSED APPLE = NORMAL RESET - FORCES SELF TEST
;
; THESE ARE HANDY DEFINES FOR THE COMPILER
;
#define     EQU     .EQU
#define     ORG     .ORG
#define     RMB     .BLOCK
#define     FCB     .BYTE
#define     FCC     .TEXT
#define     FDB     .WORD
#define     DW    .WORD ;define word
#define     DFB   .BYTE ;define byte
#define     equ     .EQU
#define     org     .ORG
#define     rmb     .BLOCK
#define     fcb     .BYTE
#define     fcc     .TEXT
#define     fdb     .WORD
#define     dfb   .BYTE
#define     END   .END
#define     HOFFLOW     #$C1  ;This is the slot number low byte $C1 = Slot 4
                ;$D1 = Slot 5
                ;use #define to assign a value to a variable name
;
; These equates are used by the Move Dos routine
;
SRCE        EQU   $6
SRCEHI            EQU   $7
DSTN        EQU   $8
DSTNHI            EQU   $9
MOVEP3            EQU   $9E25
RTSTEMP           EQU   $9E41
;
;
; These equates are used by the main DosHoss initialization code
;
PROMPT            EQU   $33
CSWL        EQU   $36
```

```
A1L         EQU   $3C
A1H         EQU   $3D
A2L         EQU   $3E
A2H         EQU   $3F
A4L         EQU   $42
A4H         EQU   $43
BFFRFLG           EQU   $45
ONERR       EQU   $D8
DOSWARM           EQU   $3D0
DOSWRMHI    EQU   $3D
DOSWRMLO    EQU   $00
PWRUP       EQU   $3F4  ;THIS IS PWREDUP HOLDS RANDOM VALUE IF A POWER UP
SAVEPWRUP   EQU   $2000 ;THIS IS PWREDUP HOLDS RANDOM VALUE IF A POWER UP
CATALOG           EQU   $A56E
CATALOGHI   EQU   $A5   ;Catalog disk
CATALOGLO   EQU   $6E   ;Catalog disk
DOSCSWL           EQU   $AA53 ;DOS output hook. Default is $FDF0
DOSCMDX           EQU   $AA5F ;DOS 3.2/3.3 Command number
DOSSTRT           EQU   $B73A ;
READKEY           EQU   $C000 ;Peek to read keyboard. Value > 127 means a key has been
pressed.
KEYSTROBE   EQU   $C010 ;Keyboard strobe. READ TO CLEAR STROBE.
SAVEKEY           EQU   $2001 ;SAVE THE LAST KEY PRESSED
SPEAKER           EQU   $C030 ;ONLY READ LOCATION $C030. SATHER DID AN STA IN HIS ORIGINAL
CODE
AN0OFF            EQU   $C058 ;Poke 0 to Clear game AN0
AN1OFF            EQU   $C05A ;Poke 0 to Clear game AN1
OPENAPL           EQU   $C061 ;*** NOT IN II+
SLDAPL            EQU   $C062 ;*** NOT IN II+
RDONLY            EQU   $C080 ;Write PROTECT RAM. Select 2nd bank of $D000 - $DFFF RAM in
language card.
                        ;$C084 does the same as $C080.
WRTONLY           EQU   $C081 ;Write ENABLE RAM. in II+ DOS 3.2 Read_Write track sector and
also
                        ;Read-Deselect 2nd bank of $D000 - $DFFF RAM in language card.
HRAMOFF           EQU   $C082 ;Write PROTECT RAM. in II+ DOS 3.2 Read_Write track sector
and also
                        ;Read-Deselect 2nd bank of $D000 - $DFFF RAM in language card.
;HOSSOFF    EQU   $C0D1 ;EXIT THE DOSHOSS ROM to Motherboard for slot 5
;BANK1             EQU   $C0D2 ;This switched to the upper bank of ROM ** no longer used by
my code
HOSSOFF           EQU   $C0C1 ;EXIT THE DOSHOSS ROM to Motherboard FOR SLOT 4 !! CURRENTLY
NOT USED
BANK1       EQU   $C0C2 ;This switched to the upper bank of ROM ** no longer used by my
code
CLRROM            EQU   $CFFF ;Turn off Flip Flops. Disable expansion ROM
BASIC       EQU   $DFFF ;*** Looks like garbage in the II+.
NRESET            EQU   $FA62 ;Normal Reset Vector for II+ and IIe
INIT        EQU   $FB2F ;Screen init. Reset Text mode
GOTOCX            EQU   $FBB4 ;*** Monitor memory location MD3.In new F8 ROM code here
saves ROM states
HOME        EQU   $FC58 ;HOME. Clear screen. Cursor top left
WAIT        EQU   $FCA8 ;Call Wait loop. Set A register with calculation value
MOVE        EQU   $FE2C ;monitor memory move
SETNORM           EQU   $FE84 ;Set video output to normal
MONITOR           EQU   $FF58 ;JSR here to find out where one is. Sets Overflow flag.
NMIDSTN           EQU   $FFFA ;Address of NMI (Non-Maskable interupt) vector
;
; CURRENTLY THE SLOT IS HARD CODED AT THE BOTTOM OF THIS CODE.
```

```
; $C1 OR $D1 TO SPECIFY THE SLOT.
;
; Set the slot value in HOFFLOW to specify the slot. See the EQU above.
;
; ** Note: I did not include Integer Basic at this timne.
; There was no room in the Brain Board bank 0 to fit Integer and Dos 3.3.
; All I really wanted was to have DOS load instantly on power up.
; Later I plan to figure out a way to load it from Bank 1 from assembly code I
; write into Apple RAM like I am doing to shut off the Brain Board.
;
           .ORG  $D000
;*******************************************************
;
;     BEGIN RESET HANDLER
;     BEGIN RESET HANDLER
; RESET VECTOR IS FFFC AND FFFD. THIS CODE BEGINS AT $D000.
; ON POWER UP, THE F8 ROM SECTION HAS FFFC-FFFD LOADED WITH D000
; SO THE CPU VECTORS TO THIS PROGRAM. ONCE DOS HAS BEEN LOADED INTO THE APPLE RAM
; THIS PROGRAM PLACES A SHORT ASSEMBLER ROUTINE AT $1000. THIS ROUTUINE TURNS
; OFF THE BRAIN BOARD AND THEN DOES SOME RESET STUFF. IT ALSO CHECKS
; FOR THE HOT KEYS TO CATALOG DISK OR GO INTO THE MONITOR.
; THIS ROUTINE CHANGES HOW SATHER DID THE EXIT. HIS DOSHOSS WAS EXPECTING AN RTS
; TO BE AT A MOTHERBOARD ROM LOCATION AFTER THE DOSHOSS WAS TURNED OFF. IT SEEMS MORE
; FLEXIBLE TO HAVE A PROGRAM RUNNING IN THE APPLE MEMORY THAT TURNS OFF THE BRAIN BOARD AND
; CAN THEN CLEAN UP REGISTERS, CLEAR SCREEN AND COMPLETE THE BOOT.
;
;
;  CLEAR THE RAM AT $2000 TO $00's
;
_BEGIN            LDX   #$FF
           LDY   #$00
_CLEAN            LDA   #$00
           STA   $2000,Y     ;CLEAN MEMORY TO BEGIN
           INY
           DEX
           CPX   #$FF
           BNE   _CLEAN
           JMP   _DOSTART    ;
;
;
_VERSION    .TEXT "WORKSV11.ASM 2-8-2026"
;
_DOSLINK    .WORD $9EBD
           .WORD $9E81
;
;
_DOSTART    BIT   SLDAPL
           BMI   _DONORM1    ;BRANCH IF 7TH BIT IS SET = KEY DOWN
           BIT   OPENAPL
           BPL   _NOBUTTN    ;BRANCH IF 7TH BIT IS CLEAR = KEY UP
_DONORM1    LDA   #$A0        ;OPEN OR CLOSED APPLE WAS DOWN
           STA   SAVEKEY           ;SO DO A NORMAL RESET BASED ON WHICH
           JMP   _NORMRES    ;WHICH APPLE KEY WAS HELD DOWN ON RESET
_DONORM2    LDA   #$CA
           STA   SAVEKEY
           JMP   _NORMRES
;
;
_NOBUTTN    LDA   PWRUP-1           ;LOAD POWER UP STATUS
```

```
            EOR   #$A5         ;
            CMP   PWRUP
            STA   PWRUP
            STA   SAVEPWRUP    ;CURIOUS ABOUT THE LAST KEY PRESSED
            BNE   _DOHOSS            ;BRANCH IF NOT A POWER UP
            LDA   KEYSTROBE    ;CHECK FOR KEYPRESS
            BPL   _DONORM            ;NO KEY DOWN, NOT A POWER UP
            LDA   READKEY
            AND   #$7F         ;DON'T CARE ABOUT MSB
            STA   SAVEKEY            ;SAVE LAST KEY PRESSED
_DOHOSS     CLD
            JSR   SETNORM            ;DO A BIUNCH OF RESET STUFF
            JSR   INIT
            JSR   HOME
            LDA   AN0OFF
            LDA   AN1OFF
            LDY   #5
            JSR   GOTOCX
            LDA   CLRROM
            JSR   _DUMBELL     ;FIRST DIFFERENT SPEAKER BEEP
            LDA   READKEY            ;WAS A POWER UP IF NO KEY DOWN
            BPL   _NOHELLO
            LDA   SAVEKEY
;           AND   #$7F         ;DON'T CARE ABOUT MSB
;           CMP   #$41         ;MASK LETTERS INTO CONTROL CODE
;           BCC   _LESS        ;BCC    *+4  BRANCH AHEAD 4 BYTES FROM PROG COUNTER
;           AND   #$1F
_LESS       TAY                ;I CORRECTED THIS LINE 12-18-2025
            CPY   #$03         ;C?
            BEQ   _RESET.C     ;CATALOG THE DISK IF C KEY DOWN
            LDX   #3
_LINKLP1    LDA   DOSCSWL,X    ;IF CSW AND KSW ARE EQUAL
            CMP   CSWL,X             ;TO DOSCSW AND DOSKSW
            BNE   _LINKDNE     ;ASSUME DOS WAS IN USE
            DEX                ;AND RECONNECT DOS
            BPL   _LINKLP1     ;THIS ONLY MATTERS IN BOOTLESS
            LDX   #3           ;RESETS SUCH AS CONTRPOL-I
_LINKLP2    LDA   _DOSLINK,X   ;THE PURPOSE IS TO PREVENT
            STA   CSWL,X             ;A CONTROL-I RESET FROM
            DEX                ;DISCONNECTING DOS
            BPL   _LINKLP2
_LINKDNE    NOP
            CPY   #$0D         ;m ??
            BEQ   _RESET.M     ;GO TO MONITOR
            LDA   #$1B         ;NO HELLO FLAG
            PHA
            CPY   #$02         ;B ??
            BEQ   _RESET.B     ;BOOT ONLY
            PLA
            LDA   #$0          ;DO HELLO FLAG
            PHA
            CPY   #$08         ;H ??
            BEQ   _RESET.B     ;MOVE DOS - WAS _DOMOVE BRANCH
            PLA
_NOHELLO    LDA   #$1B         ;NO HELLO FLAG
            PHA
_RESET.B    JSR   _MOVEDOS     ;XFER DOS
            PLA                ;GET DO HELLO FLAG
            STA   DOSCMDX
```

```
            LDA   #<DOSSTRT   ;COLD START IS $B73B
            PHA
            LDA   #>DOSSTRT
            PHA
_DONORM           JMP   _GOMRBRD
;
;
_RESET.C    NOP
            NOP
            LDX   #3          ;SET CSW AND KSW TO I/O ROUTS
_CATLP            LDA   DOSCSWL,X
            STA   CSWL,X
            DEX
            BPL   _CATLP
;           LDA   DOSWRMLO    ;LOCATION $3D0
;           PHA               ;WARM START DOS AFTER CATALOG
;           LDA   DOSWRMHI
;           PHA
;           LDA   CATALOGLO   ;$A56E (DO CATALOG)
;           PHA
;           LDA   CATALOGHI
;           PHA
            LDA   ONERR       ;FIX APPLESOFT ONERR FLAG
            AND   #$7F        ;SO DISKI/O ERROR WILL PRINT
            STA   ONERR       ;
            LDA   #1          ;FIX NO BUFFER FLAG FOR CATALOG
            STA   BFFRFLG
;           JSR   CATALOG
;           JSR   DOSWARM
            JMP   _GOMRBRD    ;EXIT DOS HOSS
;
;
;
_RESET.M    NOP
            NOP
            LDA   #<MONITOR   ;GO TO MONITOR ($FF59)
            PHA
            LDA   #>MONITOR
            PHA
            JMP   _GOMRBRD    ;EXIT DOS HOSS
;
;
;
;
;_NMISRCE   DW    $3FB
;           DW    $FA62
;
;
;
_DUMBELL    LDY   #$16        ;SLIGHTLY DIFFERENT BELL SOUND
_BELP1            LDX   #8
_BELP2            TYA
            JSR   WAIT
            LDA   SPEAKER           ;WRITE $C030 LOCATION
            DEX               ;
            BNE   _BELP2
            DEY
            BNE   _BELP1
            RTS
```

```
;
;
;
_MOVEDOS    NOP
            NOP
;
;  Move Dos
;
; I changed Sather's code and have everything in the same ROM bank.
;
;           DFB   $FF,$FF,$FF ;Not needed since Move Dos is in the same
;           JSR   _DOSAWAY    ;ROM bank.
;           RTS               ;
;
;
;
;
_DOSAWAY    LDA   #0
            STA   SRCE
            STA   DSTN
            LDA   #$9D
            STA   DSTNHI
            LDA   #$D4        ;$D400 SHOULD BE START OF DOS IN D000 ROM
            STA   SRCEHI            ;I had to pack the DOS from the original
            LDY   #0          ;DosHoss ROMs into Bank0 of the Brain Board.
_MOVELP2    LDA   (SRCE),Y
            STA   (DSTN),Y
            INY
            BNE   _MOVELP2
            INC   DSTNHI
            INC   SRCEHI
            LDA   SRCEHI
            CMP   #$F7        ;DOS should end at F6FF
            BNE   _MOVELP2    ;SO DO A COMPARE OF THE SRCEHI TO F7 AND BRANCH UNTIL EQUAL
;                             ;CHECKING USING BCC IF SRCEHI IS LESS THAN OR EQUAL TO #$F7
;
            LDA   #$60
            STA   RTSTEMP
            JSR   MOVEP3
            LDA   #$A9
            STA   RTSTEMP
            RTS
;
; Now exit to the Apple Motherboard ROM
;
_GOMRBRD    LDA   #$A9
            STA   $1000
            LDA   #-1         ;SETUP TO CLEAR THE STACK
            STA   $1001
            LDA   #$9A        ;TXS WILL CLEAR THE STACK
            STA   $1002
            LDA   #$AD        ;
            STA   $1003
            LDA   HOFFLOW           ;D1 = SLOT 5, C1 = SLOT 4
; Should also be able to get the low byte of the Brain Board slot as such:
;  (#$HOSSOFF & $00FF) replaces HOFFLOW
;
; Interesting observation. If you put this card into the wrong slot, on computer boot
; it displays 'HACKIN FOOL!' instead of Apple ][ and it locks up. Seems to be related to
```

```
; when I changed the code above from hard coded slot low byte to using HOFFLOW. The
assembler
; saw this as doing a LDA zero page instead of LDA absolute.
;
;
            STA   $1004       ;CHANGE HOFFLOW EQU VALUE TO SET SLOT
            LDA   #$C0        ;LDA $C0XX TO TURN OFF BRAIN BOARD
            STA   $1005       ;
            LDA   #$A9        ;Next I am just doing some Reset stuff
            STA   $1006       ;to clean things up after switching from
            LDA   #-1         ;the Brain Board ROM.
            STA   $1007
            LDA   #$9A        ;
            STA   $1008       ;
;
;
;
            LDA   #$D8        ;CLD
            STA   $1009
            LDA   #$20
            STA   $100A
            LDA   #$84        ;
            STA   $100B
            LDA   #$FE        ;
            STA   $100C
            LDA   #$20        ;
            STA   $100D
            LDA   #$2F        ;
            STA   $100E
            LDA   #$FB        ;
            STA   $100F
            LDA   #$20
            STA   $1010
            LDA   #$93
            STA   $1011
            LDA   #$FE
            STA   $1012
            LDA   #$20
            STA   $1013
            LDA   #$89
            STA   $1014
            LDA   #$FE
            STA   $1015
;
;
            LDA   #$20        ;LDA ABSOLUTE
            STA   $1016
            LDA   #$60        ;LOW BYTE SAVEKEY
            STA   $1017
            LDA   #$FB        ;HI BYTE SAVEKEY
            STA   $1018
            LDA   #$AD        ;CMP IMMEDIATE TO $0D CONTROL + M
            STA   $1019
            LDA   #$01        ;SAVEKEY LOW BYTE
            STA   $101A
            LDA   #$20        ;BNE
            STA   $101B
            LDA   #$C9        ;CMP IMMEDIATE TO $03 CONTROL + C
            STA   $101C       ;
```

```
            LDA   #$03        ;
            STA   $101D       ;
            LDA   #$D0        ;BNE
            STA   $101E
            LDA   #$03        ;JUMP 3 BYTES FORWARD IF NOT EQUAL
            STA   $101F
            LDA   #$20        ;
            STA   $1020
            LDA   #$6E              ;Catalog the disk by calling $A56E
            STA   $1021
            LDA   #$A5
            STA   $1022
;
            LDA   #$AD
            STA   $1023
            LDA   #$01
            STA   $1024
            LDA   #$20
            STA   $1025
            LDA   #$C9        ;CMP IMMEDIATE TO #0D (Control+M)
            STA   $1026       ;JMP TO $FF59 WHICH IS THE MONITOR
;
;
            LDA   #$0D        ;
            STA   $1027
            LDA   #$D0        ;BNE
            STA   $1028
            LDA   #$03        ;JUMP 3 BYTES FORWARD IF NOT EQUAL
            STA   $1029
            LDA   #$20        ;
            STA   $102A
            LDA   #$65        ;
            STA   $102B
            LDA   #$FF        ;
            STA   $102C
            LDA   #$EA        ;
            STA   $102D
            LDA   #$EA        ;
            STA   $102E
            LDA   #$EA
            STA   $102F
            LDA   #$EA
            STA   $1030
            LDA   #$EA
            STA   $1031
            LDA   #$4C        ;
            STA   $1032
            LDA   #$84
            STA   $1033
            LDA   #$9D
            STA   $1034
;
; Remember you can use CONTROL + B TO EXIT MONITOR AND LOAD DOS
;
            LDA   #$EA
            STA   $1035
;
            LDA   #$EA
            STA   $1036
```

```
            LDA   #$EA
            STA   $1037
            LDA   #$EA
            STA   $1038
            LDA   #$EA
            STA   $1039
            LDA   #$EA
            STA   $103A
            LDA   #$60
            STA   $103B
;
;
;
            JMP   $1000       ;JMP 1000  4C 00 10
;
;
;  Here we handle a control reset with one of the Apple keys held down
;
;
_NORMRES    LDA   #$A9        ;NEED TO TURN OFF THE BRAIN BOARD
            STA   $1050       ;THEN DO RESET BASED ON APPLE KEY
            LDA   #-1         ;SETUP TO CLEAR THE STACK
            STA   $1051
            LDA   #$9A        ;TXS WILL CLEAR THE STACK
            STA   $1052
            LDA   #$AD        ;
            STA   $1053
            LDA   HOFFLOW           ;D1 = SLOT 5, C1 = SLOT 4
;
;
            STA   $1054       ;CHANGE HOFFLOW EQU VALUE TO SET SLOT
            LDA   #$C0        ;LDA $C0XX TO TURN OFF BRAIN BOARD
            STA   $1055
            LDA   #$4C        ;DEPENDING ON APPLE KEY DOWN
            STA   $1056       ;THE APPLE WILL DO SELF TEST
            LDA   #$62        ;OR REBOOT.
            STA   $1057       ;OPEN APPLE KEY FORCES REBOOT.
            LDA   #$FA        ;CLOSED APPLE KEY FORCES SELF TEST
            STA   $1058
            LDA   #$60
            STA   $1059
            JMP   $1050
;
;
;           JMP   $FA62       ;THE OPEN AND CLOSED APPLE KEYS ARE
;           RTS               ;CHECKED DURING RESET
;
;
STOP        END               ;
```