


```

0000 1: ;-----
0000 2: ;
0000 3: ; Copyright 2010 Intergraded Logic Systems
0000 4: ; Source Code is Copyright Stephen J. Car
0000 5: ;
0000 6: ;
0000 7: ; Please do not share this source!
0000 8: ;-----
0000 9: ; Build header source
0000 10: ;
0000 11: ; Notes: o This source code MAY NOT be placed for download
0000 12: ; o "mea" is a macro that loads the address of the
0000 13: ; into the pointer specified by the second field.
0000 14: ;-----
0000 15: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 16: ; XASM st and IBM VERSIONS
0000 17: ;-----
0000 18: ;
0000 19: ; init design to detect know hardware
0000 20: ;
0000 21: ;
0000 22: ; last updated 3/9/2010 Added header
0000 23: ;
0000 24: ; added sio2usb detection 8/27/2012
0000 25: ;
0000 26: ; added sio2sd detection 9/10/2012
0000 27: ; added sio2ide detection 9/12/2012
0000 28: ; added sdrive detection 9/19/2012
0000 29: ;
0000 30: ;=====
0000 31:
0000 32: com_file_name: .macro
0000 33: ; dc.b 12,"filename.ext",$9b
0000 34: ; dc.b 12,"Hardware.com",$9b
0000 35: ; .endm
0000 36:
0000 37: ;
0000 38: ;~~~~~
0000 39: ; File revision history. Version Number in hex
0000 40: ;
0000 41: file_ver: equ $14
0000 42: ;~~~~~
0000 43: ; the Month in dec for the first time this revision
0000 44: ;was compiled
0000 45: ;
0000 46: c_month: equ 9
0000 47: ;~~~~~
0000 48: ; the day in dec for the first time this revision
0000 49: ;was compiled
0000 50: c_day: equ 19
0000 51: ;~~~~~
0000 52: ; the year in dec for the first time this revision
0000 53: ;was compiled

```

```

0000      54: c_year:                equ 2012
0000      55: ;~~~~~
0000      56: ; Type of compiler used in program
0000      57: ;
0000      58: ; 0 = unknown Compiler
0000      59: ; 1 = xasm
0000      60: ; 2 = mac_65
0000      61: ; 3 = basic
0000      62: ; 4 = compiled basic xl
0000      63: ; 5 = C65
0000      64: ; 6 = Action
0000      65: ;
0000      66: ; We can add more as time goes on!
0000      67: ;
0000      68: ;
0000      69: xasm:                equ 1
0000      70: mac_65:              equ 2
0000      71: basic:              equ 3
0000      72: basicxl:            equ 4
0000      73: c65:              equ 5
0000      74: action:              equ 6
0000      75:
0000      76: file_compiler:      equ xasm
0000      77: ;~~~~~
0000      78: ; is this relocatable code ?
0000      79: ; anyother value other than 1 or 2 would be unknown
0000      80: r..yes:              equ 1
0000      81: r..no:               equ 2
0000      82: ;
0000      83: ;~~~~~
0000      84: ; Gotta define if it can be relocated
0000      85: l_relocatable:      equ r..no
0000      86: ;
0000      87: ;
0000      88: ;~~~~~
0000      89: ; Gotta define our Header control Byte
0000      90: ;$ff=bypass intro
0000      91: ;$15 = Bypass Dos check
0000      92: r..crl:              equ $7d
0000      93: r..crlf:             equ $9b
0000      94: r..space:            equ $20
0000      95: r..bypassDOS:        equ $15
0000      96: r..bypass:           equ $ff
0000      97:
0000      98: l_frstscreenbyte:    equ r..space
0000      99: ;~~~~~
0000     100: ; The language the output file is in.
0000     101: ;
0000     102: ;
0000     103: ; 0 = Undefined
0000     104: ; 1 = English
0000     105: ; 2 = German
0000     106: ;

```

```

0000 107: r..language: equ 1
0000 108: ;~~~~~
0000 109: ;-----
0000 110: ; SIOV commands that SIO2IDE use
0000 111: ;
0000 112: ; SIO commands accepted by SIO2IDE FS
0000 113: ;
0000 114: pgm_sio2ide: equ $71 ; Program sio2ide i
0000 115: SIOC_GETDSK: equ $11 ; Get the Disk desc
0000 116: ;
0000 117: ide_master: equ $01 ; Master IDE Device
0000 118: ide_slave: equ $02 ; Slave IDE Device
0000 119: ;-----
0000 120: ;
0000 121: siov_read: equ $40 ; Read
0000 122: siov_write: equ $80 ; write
0000 123: ;-----
0000 124: ;
0000 125: ide_disk1: equ 1 ; Emulated Disk 1
0000 126: ;
0000 127: ide_timeout: equ 20 ; sio timeout
0000 128: xfersize: equ 15 ; Buffer size we ar
0000 129: ;
0000 130: ide_inuse: equ ide_master ; Ide we are progra
0000 131: siov_do: equ siov_read
0000 132: ide_disk: equ ide_disk1 ; Aux1
0000 133: ;-----
0000 134:
0000 135:
0000 136:
0000 137: .include equates
0000 138: .include globals
0000 139: .include macros
0000 140:
0000 141: .org $3000
0000 142:
0000 143:
0000 144: win_start:
0000 145: header_info:
0000 146: .include header ; t
32a7 147:
32a7 148: top:
32a7 ba 149: TSX
32a8 8e6e80 150: STX hold_stack
32ab 151:
32ab 205680 152: jsr get_xsio ; G
32ae ad1fd0 153: lda consol ; c
32b1 c907 154: cmp #7 ; l
32b3 f003 155: beq .a ; n
32b5 4cd88d 156: jmp help_menu ; S
32b8 157: .a:
32b8 ad0bd4 158: lda vcount
32bb d0fb 159: bne .a

```

```

32bd      160:
32bd 204635 161:          jsr  memory_find      ; t
32c0      162:
32c0      163: ;          jsr  cleanup          ; P
32c0      164:
32c0 4c0080 165:          jmp  show_hardware
32c3      166:
32c3      167:
32c3      168:
32c3      169: ;~~~~~
32c3      170: ; THis is the memory Finder routeen
32c3      171: ;
32c3      172: ; Syntax: jsr memory_find
32c3      173: ;
32c3 00     174: mepoint:      dc.b  0
32c4      175: ;
32c4      176: total_ram_banks:
32c4 0000    177: i_t_ram_banks:  dc.w  0
32c6      178: i_ram_lookup:
32c6      179:          ds.b  128
3346      180: ;
3346      181: m_buffer:
3346      182:          ds.b  256
3446      183: ;
3446      184: m_buffer_2:
3446      185:          ds.b  256
3546      186:
3546      187:
3546      188:
3546      189:
3546      190: ;
3546      191: memory_find:
3546      192: ;
3546 78     193:          SEI
3547 a000    194:          LDY  #0
3549 8c0ed4 195:          STY  NMIEN
354c 8c0ed2 196:          STY  IRQEN
354f ad01d3 197:          LDA  PORTB
3552 8dc332 198:          sta  mepoint
3555 48     199:          PHA
3556 a2ff    200:          LDX  #$FF
3558 8c01d3 201: .a:          STY  PORTB
355b ad0040 202:          LDA  M_4000
355e 994633 203:          STA  m_buffer,Y
3561 8e0040 204:          STX  M_4000
3564 8a     205:          TXA
3565 994634 206:          STA  m_buffer_2,Y
3568 c8     207:          INY
3569 d0ed    208:          BNE  .a
356b ca     209:          DEX
356c 8e01d3 210:          STX  PORTB
356f 8600    211:          STX  LNFLG
3571 8e0040 212:          STX  M_4000

```

```

3574 8e0080 213:          STX   M_8000
3577 8e00c0 214:          STX   M_C000
357a a200   215:          LDX   #0
357c 8c01d3 216: .b:      STY   PORTB
357f ad0040 217:          LDA   M_4000
3582 c9ff   218:          CMP   #$FF
3584 d005   219:          BNE   .c
3586 e8     220:          INX
3587 8e0040 221:          STX   M_4000
358a 8a     222:          TXA
358b 994634 223: .c:      STA   m_buffer_2,Y
358e c8     224:          INY
358f d0eb   225:          BNE   .b
3591 8ec432 226:          STX   i_t_ram_banks
3594 a201   227:          LDX   #1
3596 8a     228: .d:      TXA
3597 a0ff   229:          LDY   #$FF
3599 d94634 230: .e:      CMP   m_buffer_2,Y
359c f024   231:          BEQ   .g
359e 88     232:          DEY
359f c0ff   233:          CPY   #$FF
35a1 d0f6   234:          BNE   .e
35a3 ea     235:          NOP
35a4 8c01d3 236: .f:      STY   PORTB
35a7 b94633 237:          LDA   m_buffer,Y
35aa 8d0040 238:          STA   M_4000
35ad 88     239:          DEY
35ae c0ff   240:          CPY   #$FF
35b0 d0f2   241:          BNE   .f
35b2 68     242:          PLA
35b3 8d01d3 243:          STA   PORTB
35b6 a9c0   244:          LDA   #$C0
35b8 8d0ed4 245:          STA   NMIEN
35bb a510   246:          LDA   POKMSK
35bd 8d0ed2 247:          STA   IRQEN
35c0 58     248:          CLI
35c1 60     249:          RTS
35c2       250: ;
35c2 98     251: .g:      TYA
35c3 9dc532 252:          STA   i_ram_lookup-1,X
35c6 e8     253:          INX
35c7 e041   254:          CPX   #64+1 ;$41
35c9 90cb   255:          BCC   .d
35cb a0fe   256:          LDY   #$FE
35cd d0d5   257:          BNE   .f
35cf 4cd235 258:          JMP   .h          ; I can do
35d2       259: ;
35d2 204635 260: .h:      JSR   memory_find
35d5 adc432 261:          lda   I_t_ram_banks
35d8 85a0   262:          sta   $a0
35da 60     263:          RTS
35db       264: ;-----
35db       265: ; Buffer Table

```

```

35db      266:
35db 526561 267: sio2sd1_data:      dc.b  "RealDos Build 30"
35eb 28    268:                      dc.b  40
35ec 53494f 269: sio2sd2_data:      dc.b  "SIO2SD Detected!"
35fc      270:                      ds.b  256
36fc      271:
36fc      272: ape_version:      ds.b  512
38fc      273:
38fc      274: i_cmdtab2:      ds.b  512
3afc      275: ;~~~~~
3afc      276: ;Working buffer
3afc      277: ;
3afc      278: ;-----
3afc      279: READRTC:
3afc 710002 280:                      dc.b  $71,$00,$02,$40
3b00 063b  281:                      dc.w  su2.buf
3b02 070080 282:                      dc.b  $07,$00,$80,$00
3b06      283: ;-----
3b06 000000 284: su2.buf:          dc.b  $00,$00,$00,$00
3b0a 000000 285:                      dc.b  $00,$00,$00,$00
3b0e      286:
3b0e      287: Data_buffer:      ds.b  128
3b8e      288:
3b8e      289:
3b8e      290:
3b8e      291:
8000      292:                      .org  $8000
8000      293:
8000      294:
8000      295: show_hardware:
8000      296: ;-----
8000      297: ;Going to peek in and set 50/60 HZ Byte in
8000      298: ;RealDos info.
8000 201d81 299:                      jsr  pal_ntsc
8003      300: ;
8003      301: ;-----
8003      302: ;looking to see if the SmartOS is present!
8003 20d286 303:                      jsr  TEST_SMARTOS
8006      304: ;
8006      305: ;-----
8006      306: ;looking to see if a Multiplexer Slave
8006      307: ;is present!
8006      308:
8006 203183 309:                      jsr  mux_man2
8009      310: ;-----
8009      311: ; testing for the Black Box
8009 206f80 312:                      jsr  BLACK_BOX
800c ad0a93 313:                      lda  pbi_type
800f c900   314:                      cmp  #$00
8011 d00d   315:                      bne  .go
8013      316: ;-----
8013      317: ; Testing for the MIO
8013 20cb83 318:                      jsr  MIO_TEST

```

```

8016 ad0a93 319:          lda  pbi_type
8019 c900  320:          cmp  #$00
801b d003  321:          bne  .go
801d      322:
801d      323: ;-----
801d      324: ; testing for kpi interface
801d 20d080 325:          jsr  kpi_test
8020      326:
8020      327: ;-----
8020      328: ; Testing for the sio2sd
8020      329: .go:
8020 207e87 330:          jsr  sio2sd
8023      331:
8023      332: ;-----
8023      333: ; Testing for the SDRIVE
8023      334: ;
8023 20ae88 335:          jsr  Sdrive
8026 ad0303 336:          lda  $303
8029 c901  337:          cmp  #$01
802b f006  338:          beq  .bypass
802d      339: ;-----
802d      340: ; Testing for the sio2ide
802d      341: ;
802d 200b88 342:          jsr  sio2ide
8030      343: ;-----
8030      344: ; Testing for the sio2usb
8030      345: ;
8030 204082 346:          jsr  sio2usb
8033      347:
8033      348: .bypass:
8033      349: ;-----
8033      350: ;lets see if the is an ape interface
8033      351: ;Be carefule with this on. a low level
8033      352: ;siov call is made.
8033      353: ;since we are looking at the time clock
8033      354: ;function this routine also looks for
8033      355: ;the sio2usb time clock.
8033 201682 356:          jsr  ape_run
8036      357: ;-----
8036      358: ;this will pull info out of the ape
8036      359: ;telling if it is a shareware or
8036      360: ;fully functional Ape device.
8036      361: ;
8036      362: ;
8036 20bc81 363:          jsr  ape_ver
8039      364: ;-----
8039      365: ;Let's see if we have an R: and P: port
8039 209982 366:          jsr  modem_run
803c 20d882 367:          jsr  printer_run
803f      368: ;-----
803f      369: ;Let's see if we are running an emulator
803f 208681 370:          jsr  pc_run
8042      371: ;-----

```



```

8042      372: ;Testing for ram at $d700
8042 206484 373:      jsr   USP
8045      374:
8045      375:
8045 201d8b 376:      jsr   ck_for_rtime
8048      377: ;-----
8048      378: ; Let's See What Kind of OS
8048      379: ; We have here!
8048 20e384 380:      jsr   TEST_OS
804b      381: ;-----
804b 20c98c 382:      jsr   show_computer
804e ae6e80 383:      LDX   hold_stack
8051 9a     384:      TXS
8052      385:
8052 6c0a00 386:      jmp   ($A)
8055      387:
8055 60     388:      rts
8056      389:
8056      390:
8056      391:
8056 a50a   392: get_xsio:      LDA   COMTAB      ; c
8058 38     393:      SEC
8059 e90a   394:      SBC   #low lsio      ; T
805b 8d6c80 395:      STA   XSIO+1      ; T
805e a50b   396:      LDA   COMTAB+1      ; N
8060 e900   397:      SBC   #high lsio      ; S
8062 8d6d80 398:      STA   XSIO+2      ; p
8065 a96c   399:      lda   #$6c      ; h
8067 8d6b80 400:      sta   xsio
806a 60     401:      rts
806b      402:
806b 4c59e4 403: XSIO:      jMP   siov
806e      404:
806e 00     405: hold_stack:   dc.b   0
806f      406:
806f      407:
806f      408:
806f      409: ;
806f      410: ; Checking on the black box if it is present!
806f      411: ;
806f      412: BLACK_BOX
806f 78     413:      SEI
8070 a902   414:      LDa   #2
8072 8dc0d1 415:      STa   BB_Sensel
8075 a010   416:      ldy   #$10
8077      417: .a
8077 a202   418:      ldx   #2
8079 b900d8 419:      LDa   bb_rom,Y
807c ddcc80 420:      cmp   .e,X
807f f007   421:      beq   .b
8081 c8     422:      INY
8082 d0f3   423:      bne   .a
8084 4cc480 424:      jmp   .d

```

```

8087 60 425:      RTS
8088      426: ;
8088      427: .b
8088 c8 428:      INY
8089 ca 429:      DEX
808a 300a 430:     bmi .c
808c b900d8 431:     LDa  bb_rom,Y
808f ddcc80 432:     cmp  .e,X
8092 d0e3 433:     bne  .a
8094 f0f2 434:     beq  .b
8096      435: .c
8096      436:
8096 a902 437:     lda  #G_blackbox
8098 20108a 438:     jsr  set_hw
809b a902 439:     lda  #$02
809d 8d0a93 440:     sta  pbi_type
80a0      441:
80a0 205a8c 442:     jsr  printsi
80a3 426c61 443:     dc.b  "Black Box IS present in sy
80c4      444: ;
80c4      445: .d
80c4 d8 446:      CLD
80c5 a900 447:     LDa  #0
80c7 8dc0d1 448:     STa  BB_Sensel
80ca 58 449:      CLI
80cb 60 450:      RTS
80cc      451: ;
80cc d00829 452: .e      dc.b  $D0,$08,$29,$00
80d0      453: ;
80d0      454: ;-----
80d0      455: ;
80d0      456: ;routine to check for the kpi
80d0      457: ;
80d0      458: kpi_test
80d0 78 459:     sei
80d1 a928 460:     lda  #$28
80d3 8dff d1 461:     sta  PBIBANK ;turn on ram a
80d6 ad01d6 462:     lda  pbi_ram+1
80d9 c943 463:     cmp  #'C'
80db d03e 464:     bne  .no_kpi
80dd ad07d6 465:     lda  pbi_ram+7
80e0 c94b 466:     cmp  #'K' ;as in KProdu
80e2 d037 467:     bne  .no_kpi ;if it ain't
80e4      468:
80e4 a903 469:     lda  #G_kpi
80e6 20108a 470:     jsr  set_hw
80e9      471:
80e9 a903 472:     lda  #$03
80eb 8d0a93 473:     sta  pbi_type
80ee      474:
80ee 205a8c 475:     jsr  printsi
80f1 4b5049 476:     dc.b  "KPI interface IS present i
8116 a900 477:     lda  #$00

```

```

8118 8dff d1 478:          sta  $d1ff      ; added code f
811b      479: .no_kpi
811b 58   480:          cli
811c 60   481:          rts              ;or whicheve
811d      482: ;
811d      483: ;
811d      484: ;-----
811d      485: ;checking for pal/ntsc machines
811d      486: ;
811d      487: ;
811d      488: pal_ntsc
811d ad14 d0 489:          lda  $d014
8120 09f0 490:          ora  #$f0 ;masking high order b
8122 c9ff 491:          cmp  #$ff ;
8124 f003 492:          beq  .ntsc
8126 d030 493:          bne  .pal
8128 60   494:          rts
8129      495: .ntsc
8129 200b 8c 496:          jsr  OS_off
812c a93c 497:          lda  #60
812e 8d91 ff 498:          sta  euro_time
8131 2016 8c 499:          jsr  os_ON
8134 205a 8c 500:          jsr  printsi
8137 4e54 53 501:          dc.b  "NTSC Detected! Setting To
8157 60   502:          RTS
8158      503: .pal
8158 200b 8c 504:          jsr  OS_OFF
815b a932 505:          lda  #50
815d 8d91 ff 506:          sta  euro_time
8160 2016 8c 507:          jsr  OS_ON
8163 205a 8c 508:          jsr  printsi
8166 5041 4c 509:          dc.b  "PAL Detected! Setting To 5
8185 60   510:          RTS
8186      511: ;
8186      512: ;-----
8186      513: ;   Check to see if we are running on an emulator
8186      514: ;
8186      515: pc_run
8186 a000 516:          ldy  #$00
8188 b91a 03 517: loop_1:          lda  $031a,y
818b c948 518:          cmp  #'H' ; WAS THE EDITOR
818d f008 519:          beq  get_vec
818f c8   520:          iny
8190 c8   521:          iny
8191 c8   522:          iny
8192 c024 523:          cpy  #36
8194 d0f2 524:          bne  loop_1
8196 60   525:          rts
8197      526: ;
8197      527: get_vec
8197 a9ff 528:          lda  #$ff
8199 8d61 84 529:          sta  USP_DATA
819c      530:

```

```

819c a908 531:      lda  #G_Emulator
819e 20108a 532:      jsr  set_hw
81a1      533:
81a1 205a8c 534:      jsr  printsi
81a4 504320 535:      dc.b  "PC Emulator Detected!",$9B
81bb 60 536:      RTS
81bc      537: ;
81bc      538: ;
81bc      539: ;-----
81bc      540: Ape_ver
81bc a900 541:      lda  #$00
81be a000 542:      ldy  #$00
81c0      543: .a:
81c0 99fc38 544:      sta  i_cmdtab2,y
81c3 99fc39 545:      sta  i_cmdtab2+256,y
81c6 c8 546:      iny
81c7 c000 547:      cpy  #$00
81c9 d0f5 548:      bne  .a
81cb      549:
81cb      550:
81cb      551:
81cb      552:
81cb a20b 553: make_drive:      ldx  #$0B
81cd bd0a82 554: .a:      lda  i_cmdtab,x
81d0 9d0003 555:      sta  $300,x
81d3 ca 556:      dex
81d4 10f7 557:      bpl  .a
81d6 206b80 558:      jsr  XSIO
81d9 ad0303 559:      lda  $303
81dc c901 560:      cmp  #$01
81de f001 561:      beq  .got_ape
81e0      562:
81e0 60 563:      rts
81e1      564:
81e1      565:
81e1      566:
81e1      567: .got_ape:
81e1      568:
81e1 a911 569:      lda  #G_Ape
81e3 20108a 570:      jsr  set_hw
81e6      571:
81e6      572:      mea  i_cmdtab2,s_ptr
81e6 a9fc ----:      lda  #low i_cmdtab2
81e8 8582 ----:      sta  s_ptr
81ea a938 ----:      lda  #high i_cmdtab2
81ec 8583 ----:      sta  s_ptr + 1
81ee      573:      mea  ape_version,d_ptr
81ee a9fc ----:      lda  #low ape_version
81f0 8584 ----:      sta  d_ptr
81f2 a936 ----:      lda  #high ape_version
81f4 8585 ----:      sta  d_ptr + 1
81f6 20cc92 574:      jsr  to_str
81f9      575:      mea  ape_version,pr_ptr

```

```

81f9 a9fc ----:      lda    #low ape_version
81fb 8588 ----:      sta    pr_ptr
81fd a936 ----:      lda    #high ape_version
81ff 8589 ----:      sta    pr_ptr + 1
8201 20218c 576:      jsr    echos
8204 a99b 577:      lda    #$9b
8206 20418c 578:      jsr    echo
8209 60 579:      rts
820a 580:
820a 31 581: i_cmdtab:      dc.b  $31
820b 05 582:      dc.b  $05
820c 93 583:      dc.b  $93      ;Remote con
820d 40 584:      dc.b  $40
820e fc38 585:      dc.w  i_cmdtab2
8210 00 586:      dc.b  0
8211 00 587:      dc.b  0
8212 0001 588:      dc.w  256
8214 f1 589:      dc.b  $F1      ;f8  ;$f
8215 00 590:      dc.b  $00      ; Remote Co
8216 591:
8216 592:
8216 593:
8216 594:
8216 595: ;~~~~~
8216 596: ; Ape time Source
8216 597: ;~~~~~
8216 598:
8216 599: Ape_run:
8216 206182 600:      jsr    gettime
8219 3024 601:      bmi    .noape      ; e
821b 205a8c 602:      jsr    printsi
821e 417065 603:      dc.b  "Ape or SIO2PC&AspeQt Detec
823e 60 604:      rts      ; e
823f 605:
823f 60 606: .noape:      rts
8240 607:
8240 608: sio2usb:
8240 208b82 609:      jsr    sio2usb.GETTIME      ; s
8243 301b 610:      bmi    .a      ; e
8245 611:
8245 a912 612:      lda    #G_sio2usb
8247 20108a 613:      jsr    set_hw
824a 614:
824a 615:
824a 205a8c 616:      jsr    printsi
824d 53696f 617:      dc.b  "Sio2usb Detected!", $9b, -1
8260 618: .a:
8260 60 619:      rts      ; e
8261 620:
8261 621: ;~~~~~
8261 622: ;
8261 623: ;      Attempt to download time from APE.
8261 624: ;~~~~~

```

```

8261      625:
8261 a20b 626: gettime:      ldx  #$0B          ; s
8263 bd7e82 627: .setloop:    lda  .cmdtab,x        ;
8266 9d0003 628:          sta  $300,x          ;
8269 ca 629:          dex              ;
826a 10f7 630:          bpl  .setloop        ;
826c 18 631:          clc              ; d
826d a50a 632:          lda  comtab          ; c
826f 690d 633:          adc  #dater          ;
8271 8d0403 634:          sta  $304          ;
8274 a50b 635:          lda  comtab+1        ;
8276 6900 636:          adc  #$00          ;
8278 8d0503 637:          sta  $305          ;
827b 4c6b80 638:          jmp  XSIO          ; g
827e      639:
827e 450193 640: .cmdtab:      dc.b  $45,$01,$93,$40,$00,$0
828a      641:
828a      642: ;
828a      643:
828a 00 644: tammys_os:      dc.b  0
828b      645:
828b      646: ;~~~~~
828b      647: ;Attempt to download time from SIO2USB.
828b      648: ;~~~~~
828b      649: sio2usb.GETTIME:
828b a20b 650:          ldx  #$0B
828d bdfc3a 651: .1:          lda  READRTC,X
8290 9d0003 652:          sta  $300,X
8293 ca 653:          DEX
8294 10f7 654:          bpl  .1
8296 4c6b80 655:          jmp  XSIO
8299      656:
8299      657:
8299      658: ;-----
8299      659: ;   Check to see if we have a modem
8299      660: ;
8299      661: modem_run
8299 a000 662:          ldy  #$00
829b b91a03 663: .a:          lda  $031a,y
829e c952 664:          cmp  #'R' ; WAS THE EDITOR
82a0 f008 665:          beq  .b
82a2 c8 666:          iny
82a3 c8 667:          iny
82a4 c8 668:          iny
82a5 c024 669:          cpy  #36
82a7 d0f2 670:          bne  .a
82a9 60 671:          rts
82aa      672: ;
82aa      673: .b:
82aa ad0a93 674:          lda  pbi_type
82ad c901 675:          cmp  #$01
82af f009 676:          beq  .got_mio
82b1 c902 677:          cmp  #$02

```

```

82b3 f002 678:      beq  .got_bb
82b5 d008 679:      bne  .got_nun
82b7      680:
82b7 a91d 681: .got_bb:      lda  #G_blackbox_r
82b9 2c   682:      dc.b $2c
82ba a91c 683: .got_mio:     lda  #G_mio_r
82bc 20108a 684:      jsr  set_hw
82bf      685: .got_nun:
82bf      686:
82bf 205a8c 687:      jsr  printsi
82c2 523a20 688:      dc.b "R: handler Detected",$9B,-
82d7 60    689:      RTS
82d8      690: ;
82d8      691: ;-----
82d8      692: ;   Check to see if we have a modem
82d8      693: ;
82d8      694: printer_run:
82d8 a000 695:      ldy  #$00
82da b91a03 696: .a:      lda  $031a,y
82dd c950 697:      cmp  #'P' ; WAS THE EDITOR
82df f008 698:      beq  .b
82e1 c8    699:      iny
82e2 c8    700:      iny
82e3 c8    701:      iny
82e4 c024 702:      cpy  #36
82e6 d0f2 703:      bne  .a
82e8 60    704:      rts
82e9      705: ;
82e9      706: .b:
82e9 205a8c 707:      jsr  printsi
82ec 503a20 708:      dc.b "P: handler Detected",$9B,-
8301 60    709:      RTS
8302      710: ;
8302      711: ;
8302      712: ;-----
8302      713: ;   Check to see if we have a modem
8302      714: ;
8302      715: Mux_run:
8302 a000 716:      ldy  #$00
8304 b91a03 717: .a:      lda  $031a,y
8307 c94d 718:      cmp  #'M' ; WAS THE EDITOR
8309 f008 719:      beq  .b
830b c8    720:      iny
830c c8    721:      iny
830d c8    722:      iny
830e c024 723:      cpy  #36
8310 d0f2 724:      bne  .a
8312 60    725:      rts
8313      726: ;
8313      727: .b:
8313      728:
8313 a910 729:      lda  #G_Mux
8315 20108a 730:      jsr  set_hw

```

```

8318      731:
8318 205a8c 732:      jsr  printsi
831b 4d3a20 733:      dc.b  "M: handler Detected",$9B,-
8330 60    734:      RTS
8331      735: ;
8331      736: ; prep the multiplexer
8331      737: mux_man2:
8331 a220 738:      ldx  #$20
8333 a903 739:      lda  #3
8335 9d4203 740:      sta  $0342,x
8338 a9c5 741:      lda  # low mdev
833a 9d4403 742:      sta  $0344,x
833d a983 743:      lda  # high mdev
833f 9d4503 744:      sta  $0345,x
8342 a90c 745:      lda  #12
8344 9d4a03 746:      sta  $034a,x
8347 a900 747:      lda  #0
8349 9d4b03 748:      sta  $034b,x
834c 2056e4 749:      jsr  $e456
834f 1003 750:      bpl  cht030
8351 4c8d83 751:      jmp  chterr
8354      752:
8354 209f83 753: cht030:      jsr  xstat
8357 98    754:      tya
8358 8dc983 755:      sta  node
835b 4930 756:      eor  #$30
835d 8dca83 757:      sta  node+1
8360 8d7a83 758:      sta  node1
8363 205a8c 759:      jsr  printsi
8366 9b    760:      dc.b  $9b
8367 4d756c 761:      dc.b  "Multiplexer Slave #"
837a 7820 762: node1:      dc.b  "x "
837c c4e5f4 763:      dc.b  <+128>,"Detected!"
8385 9bff 764:      dc.b  $9B,$FF
8387 a220 765:      ldx  #$20
8389 209683 766:      jsr  cclose
838c 60    767:      rts
838d      768:
838d      769:
838d      770:
838d 98    771: chterr:      tya
838e 48    772:      pha
838f a220 773:      ldx  #$20
8391 20b883 774:      jsr  chtcls
8394 68    775:      pla
8395 60    776:      rts
8396      777: ;
8396      778: cclose:
8396 a90c 779:      lda  #$0c
8398 9d4203 780:      sta  $0342,x
839b 2056e4 781:      jsr  $e456
839e 60    782:      rts
839f      783: ;

```



```

839f      784: ;
839f      785: xstat:
839f a220 786:      ldx  #$20
83a1 a90d 787:      lda  #13
83a3 9d4203 788:      sta  $0342,x
83a6 a900 789:      lda  #$00
83a8 9d4a03 790:      sta  $034a,x
83ab 9d4b03 791:      sta  $034b,x
83ae 2056e4 792:      jsr  $e456
83b1 adeb02 793:      lda  747
83b4 8dc383 794:      sta  1747
83b7 60 795:      rts
83b8      796: ;
83b8      797: ;
83b8      798: chtcls
83b8 a220 799:      ldx  #$20
83ba a90c 800:      lda  #12
83bc 9d4203 801:      sta  $0342,x
83bf 2056e4 802:      jsr  $e456
83c2 60 803:      rts
83c3      804:
83c3      805: ;data areas that are loaded
83c3 0000 806: 1747:      dc.b  $00,$00
83c5 4d3a9b 807: mdev:      dc.b  "M:",$9B,$00
83c9 0000 808: node:      dc.b  0,0
83cb      809: ;
83cb      810: ;
83cb      811: ;
83cb      812:
83cb      813: ;
83cb      814: ;-----
83cb      815: ; This Routen Tests for a MIO in system
83cb      816: ;
83cb      817: ;
83cb      818: MIO_TEST
83cb 8d5c84 819:      sta  mio_acc
83ce 8c5d84 820:      sty  mio_y
83d1 8e5e84 821:      stx  mio_x
83d4 ba 822:      tsx
83d5 8e5f84 823:      stx  mio_stack
83d8 78 824:      sei
83d9      825:
83d9 a514 826:      lda  $14
83db c514 827: tammy:      cmp  $14
83dd f0fc 828:      beq  tammy
83df 38 829:      SEC      ; DISSABLE
83e0 ade0d1 830:      LDa  $d1e0
83e3 8d5a84 831:      STa  M_HOLD
83e6 ade2d1 832:      LDa  $d1e2
83e9 8d5b84 833:      STa  M_HOLD+1
83ec a900 834:      LDa  #$00
83ee 8de0d1 835:      STa  $d1e0
83f1 a920 836:      LDa  #$20

```

```

83f3 8de2d1 837:      STa   $d1e2
83f6 a943 838:      LDa   #'C'
83f8 cd01d6 839:      cmp   pbi_ram+1
83fb d00a 840:      bne   NO_MIO
83fd cd0ad6 841:      cmp   pbi_ram+10
8400 d005 842:      bne   NO_MIO
8402 a901 843:      lda   #$01
8404 8d6084 844:      sta   mio_there
8407      845:
8407      846: NO_MIO
8407 a900 847:      lda   #$00      ; this rese
8409 8de0d1 848:      sta   $d1e0
840c 8de2d1 849:      sta   $d1e2
840f 58 850:      CLI
8410      851:
8410 ad5a84 852:      lda   M_HOLD
8413 8de0d1 853:      sta   $d1e0
8416 ad5b84 854:      lda   M_HOLD+1
8419 8de2d1 855:      sta   $d1e2
841c      856:
841c      857:
841c ae5f84 858:      ldx   mio_stack
841f 9a 859:      txs
8420 ad5c84 860:      lda   mio_acc
8423 ac5d84 861:      ldy   mio_y
8426 ae5e84 862:      ldx   mio_x
8429 ad6084 863:      lda   mio_there
842c c900 864:      cmp   #$00
842e f029 865:      beq   _exit
8430      866:
8430      867:
8430 205a8c 868:      jsr   printsi
8433 494344 869:      dc.b  "ICD MIO Present in System!"
844f      870:
844f a901 871:      lda   #G_mio
8451 20108a 872:      jsr   set_hw
8454 a901 873:      lda   #$01
8456 8d0a93 874:      sta   pbi_type
8459      875:
8459      876: _exit
8459      877:
8459 60 878:      rts
845a      879:
845a      880:
845a 0000 881: M_HOLD:      dc.b  0,0
845c      882:
845c 00 883: mio_acc:      dc.b  0      ; accumulator
845d 00 884: mio_y:      dc.b  0      ; Y register
845e 00 885: mio_x:      dc.b  0      ; x register
845f 00 886: mio_stack:    dc.b  0      ; hold the stack
8460 00 887: mio_there:    dc.b  0      ; holding reg for m
8461      888: ;
8461      889: ;-----

```

```

8461      890: ; Need to test for ram at D7xx
8461      891: ;-----
8461      892: ;
8461 0000 893: USP_DATA:      dc.b  0,0
8463 00   894: OS_Type:      dc.b  0
8464      895: USP:
8464 ad9dd7 896:          LDa  $D79D
8467 8d6284 897:          STa  USP_DATA+1
846a a904 898:          LDa  #$04
846c 8d9dd7 899:          STa  $D79D
846f cd9dd7 900:          cmp  $D79D
8472 d026 901:          bne  NO_USP
8474 ad6284 902:          LDa  USP_DATA+1
8477 8d9dd7 903:          STa  $D79D
847a a9ff 904:          LDa  #$FF
847c 8d6184 905:          STa  USP_DATA
847f 205a8c 906:          jsr  printsi
8482 52616d 907:          dc.b  "Ram Detected at $D7xx ", $9
849a      908: ;
849a 60   909: NO_USP:          RTS
849b      910: ;
849b      911: ;off = 1  on = 0
849b      912: gtia_mux:
849b ad12d0 913:          lda  $d012
849e c901 914:          cmp  #$01
84a0 f01a 915:          beq  gtia_off
84a2 205a8c 916:          jsr  printsi
84a5 4d7578 917:          dc.b  "Mux Disable switch "
84b8 cfce 918:          dc.b  <+128>,"ON"
84ba 9bff 919:          dc.b  $9b,-1
84bc      920: gtia_off:
84bc 60   921:          rts
84bd      922: ;
84bd      923: gtia_usp:
84bd ad12d0 924:          lda  $d012
84c0 c901 925:          cmp  #$01
84c2 f01e 926:          beq  gtia_of
84c4 205a8c 927:          jsr  printsi
84c7 555350 928:          dc.b  "USP booting from ramdisk "
84e0 9bff 929:          dc.b  $9b,-1
84e2      930: gtia_of:
84e2 60   931:          rts
84e3      932: ;
84e3      933: ;
84e3      934: ;
84e3      935: TEST_OS:
84e3      936: ;          jsr  os_on
84e3 adfecf 937:          LDa  $CFFE
84e6 8d8a82 938:          sta  tammys_os
84e9 c910 939:          cmp  #$10          ; x
84eb d032 940:          bne  .USP1
84ed ad5ae4 941:          LDa  $E459+1
84f0 c933 942:          cmp  #$33

```

```

84f2 f028 943:      beq  .USPA
84f4 c970 944:      cmp  #$70          ; g
84f6 f024 945:      beq  .USPa
84f8 a900 946:      LDa  #$00          ; 0
84fa 8d6184 947:      STa  USP_DATA
84fd a901 948:      lda  #$01
84ff 8d6384 949:      sta  os_type
8502 205a8c 950:      jsr  printsi
8505 537461 951:      dc.b  "Standard OS Detected", $9B,
851b 60 952:      RTS
851c 953: ;
851c 4c2686 954: .USPA:      jmp  .USP6
851f 955: ;
851f 956: .USP1
851f c9fe 957:      cmp  #$FE          ; t
8521 d024 958:      bne  .USP2
8523 a900 959:      LDa  #$00          ; 0
8525 8d6184 960:      STa  USP_DATA
8528 a902 961:      lda  #$02
852a 8d6384 962:      sta  os_type
852d 205a8c 963:      jsr  printsi
8530 547572 964:      dc.b  "Turbo OS Detected!", $9B,
8546 60 965:      RTS
8547 966: ;
8547 967: .USP2
8547 c990 968:      cmp  #$90          ; M
8549 d023 969:      bne  .USP3
854b a9ff 970:      LDa  #$FF          ; 0
854d 8d6184 971:      STa  USP_DATA
8550 a980 972:      lda  #$80
8552 8d6384 973:      sta  os_type
8555 209b84 974:      jsr  gtia_mux
8558 205a8c 975:      jsr  printsi
855b 4d7578 976:      dc.b  "Mux OS Detected!", $9B, -1
856d 60 977:      RTS
856e 978: ;
856e 979: .USP3
856e c988 980:      cmp  #$88          ; U
8570 d033 981:      bne  .USP4
8572 a900 982:      LDa  #$00          ; 0
8574 8d6184 983:      STa  USP_DATA
8577 a907 984:      lda  #$07
8579 8d6384 985:      sta  os_type
857c 20cb86 986:      jsr  my_wait
857f 20bd84 987:      jsr  gtia_usp
8582 205a8c 988:      jsr  printsi
8585 556c74 989:      dc.b  "Ultra Speed Plus OS Detect
85a4 60 990: .NNR4:      RTS
85a5 991: ;
85a5 992: .USP4
85a5 993: ;
85a5 c900 994:      cmp  #$00          ; o
85a7 d051 995:      bne  .USP5

```

```

85a9 aca5cd 996:      ldy  $CDA5
85ac c060  997:      cpy  #$60
85ae f024  998:      beq  .TBPLUS
85b0      999:      ;
85b0 a900  1000:     LDa  #$00          ; 0
85b2 8d6184 1001:     STa  USP_DATA
85b5 a904  1002:     lda  #$04
85b7 8d6384 1003:     sta  os_type
85ba 205a8c 1004:     jsr  printsi
85bd 4f6d6e 1005:     dc.b "Omnimon OS Detected!",$9B,
85d3 60    1006:     RTS
85d4      1007:     ;
85d4      1008: .TBPLUS:
85d4 a900  1009:     LDa  #$00          ; 0
85d6 8d6184 1010:     STa  USP_DATA
85d9 a903  1011:     lda  #$03
85db 8d6384 1012:     sta  os_type
85de 205a8c 1013:     jsr  printsi
85e1 547572 1014:     dc.b "TurboS + OS Detected!",$9
85f9 60    1015:     RTS
85fa      1016:     ;
85fa      1017: .USP5
85fa      1018:     ;
85fa c9f3  1019:     cmp  #$F3          ; 0
85fc d025  1020:     bne  .USPAA
85fe a900  1021:     LDa  #$00          ; 0
8600 8d6184 1022:     STa  USP_DATA
8603 a905  1023:     lda  #$05
8605 8d6384 1024:     sta  os_type
8608 205a8c 1025:     jsr  printsi
860b 4f6d6e 1026:     dc.b "OmniView OS Detected!",$9B
8622 60    1027:     RTS
8623      1028:     ;
8623 4cad86 1029: .USPAA:      jmp  .OSDT
8626      1030:     ;
8626      1031: .USP6
8626 ad5ae4 1032:     lda  $e459+1
8629 c970  1033:     cmp  #$70          ; g
862b d027  1034:     bne  .USP7
862d a9ff  1035:     LDa  #$FF          ; 0
862f 8d6184 1036:     STa  USP_DATA
8632 a982  1037:     lda  #$82
8634 8d6384 1038:     sta  os_type
8637 205a8c 1039:     jsr  printsi
863a 4d7920 1040:     dc.b "My IDE 3.1 OS Detected!",$
8653 60    1041:     RTS
8654      1042: .usp7
8654      1043:
8654 ad5ae4 1044:     lda  $e459+1
8657 c98f  1045:     cmp  #$8f          ; g
8659 d027  1046:     bne  .USP8
865b a9ff  1047:     LDa  #$FF          ; 0
865d 8d6184 1048:     STa  USP_DATA

```

```

8660 a981 1049:      lda  #$81
8662 8d6384 1050:      sta  os_type
8665 205a8c 1051:      jsr  printsi
8668 4d7920 1052:      dc.b  "My IDE 4.2 OS Detected!",$
8681 60 1053:      RTS
8682 1054:
8682 1055:
8682 1056:
8682 1057:
8682 1058: .USP8
8682 c908 1059:      cmp  #$08          ; L
8684 d027 1060:      bne  .OSDT
8686 a900 1061:      LDa  #$00          ; 0
8688 8d6184 1062:      STa  USP_DATA
868b a906 1063:      lda  #$06
868d 8d6384 1064:      sta  os_type
8690 205a8c 1065:      jsr  printsi
8693 417065 1066:      dc.b  "Ape Warp + OS Detected!",$
86ac 60 1067:      RTS
86ad 1068: .OSDT
86ad a900 1069:      LDa  #$00
86af 8d6184 1070:      STa  USP_DATA
86b2 205a8c 1071:      jsr  printsi
86b5 556e6b 1072:      dc.b  "Unknown OS Detected", $9B,-
86ca 60 1073:      RTS
86cb 1074: ;
86cb 1075: ;
86cb 1076: ;
86cb 1077: MY_WAIT
86cb a514 1078:      LDa  RTCLOK+2
86cd c514 1079: .A:      cmp  RTCLOK+2
86cf d0fc 1080:      bne  .A
86d1 60 1081:      RTS
86d2 1082: ;
86d2 1083: ;-----
86d2 1084: TEST_SMARTOS:
86d2 ad11d0 1085:      LDa  $D011
86d5 8d7d87 1086:      STa  SMOS_DATA+2
86d8 1087: ;
86d8 20cb86 1088:      jsr  MY_WAIT
86db ad01d3 1089:      LDa  PORTB
86de 8d7c87 1090:      STa  SMOS_DATA+1
86e1 1091: ;
86e1 08 1092:      PHP
86e2 78 1093:      SEI
86e3 ad0ed4 1094:      LDa  NMIEN
86e6 48 1095:      PHA
86e7 a900 1096:      LDa  #$00
86e9 8d0ed4 1097:      STa  NMIEN
86ec 1098: ;
86ec ad7d87 1099:      LDa  SMOS_DATA+2
86ef c901 1100:      cmp  #$01
86f1 f005 1101:      beq  NOT_IN_ROM

```

```

86f3 a940 1102:      LDa  #$40
86f5 8d7b87 1103:      STa  SMOS_DATA
86f8      1104: ;
86f8      1105: NOT_IN_ROM:
86f8 a9a0 1106:      LDa  #$A0
86fa 8d80d3 1107:      STa  SMARTOS
86fd a9ff 1108:      LDa  #$FF
86ff 8d00e0 1109:      STa  Char_Set
8702 ad00e0 1110:      LDa  Char_Set
8705 c9ff 1111:      cmp  #$FF
8707 f00d 1112:      beq  SMOS_HERE
8709      1113: ;
8709 68 1114:      PLA
870a 8d0ed4 1115:      STa  NMIEN
870d 28 1116:      PLP
870e ad7c87 1117:      LDa  SMOS_DATA+1
8711 8d01d3 1118:      STa  PORTB
8714 58 1119:      CLI
8715 60 1120:      RTS
8716      1121: ;
8716      1122: SMOS_HERE:
8716 a900 1123:      LDa  #$00
8718 8d00e0 1124:      STa  char_Set
871b ad7b87 1125:      LDa  SMOS_DATA
871e 8d80d3 1126:      STa  SMARTOS
8721      1127: ;
8721 68 1128:      PLA
8722 8d0ed4 1129:      STa  NMIEN
8725 28 1130:      PLP
8726 ad7c87 1131:      LDa  SMOS_DATA+1
8729 8d01d3 1132:      STa  PORTB
872c 58 1133:      CLI
872d      1134: ;
872d 20cb86 1135:      jsr  MY_WAIT
8730 20cb86 1136:      jsr  MY_WAIT
8733 20cb86 1137:      jsr  MY_WAIT
8736      1138: ;
8736 205a8c 1139:      jsr  printsi
8739 536d61 1140:      dc.b "Smart OS Detected in ",-1
874f ad7b87 1141:      LDa  SMOS_DATA
8752 c940 1142:      cmp  #$40
8754 d00f 1143:      bne  .NOT_IN_ROM
8756 205a8c 1144:      jsr  printsi
8759 526f6d 1145:      dc.b "Rom Mode!", $9b,-1
8764 60 1146:      RTS
8765      1147: ;
8765      1148: .NOT_IN_ROM:
8765 ad7b87 1149:      LDa  SMOS_DATA
8768 c980 1150:      cmp  #$80
876a d00e 1151:      bne  .NOT_IN_RAM
876c 205a8c 1152:      jsr  printsi
876f d2e1ed 1153:      dc.b <+128>,"Ram"
8772 204d6f 1154:      dc.b " Mode!", $9b,-1

```

```

877a      1155: .NOT_IN_RAM:
877a 60    1156:          RTS
877b      1157: ;
877b 800000 1158: SMOS_DATA:      dc.b  $80,0,0
877e      1159: ;
877e      1160: ;
877e      1161: ;=====
877e      1162: ;Sio2sd Detection
877e      1163: ;this Routine sends this text to the display on the sio2sd
877e      1164: ;
877e      1165: ;"RealDos Build 30"
877e      1166: ;"SIO2SD Detected!"
877e      1167: ;=====
877e      1168: sio2sd:
877e      1169: ;
877e a970   1170:          lda  #$70
8780 8d0003 1171:          sta  $300
8783 a903   1172:          lda  #$03
8785 8d0103 1173:          sta  $301
8788 a910   1174:          lda  #$10
878a 8d0203 1175:          sta  $0302
878d a980   1176:          lda  #$80
878f 8d0303 1177:          sta  $0303
8792 a9db   1178:          lda  # low sio2sd1_data
8794 8d0403 1179:          sta  $0304
8797 a935   1180:          lda  # high sio2sd1_data
8799 8d0503 1181:          sta  $0305
879c a928   1182:          lda  #40
879e 8d0803 1183:          sta  $0308
87a1 a900   1184:          lda  #$00
87a3 8d0903 1185:          sta  $0309
87a6 a900   1186:          lda  #0
87a8 8d0a03 1187:          sta  $030a
87ab a900   1188:          lda  #$00
87ad 8d0b03 1189:          sta  $030b
87b0 206b80 1190:          jsr  xsio
87b3 ad0303 1191:          lda  $303
87b6 c901   1192:          cmp  #$01
87b8 f001   1193:          beq  .a
87ba 60     1194:          rts
87bb      1195:
87bb      1196: .a:
87bb      1197:
87bb a913   1198:          lda  #G_sio2sd
87bd 20108a 1199:          jsr  set_hw
87c0      1200:
87c0 205a8c 1201:          jsr  printsi
87c3 53696f 1202:          dc.b  "Sio2sd Detected!",$9b,-1
87d5      1203:
87d5 a970   1204:          lda  #$70
87d7 8d0003 1205:          sta  $300
87da a903   1206:          lda  #$03
87dc 8d0103 1207:          sta  $301

```



```

87df a910 1208:      lda  #$10
87e1 8d0203 1209:      sta  $0302
87e4 a980 1210:      lda  #$80
87e6 8d0303 1211:      sta  $0303
87e9 a9ec 1212:      lda  # low sio2sd2_data
87eb 8d0403 1213:      sta  $0304
87ee a935 1214:      lda  # high sio2sd2_data
87f0 8d0503 1215:      sta  $0305
87f3 a928 1216:      lda  #40
87f5 8d0803 1217:      sta  $0308
87f8 a900 1218:      lda  #$00
87fa 8d0903 1219:      sta  $0309
87fd a901 1220:      lda  #1
87ff 8d0a03 1221:      sta  $030a
8802 a900 1222:      lda  #$00
8804 8d0b03 1223:      sta  $030b
8807 206b80 1224:      jsr  xsio
880a 60 1225:      rts
880b 1226: ;-----
880b 1227: ;Look and See if we have a sio2ide Master
880b 1228: ; and/or slave ide device
880b 1229: ;-----
880b 1230: sio2ide:
880b a901 1231:      lda  #ide_master
880d 8d0103 1232:      sta  $301      ; D
8810 207688 1233:      jsr  get_sio2ide
8813 1234:
8813 206b80 1235:      jsr  xsio
8816 ad0303 1236:      lda  $303
8819 c901 1237:      cmp  #$01
881b f002 1238:      beq  .a
881d d022 1239:      bne  .b
881f 1240:
881f 1241: .a:
881f a914 1242:      lda  #G_sio2idem
8821 20108a 1243:      jsr  set_hw
8824 1244:
8824 205a8c 1245:      jsr  printsi
8827 53696f 1246:      dc.b  "Sio2ide "
882f cde1f3 1247:      dc.b  <+128>,"Master"
8835 204465 1248:      dc.b  " Detected!",$9b
8840 ff 1249:      dc.b  -1
8841 1250: .b:
8841 1251:
8841 a902 1252:      lda  #ide_slave
8843 8d0103 1253:      sta  $301      ; D
8846 207688 1254:      jsr  get_sio2ide
8849 1255:
8849 206b80 1256:      jsr  xsio
884c ad0303 1257:      lda  $303
884f c901 1258:      cmp  #$01
8851 f001 1259:      beq  .aa
8853 60 1260:      rts

```

```

8854      1261:
8854      1262: .aa:
8854 a915  1263:      lda  #G_sio2ides
8856 20108a 1264:      jsr  set_hw
8859 205a8c 1265:      jsr  printsi
885c 53696f 1266:      dc.b "Sio2ide "
8864 d3ece1 1267:      dc.b <+128>,"Slave"
8869 204465 1268:      dc.b " Detected!",$9b
8874 ff    1269:      dc.b -1
8875 60    1270:      rts
8876      1271:
8876      1272:
8876 a971  1273: get_sio2ide:      lda  #pgm_sio2ide
8878 8d0003 1274:      sta  $300      ; D
887b      1275:
887b a911  1276:      lda  #SIOC_GETDSK      ; s
887d 8d0203 1277:      sta  $0302      ; D
8880 a940  1278:      lda  #siov_do      ; $
8882 8d0303 1279:      sta  $0303      ; D
8885      1280:
8885 a90e  1281:      lda  # low Data_buffer
8887 8d0403 1282:      sta  $0304      ; D
888a a93b  1283:      lda  # high Data_buffer
888c 8d0503 1284:      sta  $0305      ; D
888f      1285:
888f a914  1286:      lda  #ide_timeout
8891 8d0603 1287:      sta  $0306      ; T
8894 a900  1288:      lda  #$00
8896 8d0703 1289:      sta  $0307      ; D
8899      1290:
8899      1291:
8899 a90f  1292:      lda  #Low xfersize
889b 8d0803 1293:      sta  $0308      ; N
889e a900  1294:      lda  #high xfersize
88a0 8d0903 1295:      sta  $0309      ; N
88a3      1296:
88a3 a901  1297:      lda  #ide_disk
88a5 8d0a03 1298:      sta  $030a      ; D
88a8 a900  1299:      lda  #$00
88aa 8d0b03 1300:      sta  $030b      ; D
88ad 60    1301:      rts
88ae      1302:
88ae      1303:
88ae      1304: ;=====
88ae      1305: ;
88ae      1306: ;This detect the sdrive
88ae      1307: ;
88ae      1308:
88ae      1309:
88ae      1310: SDrive:
88ae      1311:
88ae a971  1312:      lda  #$71
88b0 8d0003 1313:      sta  $300      ; D

```

```

88b3 a901 1314:      lda  #$01
88b5 8d0103 1315:      sta  $301          ; D
88b8      1316:
88b8 a9e0 1317:      lda  #$e0
88ba 8d0203 1318:      sta  $0302          ; s
88bd a940 1319:      lda  #siov_read
88bf 8d0303 1320:      sta  $0303
88c2 a910 1321:      lda  # low sdrive_data
88c4 8d0403 1322:      sta  $0304          ; D
88c7 a989 1323:      lda  # high sdrive_data
88c9 8d0503 1324:      sta  $0305          ; D
88cc      1325:
88cc a907 1326:      lda  #7
88ce 8d0603 1327:      sta  $0306          ; T
88d1 a900 1328:      lda  #$00
88d3 8d0703 1329:      sta  $0307          ; D
88d6      1330:
88d6 a926 1331:      lda  #low 38
88d8 8d0803 1332:      sta  $0308          ; N
88db a900 1333:      lda  #high 38
88dd 8d0903 1334:      sta  $0309          ; N
88e0 a926 1335:      lda  #low 38
88e2 8d0a03 1336:      sta  $030a          ; D
88e5 a900 1337:      lda  #high 38
88e7 8d0b03 1338:      sta  $030b          ; D
88ea      1339:
88ea 206b80 1340:      jsr  xsio
88ed ad0303 1341:      lda  $303          ; D
88f0 c901 1342:      cmp  #$01
88f2 f001 1343:      beq  .a
88f4 60 1344:      rts
88f5      1345:
88f5 205a8c 1346: .a:      jsr  printsi
88f8 534472 1347:      dc.b  "SDrive Detected!", $9b
8909 ff 1348:      dc.b  -1
890a      1349:
890a a916 1350:      lda  #G_sdrive
890c 20108a 1351:      jsr  set_hw
890f      1352:
890f 60 1353:      rts
8910      1354:
8910      1355:
8910      1356:
8910      1357:
8910      1358: sdrive_data:
8910      1359:      ds.b  256
8a10      1360:
8a10      1361: ;
8a10      1362: ;~~~~~
8a10      1363: ;load the acc with the flag you want to set
8a10      1364: ;   lda  #Value
8a10      1365: ;   jsr  set_hw
8a10      1366: ;~~~~~

```

```

8a10      1367:
8a10      1368: set_hw:
8a10 8d288a 1369:          sta  .hold
8a13 200b8c 1370:          jsr  os_off
8a16      1371:          mea  int.hardware,s_ptr
8a16 a9ed  ----:          lda  #low int.hardware
8a18 8582  ----:          sta  s_ptr
8a1a a9ff  ----:          lda  #high int.hardware
8a1c 8583  ----:          sta  s_ptr + 1
8a1e ad288a 1372:          lda  .hold
8a21 208a8a 1373:          jsr  setlvl
8a24 20168c 1374:          jsr  os_on
8a27 60    1375:          rts
8a28      1376:
8a28 00    1377: .hold:          dc.b  0
8a29      1378:
8a29      1379:
8a29      1380: ;
8a29      1381: ;~~~~~
8a29      1382: ;checks user security level pointed to by S_PTR to see if
8a29      1383: ;bit specified in the A register is ON. Returns MINus if
8a29      1384: ;user doesn't have the specified security level.
8a29      1385: ;~~~~~
8a29      1386:
8a29 8c848a 1387: seclvl:          sty  .tempy
8a2c 8e858a 1388:          stx  .tempx
8a2f c900  1389:          cmp  #0          ;se
8a31 f042  1390:          beq  .passed      ;ye
8a33 a8    1391:          tay          ;-1
8a34 a980  1392:          lda  #128
8a36 8d868a 1393:          sta  .temp          ;in
8a39 a900  1394:          lda  #0
8a3b 8d878a 1395:          sta  .temp+1
8a3e 8d888a 1396:          sta  .temp+2
8a41 8d898a 1397:          sta  .temp+3
8a44 88    1398:          dey
8a45 f011  1399:          beq  .mask          ;if
8a47 4e868a 1400: .shloop:        lsr  .temp          ;sh
8a4a 6e878a 1401:          ror  .temp+1
8a4d 6e888a 1402:          ror  .temp+2
8a50 6e898a 1403:          ror  .temp+3
8a53 88    1404:          dey
8a54 d0f1  1405:          bne  .shloop
8a56 a000  1406:          ldy  #0
8a58 b9868a 1407: .mask:          lda  .temp,y          ;no
8a5b 3182  1408:          and  (s_ptr),y
8a5d 99868a 1409:          sta  .temp,y
8a60 c8    1410:          iny
8a61 c004  1411:          cpy  #4
8a63 d0f3  1412:          bne  .mask
8a65 ad868a 1413: .check:          lda  .temp
8a68 0d878a 1414:          ora  .temp+1          ;i
8a6b 0d888a 1415:          ora  .temp+2

```

```

8a6e 0d898a 1416:      ora  .temp+3
8a71 c900  1417:      cmp  #0
8a73 f004  1418:      beq  .failed
8a75 a900  1419: .passed:  lda  #0
8a77 f002  1420:      beq  .return
8a79 a980  1421: .failed:  lda  #$80
8a7b 08    1422: .return:  php
8a7c ac848a 1423:      ldy  .tempy
8a7f ae858a 1424:      ldx  .tempx
8a82 28    1425:      plp
8a83 60    1426:      rts
8a84      1427:
8a84 00    1428: .tempy:   dc.b  0
8a85 00    1429: .tempx:   dc.b  0
8a86 000000 1430: .temp:    dc.b  0,0,0,0
8a8a      1431:
8a8a      1432:
8a8a      1433: ; sets the bit specified (in the A register) of the se
8a8a      1434: ; level pointed to by S_PTR to a TRUE value.
8a8a      1435: ; ~~~~~
8a8a      1436:
8a8a 8cc68a 1437: setlvl:   sty  .tempy
8a8d 8ec78a 1438:      stx  .tempx
8a90 c900  1439:      cmp  #0          ;se
8a92 f031  1440:      beq  .return      ;ye
8a94 a8    1441:      tay          ;:-
8a95 a980  1442:      lda  #128
8a97 8dc88a 1443:      sta  .temp          ;in
8a9a a900  1444:      lda  #0
8a9c 8dc98a 1445:      sta  .temp+1
8a9f 8dca8a 1446:      sta  .temp+2
8aa2 8dcb8a 1447:      sta  .temp+3
8aa5 88    1448:      dey
8aa6 f011  1449:      beq  .mask          ;if
8aa8 4ec88a 1450: .shloop:  lsr  .temp          ;sh
8aab 6ec98a 1451:      ror  .temp+1
8aae 6eca8a 1452:      ror  .temp+2
8ab1 6ecb8a 1453:      ror  .temp+3
8ab4 88    1454:      dey
8ab5 d0f1  1455:      bne  .shloop
8ab7 a000  1456:      ldy  #0
8ab9 b9c88a 1457: .mask:    lda  .temp,y          ;no
8abc 1182  1458:      ora  (s_ptr),y
8abe 9182  1459:      sta  (s_ptr),y
8ac0 c8    1460:      iny
8ac1 c004  1461:      cpy  #4
8ac3 d0f4  1462:      bne  .mask
8ac5 60    1463: .return:  rts
8ac6      1464:
8ac6 00    1465: .tempy:   dc.b  0
8ac7 00    1466: .tempx:   dc.b  0
8ac8 000000 1467: .temp:    dc.b  0,0,0,0
8acc      1468:

```

```

8acc      1469:
8acc      1470: ;   CLEARS the bit specified (in the A register) of the
8acc      1471: ; security level pointed to by S_PTR to a FALSE value.
8acc      1472: ; ~~~~~
8acc      1473:
8acc 8c178b 1474: clrlvl:   sty   .tempy
8acf 8e188b 1475:           stx   .tempx
8ad2 c900   1476:           cmp   #0           ;sec lvl of zero?
8ad4 f040   1477:           beq   .return      ;yes..just exit (s
8ad6 a8     1478:           tay           ;-1 for shifting
8ad7 a980   1479:           lda   #128
8ad9 8d198b 1480:           sta   .temp        ;initialize mask
8adc a900   1481:           lda   #0
8ade 8d1a8b 1482:           sta   .temp+1
8ae1 8d1b8b 1483:           sta   .temp+2
8ae4 8d1c8b 1484:           sta   .temp+3
8ae7 88     1485:           dey
8ae8 f011   1486:           beq   .mask        ;if it was one, no
8aea 4e198b 1487: .shloop:   lsr   .temp        ;shift bits down t
8aed 6e1a8b 1488:           ror   .temp+1
8af0 6e1b8b 1489:           ror   .temp+2
8af3 6e1c8b 1490:           ror   .temp+3
8af6 88     1491:           dey
8af7 d0f1   1492:           bne   .shloop
8af9 a000   1493:           ldy   #0
8afb b9198b 1494: .mask:     lda   .temp,y      ;now, XOR the bit
8afe 49ff   1495:           eor   #$ff
8b00 99198b 1496:           sta   .temp,y
8b03 c8     1497:           iny
8b04 c004   1498:           cpy   #4
8b06 d0f3   1499:           bne   .mask
8b08 a000   1500:           ldy   #0
8b0a b9198b 1501: .low:      lda   .temp,y      ;now, AND the bit
8b0d 3182   1502:           and   (s_ptr),y
8b0f 9182   1503:           sta   (s_ptr),y
8b11 c8     1504:           iny
8b12 c004   1505:           cpy   #4
8b14 d0f4   1506:           bne   .low
8b16 60     1507: .return:   rts
8b17      1508:
8b17 00     1509: .tempy:    dc.b   0
8b18 00     1510: .tempx:    dc.b   0
8b19 000000 1511: .temp:     dc.b   0,0,0,0
8b1d      1512:
8b1d      1513:
8b1d      1514:
8b1d      1515:
8b1d      1516:
8b1d      1517:
8b1d      1518: ;=====
8b1d      1519:
8b1d      1520: ;   Start of clock operative check
8b1d      1521: ;   -----

```

```

8b1d 1522:
8b1d 1523: ck_for_rtime:
8b1d a960 1524:      lda  #$60          ; W
8b1f 8d1d8b 1525:      sta  ck_for_rtime      ; t
8b22 1526:                      ; r
8b22 a200 1527:      ldx  #0
8b24 20a98b 1528:      jsr  read             ; g
8b27 c93c 1529:      cmp  #60
8b29 b020 1530:      bcs  bsecs           ; i
8b2b 8d658b 1531:      sta  help
8b2e a514 1532:      lda  $14
8b30 18 1533:      clc
8b31 695a 1534:      adc  #90             ; l
8b33 c514 1535: .a:      cmp  $14
8b35 d0fc 1536:      bne  .a
8b37 20a98b 1537:      jsr  read             ; r
8b3a cd658b 1538:      cmp  help
8b3d f00c 1539:      beq  bsecs           ; i
8b3f 38 1540:      sec
8b40 ed658b 1541:      sbc  help             ; m
8b43 b002 1542:      bcs  okc1
8b45 693c 1543:      adc  #60             ; w
8b47 c903 1544: okc1:      cmp  #3
8b49 9050 1545:      bcc  cont1           ; t
8b4b a966 1546: bsecs:      lda  # low msg
8b4d 8d4403 1547:      sta  $344
8b50 a98b 1548:      lda  # high msg
8b52 8d4503 1549:      sta  $345
8b55 a950 1550: domsg:      lda  #80
8b57 8d4803 1551:      sta  $348
8b5a a909 1552:      lda  #9
8b5c 8d4203 1553:      sta  $342
8b5f a200 1554:      ldx  #0
8b61 2056e4 1555:      jsr  $E456
8b64 60 1556:      rts                  ; r
8b65 1557:
8b65 00 1558: help:      dc.b  0
8b66 522d54 1559: msg:      dc.b  "R-Time 8 Clock "
8b75 ceef4 1560:      dc.b  <+128>,"Not Present"
8b80 9b 1561:      dc.b  $9B
8b81 522d54 1562: insta:      dc.b  "R-Time 8 Clock is Present"
8b9b 1563:
8b9b 1564:
8b9b 1565: ;      Continue with initz...
8b9b 1566: ;      -----
8b9b 1567:
8b9b a981 1568: cont1:      lda  # low insta
8b9d 8d4403 1569:      sta  $344
8ba0 a98b 1570:      lda  # high insta
8ba2 8d4503 1571:      sta  $345
8ba5 20558b 1572:      jsr  domsg
8ba8 60 1573:      rts
8ba9 1574:

```

```

8ba9      1575:
8ba9      1576: ;   Read a data register from clock chip
8ba9      1577: ;   -----
8ba9      1578: ; in:
8ba9      1579: ;   X   = register number to read
8ba9      1580: ; out:
8ba9      1581: ;   A   = value of the register
8ba9      1582:
8ba9 8eec8b 1583: read:      stx   .d+1
8bac 8cea8b 1584:      sty   .c+1
8baf acffaf 1585:      ldy   banks           ; g
8bb2      1586:
8bb2 20ef8b 1587:      jsr   waitcl          ; a
8bb5 c008  1588:      CPY   #8
8bb7 d00c  1589:      BNE   .a
8bb9 8eb8d5 1590:      stx   port            ; s
8bbc adb8d5 1591:      lda   port            ; r
8bbf aeb8d5 1592:      ldx   port            ; X
8bc2 18    1593:      clc
8bc3 9009  1594:      bcc   .b
8bc5 8eb9d5 1595: .a:      STX   PORT+1
8bc8 adb9d5 1596:      LDA   PORT+1
8bcb aeb9d5 1597:      LDX   PORT+1
8bce      1598:
8bce 290f  1599: .b:      and   #15
8bd0 8dee8b 1600:      sta   .temp
8bd3 0a    1601:      asl
8bd4 0a    1602:      asl
8bd5 6dee8b 1603:      adc   .temp
8bd8 0a    1604:      asl           ; temp := p
8bd9 8dee8b 1605:      sta   .temp
8bdc 8a    1606:      txa           ; add low n
8bdd 290f  1607:      and   #15
8bdf 6dee8b 1608:      adc   .temp
8be2      1609:
8be2 c010  1610:      cpy   #16           ; current b
8be4 b003  1611:      bcs   .c
8be6 9900d5 1612:      sta   bankd,y       ; restore t
8be9 a000  1613: .c:      ldy   #0           ; restore v
8beb a200  1614: .d:      ldx   #0           ; restore v
8bed 60    1615:      rts
8bee      1616:
8bee      1617:
8bee 00    1618: .temp:      dc.b   0           ; temp for
8bef      1619:
8bef      1620:
8bef      1621:
8bef      1622: ;   Wait until clock chip is NOT busy (with timeout)
8bef      1623: ;   -----
8bef      1624:
8bef a9ff  1625: waitcl:      lda   #-1           ; w
8bf1 48    1626:      pha
8bf2 c008  1627: .a:      CPY   #8           ; Y IS BANK

```



```

8bf4 d005 1628:      BNE .b
8bf6 adb8d5 1629:      lda port
8bf9 b003 1630:      BCS .c      ; JUMP ALWA
8bfb adb9d5 1631: .b:      LDA PORT+1
8bfe 290f 1632: .c:      and #15
8c00 f007 1633:      beq .d
8c02 68 1634:      pla
8c03 38 1635:      sec
8c04 e901 1636:      sbc #1
8c06 48 1637:      pha
8c07 d0e9 1638:      bne .a
8c09 68 1639: .d:      pla
8c0a 60 1640:      rts
8c0b 1641:
8c0b 1642: ;~~~~~
8c0b 1643: ; OS on and off ROUTEEN
8c0b 1644: ;~~~~~
8c0b 1645: ;
8c0b 1646:
8c0b 1647: os_off:
8c0b 78 1648:      SEI
8c0c ad01d3 1649:      LDA PORTB
8c0f 29fe 1650:      AND #$FE
8c11 8d01d3 1651:      STA PORTB
8c14 58 1652:      CLI
8c15 60 1653:      RTS
8c16 1654: ;
8c16 1655: os_on:
8c16 78 1656:      SEI
8c17 ad01d3 1657:      LDA PORTB
8c1a 0901 1658:      ORA #1
8c1c 8d01d3 1659:      STA PORTB
8c1f 58 1660:      CLI
8c20 60 1661:      RTS
8c21 1662:
8c21 1663:
8c21 1664: ; prints: prints string with address in PR_PTR, length by
8c21 1665: ; ~~~~~
8c21 1666:
8c21 1667: prints:
8c21 1668: echos:
8c21 1669: pri2:
8c21 a000 1670:      ldy #0
8c23 b188 1671:      lda (pr_ptr),y      ;ge
8c25 f017 1672:      beq .xpri      ;ex
8c27 8d408c 1673:      sta .maxlth
8c2a c8 1674:      iny
8c2b 8c3f8c 1675:      sty .tempy
8c2e 1676:
8c2e ac3f8c 1677: .pril:      ldy .tempy
8c31 b188 1678:      lda (pr_ptr),y
8c33 20418c 1679:      jsr echo
8c36 ee3f8c 1680: .prnext:      inc .tempy

```

```

8c39 ce408c 1681:          dec    .maxlth
8c3c d0f0  1682:          bne    .pril
8c3e 60    1683: .xpri:          rts
8c3f      1684:
8c3f 01    1685: .tempy:          dc.b  1
8c40 01    1686: .maxlth:         dc.b  1
8c41      1687:
8c41      1688:
8c41      1689:
8c41      1690:
8c41      1691: ;-----
8c41      1692: ; This is a print utility to let
8c41      1693: ; what is going on.
8c41      1694: ;
8c41      1695: ; prints Routeen
8c41      1696: ; this is a slow print to the screen;
8c41      1697: ; usage jsr prints
8c41      1698: ; dc.b $9b," string to be printed", $ff
8c41      1699: ;-----
8c41      1700: ;
8c41      1701: ;
8c41      1702: ;=====
8c41      1703: ;   print/input routines
8c41      1704: ;=====
8c41      1705: ;
8c41      1706: ;   Print character
8c41      1707: ;   -----
8c41      1708: ; in:
8c41      1709: ;   A   = character to print
8c41      1710: ; out:
8c41      1711: ;   all registers preserved
8c41      1712: ;
8c41 208c8c 1713: echo:          JSR    SAVER
8c44 204a8c 1714:                JSR    ZOUT
8c47 4cbc8c 1715:                JMP    RESALL
8c4a      1716: ;
8c4a 8d568c 1717: ZOUT:          STA    ZTEMP+1
8c4d ad4703 1718:                LDA    $0347
8c50 48    1719:                PHa
8c51 ad4603 1720:                LDA    $0346
8c54 48    1721:                PHa
8c55 a900  1722: ZTEMP:         LDA    #0
8c57 a200  1723:                LDX    #0          ;1
8c59 60    1724:                RTS
8c5a      1725: ;
8c5a      1726: ;
8c5a      1727: ;   Print text
8c5a      1728: ;   -----
8c5a      1729: ; out:
8c5a      1730: ;   all registers preserved
8c5a      1731: ; notes:
8c5a      1732: ;   This print routine will print all characters
8c5a      1733: ;   following the JSR PRINT until a delimiter of

```

```

8c5a      1734: ;   -1 ($FF) is reached. PRINT will return to the
8c5a      1735: ;   point one byte beyond the delimiter.
8c5a      1736: ;
8c5a 208c8c 1737: printsi:      JSR   SAVER
8c5d ba    1738:      TSX
8c5e bd0501 1739:      LDA   $0105,X
8c61 8d6d8c 1740:      STA   ZOCH+1
8c64 bd0601 1741:      LDA   $0106,X
8c67 8d6e8c 1742:      STA   ZOCH+2
8c6a      1743: ;
8c6a a001   1744:      ldy   #1
8c6c b9ffff 1745: ZOCH:      LDA   $FFFF,Y
8c6f c9ff   1746:      CMP   #$FF
8c71 f006   1747:      BEQ   Zecho
8c73 20418c 1748:      JSR   echo
8c76 c8     1749:      INY
8c77 d0f3   1750:      BNE   ZOCH
8c79 98     1751: Zecho:     TYA
8c7a 18     1752:      CLC
8c7b 7d0501 1753:      ADC   $0105,X
8c7e 9d0501 1754:      STA   $0105,X
8c81 bd0601 1755:      LDA   $0106,X
8c84 6900   1756:      ADC   #0
8c86 9d0601 1757:      STA   $0106,X
8c89 4cbc8c 1758:      JMP   RESALL
8c8c      1759: ;
8c8c      1760: ;
8c8c      1761: ;   SAVE & RESTORE registers
8c8c      1762: ;   -----
8c8c      1763: ;
8c8c 08     1764: SAVER:     PHP
8c8d 48     1765:      PHA
8c8e 48     1766:      PHA
8c8f 48     1767:      PHA
8c90 08     1768:      PHP
8c91 48     1769:      PHA
8c92 8a     1770:      TXA
8c93 48     1771:      PHA
8c94 ba     1772:      TSX
8c95 bd0901 1773:      LDA   $0109,X
8c98 9d0501 1774:      STA   $0105,X
8c9b bd0701 1775:      LDA   $0107,X
8c9e 9d0901 1776:      STA   $0109,X
8ca1 bd0101 1777:      LDA   $0101,X
8ca4 9d0701 1778:      STA   $0107,X
8ca7 bd0801 1779:      LDA   $0108,X
8caa 9d0401 1780:      STA   $0104,X
8cad bd0601 1781:      LDA   $0106,X
8cb0 9d0801 1782:      STA   $0108,X
8cb3 98     1783:      TYA
8cb4 9d0601 1784:      STA   $0106,X
8cb7 68     1785:      PLA
8cb8 aa     1786:      TAX

```

```

8cb9 68 1787:          PLA
8cba 28 1788:          PLP
8cbb 60 1789:          RTS
8cbc      1790: ;
8cbc 68 1791: RESALL:          PLA
8cbd a8 1792:          TAY
8cbe 68 1793:          PLA
8cbf aa 1794:          TAX
8cc0 68 1795:          PLA
8cc1 28 1796:          PLP
8cc2 60 1797:          RTS
8cc3      1798: ;
8cc3      1799: ;
8cc3 a514 1800: jiffy:      LDA  RTCLOK+2
8cc5 c514 1801: .a:        CMP  RTCLOK+2
8cc7 f0fc 1802:          BEQ  .a
8cc9      1803: ;~~~~~
8cc9      1804: show_computer:
8cc9      1805:
8cc9 205a8c 1806:          JSR  printsi
8ccc 9b436f 1807:          dc.b  $9b,"Computer is: "
8cda ff 1808:          dc.b  -1
8cdb      1809:
8cdb aec432 1810:          ldx  total_ram_banks
8cde e004 1811:          CPX  #4
8ce0 901d 1812:          BCC  L800xl
8ce2 d046 1813:          BNE  L130xe_192
8ce4 a901 1814:          lda  #01
8ce6 205a8c 1815:          JSR  printsi
8ce9 313238 1816:          dc.b  "128K - 130XE (64K)", $9b
8cfd ff 1817:          dc.b  -1
8cfe      1818:
8cfe 60 1819:          rts
8cff      1820: ;
8cff 205a8c 1821: 1800xl:      JSR  printsi
8d02 383030 1822:          dc.b  "800XL unmodified", $9b
8d13 2d2d2d 1823:          dc.b  "---- No RAM Disk----", $9b
8d28 ff 1824:          dc.b  -1
8d29 60 1825:          RTS
8d2a      1826: ;
8d2a e00c 1827: L130xe_192:  CPX  #12
8d2c f01c 1828:          BEQ  1800xl_192
8d2e b038 1829:          BCS  1320xe
8d30 205a8c 1830:          JSR  printsi
8d33 313932 1831:          dc.b  "192K - 130XE (128K)", $9b
8d48 ff 1832:          dc.b  -1
8d49      1833:
8d49 60 1834:          rts
8d4a      1835: ;
8d4a      1836: 1800xl_192:
8d4a 205a8c 1837:          JSR  printsi
8d4d 323536 1838:          dc.b  "256K - 800/1200XL (192K)",
8d66 ff 1839:          dc.b  -1

```

```

8d67 60 1840:          rts
8d68      1841: ;
8d68 e010 1842: 1320xe:          CPX  #16
8d6a d01a 1843:          BNE  11200xl
8d6c 205a8c 1844:          JSR  printsi
8d6f 333230 1845:          dc.b  "320K - 130XE (256K)", $9b
8d84 ff 1846:          dc.b  -1
8d85 60 1847:          rts
8d86      1848: ;
8d86      1849: ;
8d86 e01c 1850: 11200xl:          CPX  #28
8d88 d016 1851:          BNE  1576kxe
8d8a 205a8c 1852:          JSR  printsi
8d8d 313230 1853:          dc.b  "1200xe - (512K)", $9b
8d9e ff 1854:          dc.b  -1
8d9f 60 1855:          rts
8da0      1856: ;
8da0      1857: ;   Gotta fix the 1 meg thing here
8da0      1858: ;
8da0 e020 1859: 1576kxe:          cpx  #32
8da2 d019 1860:          bne  11088kxe
8da4      1861:
8da4 205a8c 1862:          JSR  printsi
8da7 353736 1863:          dc.b  "576K - 130XE (512K)", $9b
8dbb ff 1864:          dc.b  -1
8dbc 60 1865:          rts
8dbd      1866: 11088kxe:
8dbd 205a8c 1867:          JSR  printsi
8dc0 31206d 1868:          dc.b  "1 meg - 130XE (1024K)", $9b
8dd6 ff 1869:          dc.b  -1
8dd7 60 1870:          RTS
8dd8      1871:
8dd8      1872: ; ~~~~~
8dd8      1873: ; help menu section
8dd8      1874: ; ~~~~~
8dd8      1875: help_menu:
8dd8      1876:          .include  notice
92b6      1877:
92b6 60 1878:          rts
92b7      1879: ;
92b7      1880: ; ~~~~~
92b7      1881: ; GET KEY ROUTEEN
92b7      1882: ; this routeen get's one key and return it to the accumlato
92b7      1883: ; ~~~~~
92b7      1884: ;
92b7 86b6 1885: getkey:          stx  $b6
92b9 84b7 1886:          sty  $b7
92bb 20c392 1887:          jsr  .a
92be a6b6 1888:          ldx  $b6
92c0 a4b7 1889:          ldy  $b7
92c2 60 1890:          rts
92c3 ad25e4 1891: .a:          lda  $e420+5
92c6 48 1892:          pha

```

```

92c7 ad24e4 1893:          lda   $e420+4
92ca 48    1894:          pha
92cb 60    1895:          rts
92cc      1896:
92cc      1897:
92cc      1898: ;   Make a length-byte string from an ASCII ($00 termina
92cc      1899: ;   ~~~~~
92cc      1900:
92cc a000 1901: to_str:          ldy   #0
92ce b182 1902: .tfe:          lda   (s_ptr),y      ;fi
92d0 f007 1903:          beq   .gotend
92d2 c900 1904:          cmp   #$00          ;9b
92d4 f003 1905:          beq   .gotend
92d6      1906: ;          cmp   #$9b
92d6      1907: ;          beq   .gotend
92d6 c8    1908:          iny
92d7 d0f5 1909:          bne   .tfe
92d9 8cf492 1910: .gotend:      sty   .lth          ;sa
92dc c000 1911:          cpy   #0
92de f00c 1912:          beq   .rtn
92e0 a000 1913:          ldy   #0
92e2 b182 1914: .loop:          lda   (s_ptr),y
92e4 c8    1915:          iny
92e5 9184 1916:          sta   (d_ptr),y
92e7 ccf492 1917:          cpy   .lth
92ea 90f6 1918:          bcc   .loop
92ec a000 1919: .rtn:          ldy   #0
92ee adf492 1920:          lda   .lth
92f1 9184 1921:          sta   (d_ptr),y
92f3 60    1922:          rts
92f4      1923:
92f4 01    1924: .lth:          dc.b   1
92f5      1925:
92f5      1926:
92f5      1927: ;
92f5      1928: ;~~~~~
92f5      1929: ;   OS on and off ROUTEEN
92f5      1930: ;~~~~~
92f5 200b8c 1931: cleanup:      jsr   os_off
92f8 a900 1932:          lda   #$00
92fa 8dedff 1933:          sta   int.hardware
92fd 8deeff 1934:          sta   int.hardware+1
9300 8defff 1935:          sta   int.hardware+2
9303 8df0ff 1936:          sta   int.hardware+3
9306 20168c 1937:          jsr   os_on
9309 60    1938:          rts
930a      1939:
930a      1940: ;-----
930a      1941: ; 1= mio
930a      1942: ; 2= Black Box
930a      1943: ; 3= kpi
930a      1944: ; 4= IDEa
930a      1945: ;

```

```
930a 00  1946: pbi_type:      dc.b  0
930b      1947: ;=====
930b      1948:
930b      1949:
930b      1950: win_end:    ds.b  0
930b      1951:
930b      1952:
```

End assembly: no errors