

```

C792:D0 F1 C785 246 bne blp3
C794:AD 61 C0 247 blp4 LDA $C061 ;test for both Open and Closed Apple
C797:2D 62 C0 248 AND $C062 ; pressed
C79A:0A 249 asl a ;put result in carry
C79B:E6 FF 250 INC $FF
C79D:A5 FF 251 LDA $FF
C79F:90 03 C7A4 252 bcc dquit
C7A1:4C 00 C6 253 jmp DIAGS
C7A4: 254 *
C7A4:AD 51 C0 255 dquit lda TEXT ;put success message on the screen
C7A7:A0 08 256 ldy #8
C7A9:B9 F6 C7 257 suc2 lda success,y
C7AC:99 B8 05 258 sta SCREEN,y
C7AF:88 259 dey
C7B0:10 F7 C7A9 260 bpl suc2
C7B2:30 E0 C794 261 bmi blp4 ;loop forever
C7B4: 262 *
C7B4: C7B4 263 setv equ *
C7B4:53 43 2B 29 264 ntbl dfb 83,67,43,41,7
C7B9:00 89 31 03 265 swtbl0 dfb $00,$89,$31,$03,$05,$09,$0b,$01,$00,$83,$51,$53,$55,$57,$0F,$0D
C7C9:00 81 31 04 266 swtbl1 dfb $00,$81,$31,$04,$06,$0A,$0C,$02,$00,$84,$52,$54,$56,$58,$10,$0E
C7D9:00 11 FF 13 267 rswtbl dfb $00,$11,$FF,$13,$14,$16,$17,$18,$00,$12,$1A,$1B,$1C,$1D,$1E,$1F,$00
C7EA: 268 MSB ON
C7EA:D2 C1 CD A0 269 rmess asc "RAM ZP"
C7F0:CD CD D5 C9 270 smess asc "MMUIOU"

C7F6:D3 F9 F3 F4 272 success asc "System OK"
C7FF: C7FF 273 zzzend equ *
C7FF: 22 INCLUDE C8SPACE
C7FF: 0001 1 DS C8ORG-$,0 ;pad to C800
C800: 2 *
C800: 3 * This entry point is only used by Pascal 1.0
C800: 4 *
C800:4C B0 C9 5 JMP PINIT1.0 ;PASCAL 1.0 INIT
C803: 6 *
C803: 7 * BASIC initialization:
C803: 8 *
C803: 9 * This is called by the $C3 space only after a PR#3 or
C803: 10 * the equivalent (a JSR $C300).
C803: 11 *
C803: 12 * It causes a copy of the $F8 ROM to be placed in the
C803: 13 * language card if the language card is switched in and
C803: 14 * the ID byte doesn't match. It sets up all the
C803: 15 * screenhole variables to support its operation. If the
C803: 16 * 80 column card is detected, it sets things up for 80 column
C803: 17 * operation, else 40 column operation. Then it clears the
C803: 18 * screen and prints the character that was in the accumulator
C803: 19 * upon entry.
C803: 20 *
C803: C803 21 BASICINIT EQU *
C803:20 F4 CE 22 JSR COPYROM ;If LC in, copy F8 to it
C806:20 2A C8 23 JSR C3HOOKS ;out=$C307, in=$C305
C809:20 2E CD 24 JSR DO40 ;set full 40-col window

```

```

C80C:A9 01 25 LDA #M.MOUSE ;init with mouse text off
C80E:8D FB 04 26 STA MODE ;Set BASIC video mode
C811: 27 *
C811: 28 * IS THERE A CARD?
C811: 29 *
C811:20 90 CA 30 JSR TESTCARD ;SEE IF CARD PLUGGED IN
C814:D0 08 C81E 31 BNE CLEARIT ;=>IT'S 40
C816:06 21 32 ASL WNDWIDTH ;SET 80-COL WINDOW
C818:8D 01 C0 33 STA SET80COL ;ENABLE 80 STORE
C81B:8D 0D C0 34 STA SET80VID ; AND 80 VIDEO
C81E: 35 *
C81E: 36 * HOME & CLEAR:
C81E: 37 *
C81E: C81E 38 CLEARIT EQU *
C81E:8D 0F C0 39 STA SETALTCHAR ;SET NORM/INV LCASE
C821:20 90 CC 40 JSR X.FF ;CLEAR IT
C824:AC 7B 05 41 LDY OURCH ;set up cursor for store
C827:4C 7E C8 42 JMP BPRINT ;always print a character
C82A: 43 *
C82A:A9 07 44 C3HOOKS LDA #>BASICOUT ;set output hook first
C82C:85 36 45 STA CSWL
C82E:A9 C3 46 LDA #<CNOO
C830:85 37 47 STA CSWH
C832: 48 *
C832: 49 * C3IN is called by IN#0 if CSWH = #<C3
C832: 50 *
C832:A9 05 51 C3IN LDA #>BASICIN ;set input hook
C834:85 38 52 STA KSWL
C836:A9 C3 53 LDA #<CNOO
C838:85 39 54 STA KSWH
C83A:60 55 RTS ;exit with A=$C3 for IN#0 stuff
C83B: 56 *
C83B:E6 4E 57 GETKEY INC RNDL ;BUMP RANDOM SEED
C83D:D0 02 C841 58 BNE GETK2
C83F:E6 4F 59 INC RNDH
C841:AD 00 C0 60 GETK2 LDA KBD ;KEYPRESS?
C844:10 F5 C83B 61 BPL GETKEY ;=>NOPE
C846:8D 10 C0 62 STA KBDSTRB ;CLEAR STROBE
C849:60 63 RTS
C84A: 64 *
C84A: 65 *****
C84A: 66 *
C84A: 67 * PASCAL 1.0 INPUT HOOK:
C84A: 68 *
C84A: 0003 69 DS C8ORG+$4D-$,0 ;pad to 1.0 hooks
C84D: 0000 70 IFNE *-C8ORG-$4D ;ERR IF WRONG ADDR
71 FAIL 2,'C84D HOOK ALIGNMENT'
C84D: 72 FIN
C84D:4C 50 C3 73 JMP JPREAD ;=>GO TO STANDARD READ
74 *****
C850: 75 *
C850: 76 * CSETUP compensates for everything that the user
C850: 77 * can do to change the cursor status: poke CV, CH,
C850: 78 * OURCH, WNDWIDTH. It updates the video firmware's

```