


```

0000 1: ;-----
0000 2: ; title 'RamDisk handler for SpartaDOS and Rea
0000 3: ;
0000 4: ; Copyright 2010 Intergrated Logic Systems
0000 5: ; Source Code is Copyright Stephen J. Car
0000 6: ; Please do not share this source!
0000 7: ;-----
0000 8: ; This program is to replace
0000 9: ; rd.com
0000 10: ; With Ram_disk.com
0000 11: ;
0000 12: ; Code History. Init design, August 2005
0000 13: ; last updaed 8/21/2005 sjc
0000 14: ; Added registration code 6/19/2010
0000 15: ;
0000 16: ;
0000 17: ;
0000 18: ; ~~~~~
0000 19: ;-----
0000 20: ; Notes: o This source code MAY NOT be placed for download
0000 21: ; o "mea" is a macro that loads the address of the
0000 22: ; into the pointer specified by the second field.
0000 23: ;-----
0000 24: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 25: ; XASM st and IBM VERSIONS
0000 26: ;-----
0000 27: ;
0000 28: ; File revision history. Variables
0000 29: ;
0000 30: com_file_name: .macro
0000 31: ; dc.b 13,"filename.ext", $9b
0000 32: ; dc.b 12,"Ram_disk.com", $9b
0000 33: ; .endm
0000 34: ;
0000 35: ;
0000 36: ; ~~~~~
0000 37: ; File revision history. Version Number in hex
0000 38: ;
0000 39: file_ver: equ $12
0000 40: ; ~~~~~
0000 41: ; the Month in dec for the first time this revision
0000 42: ; was compiled
0000 43: ;
0000 44: c_month: equ 8
0000 45: ; ~~~~~
0000 46: ; the day in dec for the first time this revision
0000 47: ; was compiled
0000 48: c_day: equ 28
0000 49: ; ~~~~~
0000 50: ; the year in dec for the first time this revision
0000 51: ; was compiled
0000 52: c_year: equ 2012
0000 53: ; ~~~~~

```

```

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

```

```

0000    107:          .include equates
0000    108:          .include globals
0000    109:          .include macros
0000    110:
3100    111:          .org $3100
3100    112: ;
3100    113: win_start:
3100    114: header_info:
3100    115:          .include header ; t
33a7    116:
33a7 202c3d 117:          jsr os_off
33aa    118:          mea tsr,s_ptr
33aa a9f5 ----:      lda #low tsr
33ac 8582 ----:      sta s_ptr
33ae a9ff ----:      lda #high tsr
33b0 8583 ----:      sta s_ptr + 1
33b2 a901 119:          lda #ramdisk
33b4 20893c 120:          jsr seclvl ; c
33b7 3051 121:          bmi .aa
33b9    122:
33b9 20373d 123:          jsr os_on
33bc 201938 124:          jsr printsi
33bf 9b 125:          dc.b $9b
33c0 202020 126:          dc.b " "
33c3 bdbdbe 127:          dc.b <+128>,"==>"
33c6 a0d2e1 128:          dc.b <+128>," Ramdisk already Pr
33e0 bcbdbd 129:          dc.b <+128>,"<=="
33e3 9b 130:          dc.b $9b
33e4 202020 131:          dc.b " "
33e7 bdbdbe 132:          dc.b <+128>,"==>"
33ea c1ee4 133:          dc.b <+128>,"And Registered with
3404 bcbdbd 134:          dc.b <+128>,"<=="
3407 9bff 135:          dc.b $9b,-1
3409 60 136:          rts ; w
340a    137:
340a    138: .aa:
340a    139: bank_finder:
340a ad01d3 140:          lda PORTB
340d 48 141:          pha
340e a900 142:          lda #0
3410 8d2f02 143:          sta SDMCTL
3413 20c536 144:          jsr jiffy
3416 a21f 145:          ldx #31 ;$1F
3418 bda939 146: .a:          lda ram_lookup,X
341b 4d01d3 147:          eor PORTB
341e 297e 148:          and #$7E
3420 4d01d3 149:          eor PORTB
3423 8d01d3 150:          sta PORTB
3426 ad0070 151:          lda $7000
3429 9dca3a 152:          sta ram_ref,X
342c bda939 153:          lda ram_lookup,X
342f 8d0070 154:          sta $7000
3432 ad0170 155: .b:          lda $7001

```

```

3435 9dea3a 156:      sta  Reference,X
3438 bda939 157:      lda  ram_lookup,X
343b 0a    158:      asl
343c 8d0170 159:      sta  $7001
343f ca    160:      DEX
3440 10d6  161:      bpl  .a
3442 a9ff  162:      lda  #$FF
3444 8d0030 163:      sta  $3000
3447 a9ee  164:      lda  #$EE
3449 8d0130 165:      sta  $3001
344c e8    166:      .bb:      INX
344d bda939 167:      lda  ram_lookup,X
3450 4d01d3 168:      eor  PORTB
3453 297e  169:      and  #$7E
3455 4d01d3 170:      eor  PORTB
3458 8d01d3 171:      sta  PORTB
345b bda939 172:      lda  ram_lookup,X
345e cd0070 173:      cmp  $7000
3461 d00e  174:      bne  .c
3463 bda939 175:      lda  ram_lookup,X
3466 0a    176:      asl
3467 cd0170 177:      cmp  $7001
346a d005  178:      bne  .c
346c e01f  179:      cpx  #$1F
346e d0dc  180:      bne  .bb
3470 e8    181:      INX
3471 8e9b39 182:      .c:      stx  total_ram_banks
3474 a200  183:      ldx  #0
3476 bda939 184:      .d:      lda  ram_lookup,X
3479 4d01d3 185:      eor  PORTB
347c 297e  186:      and  #$7E
347e 4d01d3 187:      eor  PORTB
3481 8d01d3 188:      sta  PORTB
3484 bdca3a 189:      lda  ram_ref,X
3487 8d0070 190:      sta  $7000
348a bdea3a 191:      lda  Reference,X
348d 8d0170 192:      sta  $7001
3490 e8    193:      INX
3491 e020  194:      cpx  #$20
3493 d0e1  195:      bne  .d
3495 68    196:      PLA
3496 8d01d3 197:      sta  PORTB
3499 a922  198:      lda  #$22
349b 8d2f02 199:      sta  SDMCTL
349e 20c536 200:      jsr  jiffy
34a1      201:
34a1      202:
34a1      203:
34a1      204:
34a1      205:      ;      jsr  printsi
34a1      206:      ;      dc.b  "Ramdisk Ver 1.0 9-10-200
34a1      207:      ;      dc.b  $9b
34a1      208:      ;      dc.b  "Copyright 2010 ILS"

```

```

34a1 209: ;          dc.b  " "
34a1 210: ;          dc.b  $9b,$9b
34a1 211: ;          dc.b  -1
34a1 212: Show_machine:
34a1 213:
34a1 201938 214:      jsr  printsi
34a4 9b9b 215:      dc.b  $9b,$9b
34a6 216:
34a6 436f6d 217:      dc.b  "Computer is: "
34b3 ff 218:      dc.b  -1
34b4 219:
34b4 ae9b39 220:      ldx  total_ram_banks
34b7 e004 221:      cpx  #4
34b9 901d 222:      bcc  .a
34bb d04a 223:      bne  .b
34bd 201938 224:      jsr  printsi
34c0 313238 225:      dc.b  "128K - 130XE (64K)", $9b
34d4 ff 226:      dc.b  -1
34d5 227:
34d5 4c8735 228:      jmp  look_for_drv_num
34d8 229: ;
34d8 201938 230: .a:      jsr  printsi
34db 383030 231:      dc.b  "800XL unmodified", $9b
34ec 2d2d20 232:      dc.b  "-- No RAM Disk Installed",
3505 ff 233:      dc.b  -1
3506 60 234:      RTS
3507 235: ;
3507 e00c 236: .b:      cpx  #$0C
3509 f01e 237:      beq  .c
350b b03c 238:      bcs  .d
350d 201938 239:      jsr  printsi
3510 313932 240:      dc.b  "192K - 130XE (128K)", $9b
3525 ff 241:      dc.b  -1
3526 242:
3526 4c8735 243:      jmp  look_for_drv_num
3529 244: ;
3529 201938 245: .c:      jsr  printsi
352c 323536 246:      dc.b  "256K - 800/1200XL (192K)",
3545 ff 247:      dc.b  -1
3546 248:
3546 4c8735 249:      jmp  look_for_drv_num
3549 250: ;
3549 e010 251: .d:      cpx  #$10
354b d01c 252:      bne  .e
354d 201938 253:      jsr  printsi
3550 333230 254:      dc.b  "320K - 130XE (256K)", $9b
3565 ff 255:      dc.b  -1
3566 256:
3566 257:
3566 4c8735 258:      jmp  look_for_drv_num
3569 259: ;
3569 eec939 260: .e:      inc  lotta_ram
356c eec939 261:      inc  lotta_ram

```

```

356f 201938 262:      jsr  printsi
3572 353736 263:      dc.b  "576K - 130XE (512K)", $9b
3586 ff      264:      dc.b  -1
3587          265:
3587          266:
3587          267:
3587 a00a 268: look_for_drv_num:  ldy  # bufoff
3589 b10a 269:      lda  (comtab), Y
358b 18     270:      CLC
358c 693f 271:      adc  #lbuf
358e a8     272:      TAY
358f b10a 273:      lda  (comtab), Y
3591 c99b 274:      cmp  #$9B
3593 d027 275:      bne  read_cli
3595          276:
3595 201938 277:      jsr  printsi
3598 2d2d20 278:      dc.b  "-- No Drive Number Specifi
35b5 ff      279:      dc.b  -1
35b6 60     280:      RTS
35b7          281: ;
35b7 4cffff 282: I_ZCRNAME:      JMP  $ffff ;8ae
35ba          283: ;
35ba ff      284: no.format:      dc.b  -1
35bb 00     285: L32DC:          dc.b  $00
35bc          286:
35bc          287:
35bc a50a 288: read_cli:      lda  comtab
35be 18     289:      CLC
35bf 6903 290:      adc  # low ZCRNAME
35c1 8db835 291:      sta  I_ZCRNAME+1
35c4 a50b 292:      lda  comtab+1
35c6 6900 293:      adc  # high ZCRNAME
35c8 8db935 294:      sta  I_ZCRNAME+2
35cb 20b735 295:      jsr  I_ZCRNAME
35ce d032 296:      bne  .parm.error
35d0 a022 297:      ldy  #COMFNAM+1
35d2 b10a 298:      lda  (comtab), Y
35d4 290f 299:      and  #$0F
35d6 8d0507 300:      sta  ram_drive
35d9 20b735 301:      jsr  I_ZCRNAME
35dc f03c 302:      beq  .y
35de a024 303:      ldy  #COMFNAM+3
35e0 b10a 304:      lda  (comtab), Y
35e2 c92f 305:      cmp  #'/'      ;$2F
35e4 d01c 306:      bne  .parm.error
35e6 c8     307: .a:      INY
35e7 b10a 308:      lda  (comtab), Y
35e9 c99b 309:      cmp  #$9B
35eb f02d 310:      beq  .y
35ed c945 311:      cmp  #'E'      ;$45
35ef d007 312:      bne  .bb
35f1 a904 313:      lda  #4
35f3 8d9c39 314:      sta  reserve_ram

```

```

35f6 d0ee 315:      bne .a
35f8 c94e 316: .bb:      cmp #'N'      ;$4E
35fa d006 317:      bne .parm.error
35fc ee3a35 318:      inc no.format
35ff 4ce635 319:      jmp .a
3602      320: ;
3602 201938 321: .parm.error:      jsr printsi
3605 2d2d20 322:      dc.b "-- Parameter Error",$9b
3618 ff 323:      dc.b -1
3619 60 324:      RTS
361a      325: ;
361a a50c 326: .y:      lda DOSINI
361c 8d5038 327:      sta initz+1
361f a50d 328:      lda DOSINI+1
3621 8d5138 329:      sta initz+2
3624 ad9c39 330:      lda reserve_ram
3627 f051 331:      beq install.drv
3629 ad9b39 332:      lda total_ram_banks
362c c904 333:      cmp #4
362e d027 334:      bne .z
3630 201938 335:      jsr printsi
3633 457272 336:      dc.b "Error -- No Extra RAM For
3655 ff 337:      dc.b -1
3656 60 338:      RTS
3657      339: ;
3657 201938 340: .z:      jsr printsi
365a 3c5241 341:      dc.b "<RAMDISK Uses 64K Less Mem
3679 ff 342:      dc.b -1
367a      343:
367a      344:
367a      345: install.drv:
367a      346:
367a      347:
367a      348:
367a ade702 349:      lda memlo      ; s
367d 8d153c 350:      sta segtab+dorgadr      ; P
3680 8d1b3c 351:      sta segtab+blkdes      ; P
3683 ade802 352:      lda memlo+1      ; s
3686 8d163c 353:      sta segtab+dorgadr+1      ; P
3689 8d1c3c 354:      sta segtab+blkdes+1      ; P
368c      355:
368c 204a3b 356:      jsr rlocate      ; o
368f      357:
368f 20623c 358:      jsr set_ramdisk      ;ad
3692      359:
3692 a94f 360: lb01:      lda # low initz
3694 850c 361:      sta DOSINI
3696 a938 362: hb01:      lda # high initz
3698 850d 363:      sta DOSINI+1
369a 38 364:      SEC
369b a50a 365:      lda comtab
369d e90a 366:      sbc # low LSIO
369f 8532 367:      sta BUFRLO

```



```

36a1 a50b 368:      lda  comtab+1
36a3 e900 369:      sbc  # high LSIO
36a5 8533 370:      sta  BUFRLO+1
36a7 a000 371:      ldy  #0
36a9 b132 372:      lda  (BUFRLO),Y
36ab 8d6d38 373: w01:      sta  dos_xiov+1
36ae a95d 374: lb02:      lda  # low ram_siov
36b0 9132 375:      sta  (BUFRLO),Y
36b2 c8 376:      INY
36b3 b132 377:      lda  (BUFRLO),Y
36b5 8d6e38 378: w02:      sta  dos_xiov+2
36b8 a938 379: hb02      lda  # high ram_siov
36ba 9132 380:      sta  (BUFRLO),Y
36bc 205238 381: w03:      jsr  L356D
36bf 2cba35 382:      bit  no.format
36c2 3008 383:      bmi  show.fmt
36c4 60 384:      RTS
36c5      385: ;
36c5 a514 386: jiffy:      lda  RTCLOCK+2
36c7 c514 387: .a:      cmp  RTCLOCK+2
36c9 f0fc 388:      beq  .a
36cb 60 389:      RTS
36cc      390: ;
36cc 201938 391: show.fmt:      jsr  printsi
36cf 202020 392:      dc.b  " <RAM Disk Formatted SD>
36ea ff 393:      dc.b  -1
36eb      394:
36eb ad01d3 395: format.sd:      lda  PORTB
36ee 48 396:      PHA
36ef ae9c39 397:      ldx  reserve_ram
36f2 29fd 398:      and  #$FD
36f4 5da939 399:      eor  ram_lookup,X
36f7 2dc939 400:      and  lotta_ram
36fa 4d01d3 401:      eor  PORTB
36fd 8d01d3 402:      sta  PORTB
3700 a900 403:      lda  #0
3702 a8 404:      TAY
3703      405: .a:
3703 990040 406:      STA  ram_window,Y
3706 990041 407:      STA  ram_window+256,Y
3709 990042 408:      STA  ram_window+512,Y
370c 990043 409:      STA  ram_window+768,Y
370f c8 410:      INY
3710 d0f1 411:      bne  .a
3712      412:
3712      413:
3712 a02a 414:      ldy  #42
3714 b9ee37 415: .b:      lda  boot_sector,Y
3717 990040 416:      sta  ram_window,Y
371a 88 417:      DEY
371b 10f7 418: .c:      bpl  .b
371d ad9b39 419:      lda  total_ram_banks
3720 38 420:      SEC

```

```

3721 ed9c39 421:      sbc   reserve_ram
3724 aa      422:      TAX
3725 8ebb35 423:      stx   L32DC
3728 a000 424:      ldy   #0
372a bd9937 425: .d:      lda   .j+1,X
372d 990b40 426:      sta   ram_window+11,Y
3730 e8      427:      INX
3731 c8      428:      INY
3732 c004 429:      cpy   #4
3734 d0f4 430:      bne   .d
3736 adbb35 431:      lda   L32DC
3739 4a      432:      lsr
373a 4a      433:      lsr
373b aa      434:      TAX
373c 209037 435: .e:      jsr   .h
373f ad9537 436:      lda   .i+1
3742 18      437:      CLC
3743 6940 438:      adc   #$40
3745 8d9537 439:      sta   .i+1
3748 ad9637 440:      lda   .i+2
374b 6900 441:      adc   #0
374d 8d9637 442:      sta   .i+2
3750 ca      443:      DEX
3751 d0e9 444:      bne   .e
3753 aebb35 445:      ldx   L32DC
3756 bdb937 446:      lda   L34D4,X
3759 8d0f40 447:      sta   ram_window+15
375c 18      448:      CLC
375d 6902 449:      adc   #2
375f 8d0940 450:      sta   ram_window+9
3762 bdba37 451:      lda   L34D5,X
3765 8d8040 452:      sta   ram_window+128
3768 bdbb37 453: .f:      lda   L34D6,X
376b 8532 454:      sta   BUFRLO
376d bdbc37 455:      lda   L34D7,X
3770 8533 456:      sta   BUFRLO+1
3772 18      457:      CLC
3773 a004 458:      ldy   #4
3775 ad0940 459:      lda   ram_window+9
3778 6901 460:      adc   #1
377a 9132 461:      sta   (BUFRLO),Y
377c a200 462:      ldx   #0
377e a080 463:      ldy   #$80
3780 bddd37 464: .g:      lda   r_directory,X
3783 9132 465:      sta   (BUFRLO),Y
3785 e8      466:      INX
3786 c8      467:      INY
3787 e011 468:      cpx   #$11
3789 d0f5 469:      bne   .g
378b 68      470:      PLA
378c 8d01d3 471:      sta   PORTB
378f 60      472:      RTS
3790      473: ;

```

```

3790 a000 474: .h:      ldy  #0
3792 a9ff 475:      lda  #$FF
3794 998040 476: .i:      sta  ram_window+128,Y
3797 c8 477:      INY
3798 c040 478: .j:      cpy  #$40
379a d0f8 479:      bne  .i
379c 60 480:      RTS
379d 481: ;
379d 482: 134b8:
379d 0002fb 483:      dc.b  $00,$02,$FB,$01,$00,$04,$FB
37a5 0006fa 484:      dc.b  $00,$06,$FA,$05,$00,$08,$FA
37ad 000af9 485:      dc.b  $00,$0A,$F9,$09,$00,$0C,$F9
37b5 000ef8 486:      dc.b  $00,$0E,$F8,$0D
37b9 487:
37b9 00 488: L34D4:      dc.b  $00
37ba 10 489: L34D5:      dc.b  $10
37bb f8 490: 134d6:      dc.b  $f8
37bc 0f 491: L34D7:      dc.b  $0F
37bd 492:
37bd 010700 493:      dc.b  $01,$07,$00,$41,$01,$07,$00
37c5 020380 494:      dc.b  $02,$03,$80,$41,$02,$03,$80
37cd 030100 495:      dc.b  $03,$01,$00,$42,$03,$01,$00
37d5 040080 496:      dc.b  $04,$00,$80,$42,$04,$00,$80
37dd 497: r_directory:
37dd 080000 498:      dc.b  $08,$00,$00,$11,$00,$00
37e3 4d4149 499:      dc.b  "MAIN  "
37ee 500:
37ee 501: boot_sector:
37ee 000300 502:      dc.b  $00,$03,$00,$07,$e0,$07,$4c
37f5 803003 503:      dc.b  $80,$30,$03,$00,$00,$04,$fb
37fd 010200 504:      dc.b  $01,$02,$00,$20,$00,$00,$00
3804 52414d 505:      dc.b  "RAM-DISK",$00
380d 802200 506:      dc.b  $80,$22,$00,$00,$00,$00,$00
3815 160000 507:      dc.b  $16,$00,$00,$00
3819 508:
3819 509:
3819 68 510: printsi:      PLA
381a 8d2a38 511:      sta  .b+1
381d 68 512:      PLA
381e 8d2b38 513:      sta  .b+2
3821 ee2a38 514: .a:      inc  .b+1
3824 d003 515:      bne  .b
3826 ee2b38 516:      inc  .b+2
3829 adffff 517: .b:      lda  $FFFF
382c c9ff 518:      cmp  #$FF
382e f006 519:      beq  .c
3830 203f38 520:      jsr  echo
3833 4c2138 521:      jmp  .a
3836 522: ;
3836 ad2b38 523: .c:      lda  .b+2
3839 48 524:      PHA
383a ad2a38 525:      lda  .b+1
383d 48 526:      PHA

```

```

383e 60 527:          RTS
383f 528: ;
383f a200 529: echo:      ldx  #0
3841 8e4803 530:          stx  ICBLL
3844 8e4903 531:          stx  ICBLL+1
3847 a00b 532:          ldy  #$0B
3849 8c4203 533:          sty  ICCOM
384c 4c56e4 534:          jmp  CIOV
384f 535: ;
384f 200000 536: initz:      jsr  $0000
3852 537:
3852 538:
3852 539: L356D:
3852 a9ca 540: lb03:      lda  # low zend
3854 8de702 541:          sta  MEMLO
3857 a93a 542: hb03:      lda  # high zend
3859 8de802 543:          sta  MEMLO+1
385c 60 544:          RTS
385d 545: ;
385d 546: ram_siov:
385d ad0003 547:          lda  DDEVIC
3860 c931 548:          cmp  #$31
3862 d008 549:          bne  dos_xiov
3864 ad0103 550:          lda  DUNIT
3867 cd0507 551:          cmp  ram_drive
386a f003 552:          beq  ram_driver
386c 4c0000 553: dos_xiov:   jmp  $0000
386f 554:
386f 555:
386f ae01d3 556: ram_driver: ldx  PORTB
3872 78 557:          SEI
3873 a900 558:          lda  #0
3875 8d0ed4 559:          sta  NMIE
3878 ad0403 560:          lda  DBUFLO
387b 8534 561:          sta  BFENLO
387d ad0503 562:          lda  DBUFLO+1
3880 8535 563:          sta  BFENLO+1
3882 ad0203 564:          lda  DCOMND
3885 c953 565:          cmp  #'S'      ;$53
3887 d01b 566:          bne  ck_rc
3889 ada439 567: w09:      lda  byte.sector.low
388c 4a 568:          lsr
388d 4a 569:          lsr
388e 4930 570:          eor  #48      ;$30
3890 aca039 571: w10:      ldy  sectors_per_trac
3893 c01a 572:          cpy  #26      ;$1A
3895 d002 573:          bne  .a
3897 0980 574:          ORA  #$80
3899 a000 575: .a:      ldy  #0
389b 9134 576:          sta  (BFENLO),Y
389d c8 577:          INY
389e a9ff 578:          lda  #$FF
38a0 9134 579:          sta  (BFENLO),Y

```

```

38a2 3022 580:          bmi  show_stat
38a4 c94e 581: ck_rc:      cmp   #'N'
38a6 d00c 582:          bne   ck_set_cfg
38a8 a00b 583:          ldy   #$0B
38aa      584: ck_rc.a:
38aa b99d39 585: w11:      lda   percom_set,Y
38ad 9134 586:          sta   (BFENLO),Y
38af 88 587:          DEY
38b0 10f8 588:          bpl   ck_rc.a
38b2 3012 589:          bmi   show_stat
38b4 c94f 590: ck_set_cfg: cmp   #'O'
38b6 d011 591:          bne   ck_fmt
38b8 a007 592:          ldy   #7
38ba b134 593:          lda   (BFENLO),Y
38bc 8da439 594: w12:      sta   byte.sector.low
38bf a003 595:          ldy   #3
38c1 b134 596:          lda   (BFENLO),Y
38c3 8da039 597: w13:      sta   sectors_per_trac
38c6      598: show_stat:
38c6 4c8b39 599: w14:      jmp   d_status
38c9      600: ;
38c9 c921 601: ck_fmt:      cmp   #'!'
38cb d00b 602:          bne   ck_fdd
38cd a000 603: format_sectors: ldy   #0
38cf a9ff 604:          lda   #$FF
38d1 9134 605:          sta   (BFENLO),Y
38d3 c8 606:          INY
38d4 9134 607:          sta   (BFENLO),Y
38d6 d0ee 608:          bne   show_stat
38d8      609:
38d8      610:
38d8 c922 611: ck_fdd:      cmp   #$22      ;""
38da d00c 612:          bne   sector_rw
38dc a91a 613:          lda   #26
38de 8da039 614: w15:      sta   sectors_per_trac
38e1 a980 615:          lda   #$80
38e3 8da439 616: w16:      sta   byte.sector.low
38e6 d0e5 617:          bne   format_sectors
38e8      618:
38e8      619:
38e8      620:
38e8 c952 621: sector_rw:   cmp   #'R'
38ea f00b 622:          beq   Read.write_sec
38ec c957 623:          cmp   #'W'
38ee f007 624:          beq   Read.write_sec
38f0 c950 625:          cmp   #'P'
38f2 f003 626:          beq   Read.write_sec
38f4 4c8e39 627: w20:      jmp   error_139
38f7      628: ;
38f7 a900 629: Read.write_sec: lda   #0
38f9 8532 630:          sta   BUFRLO
38fb ad0a03 631:          lda   DAUX1
38fe 38 632:          SEC

```

```

38ff e901 633:      sbc  #1
3901 8533 634:      sta  BUFRLO+1
3903 ad0b03 635:      lda  DAUX1+1
3906 e900 636:      sbc  #0
3908 0633 637:      ASL  BUFRLO+1
390a 2a 638:      rol
390b 2c0803 639:      bit  DBYTLO
390e 3006 640:      bmi  rw.a
3910 0ea439 641: w17:      ASL  byte.sector.low
3913 0633 642:      ASL  BUFRLO+1
3915 2a 643:      rol
3916 48 644: rw.a:      PHA
3917 a533 645:      lda  BUFRLO+1
3919 4a 646:      lsr
391a 4a 647:      lsr
391b 6632 648:      ROR  BUFRLO
391d 0940 649:      ORA  #$40
391f 8533 650:      sta  BUFRLO+1
3921 8a 651:      TXA
3922 29fe 652:      and  #$FE
3924 8d01d3 653:      sta  PORTB
3927 ad0203 654:      lda  DCOMND
392a c952 655:      cmp  #'R'      ;$52
392c f00d 656:      beq  rw.c
392e a000 657:      ldy  #0
3930 b134 658: rw.b:      lda  (BFENLO),Y
3932 99ca39 659: w05:      sta  rd_buffer,Y
3935 c8 660:      INY
3936 cc0803 661:      cpy  DBYTLO
3939 d0f5 662:      bne  rw.b
393b 68 663: rw.c:      PLA
393c 18 664:      CLC
393d 6d9c39 665: w21:      adc  reserve_ram
3940 cd9b39 666: w18:      cmp  total_ram_banks
3943 b049 667:      bcs  error_139
3945 668:
3945 669:
3945 a8 670:      TAY
3946 ad01d3 671:      lda  PORTB
3949 29fd 672:      and  #$FD
394b 59a939 673: w19:      eor  ram_lookup,Y
394e 2dc939 674: w22:      and  lotta_ram
3951 4d01d3 675:      eor  PORTB
3954 8d01d3 676:      sta  PORTB
3957 677:
3957 678:
3957 679:
3957 a000 680:      ldy  #0
3959 ad0203 681:      lda  DCOMND
395c c952 682:      cmp  #'R'      ;$52
395e d020 683:      bne  rw.f
3960 b132 684: rw.d:      lda  (BUFRLO),Y
3962 99ca39 685: w06:      sta  rd_buffer,Y

```

```

3965 c8 686:      INY
3966 cc0803 687:      cpy  DBYTLO
3969 d0f5 688:      bne  rw.d
396b 8a 689:      TXA
396c 29fe 690:      and  #$FE
396e 8d01d3 691:      sta  PORTB
3971 a000 692:      ldy  #0
3973      693: rw.e:
3973 b9ca39 694: w08:      lda  rd_buffer,Y
3976 9134 695:      sta  (BFENLO),Y
3978 c8 696:      INY
3979 cc0803 697:      cpy  DBYTLO
397c d0f5 698:      bne  rw.e
397e f00b 699:      beq  d_status
3980      700: rw.f:
3980 b9ca39 701: w07:      lda  rd_buffer,Y
3983 9132 702:      sta  (BUFRLO),Y
3985 c8 703:      INY
3986 cc0803 704:      cpy  DBYTLO
3989 d0f5 705:      bne  rw.f
398b a001 706: d_status:      ldy  #1
398d 2c 707:      dc.b  $2c
398e a08b 708: error_139:      ldy  #139
3990 8e01d3 709:      stx  PORTB
3993 a9e0 710:      lda  #$E0
3995 8d0ed4 711:      sta  NMEN
3998 58 712:      CLI
3999 98 713:      TYA
399a 60 714:      RTS
399b      715: ;
399b 00 716: total_ram_banks:      dc.b  $00
399c 00 717: reserve_ram:      dc.b  $00
399d      718:
399d      719:
399d 28 720: percom_set:      dc.b  $28
399e 01 721:      dc.b  $01
399f 12 722:      dc.b  $12
39a0 00 723: sectors_per_trac:      dc.b  $00
39a1 00 724:      dc.b  $00
39a2 00 725:      dc.b  $00
39a3 00 726:      dc.b  $00
39a4 80 727: byte.sector.low:      dc.b  $80
39a5 ff 728:      dc.b  $FF
39a6 00 729:      dc.b  $00
39a7 00 730:      dc.b  $00
39a8 00 731:      dc.b  $00
39a9      732:
39a9      733:
39a9      734: ;ram_drive:      dc.b  $01
39a9      735:
39a9 60 736: ram_lookup:      dc.b  $60
39aa 64 737:      dc.b  $64
39ab 68 738:      dc.b  $68

```

```

39ac 6c 739:          dc.b  $6c
39ad 20 740:          dc.b  $20
39ae 24 741:          dc.b  $24
39af    742:
39af 28 743:          dc.b  $28
39b0 2c 744:          dc.b  $2c
39b1 40 745:          dc.b  $40
39b2 44 746:          dc.b  $44
39b3 48 747:          dc.b  $48
39b4 4c 748:          dc.b  $4c
39b5 00 749:          dc.b  $00
39b6 04 750:          dc.b  $04
39b7    751:
39b7 08 752:          dc.b  $08
39b8 0c 753:          dc.b  $0C
39b9 62 754:          dc.b  $62
39ba 66 755:          dc.b  $66
39bb 6a 756:          dc.b  $6a
39bc 6e 757:          dc.b  $6e
39bd 22 758:          dc.b  $22
39be 26 759:          dc.b  $26
39bf 2a 760:          dc.b  $2a
39c0 2e 761:          dc.b  $2e
39c1 42 762:          dc.b  $42
39c2 46 763:          dc.b  $46
39c3 4a 764:          dc.b  $4a
39c4 4e 765:          dc.b  $4e
39c5 02 766:          dc.b  $02
39c6 06 767:          dc.b  $06
39c7 0a 768:          dc.b  $0a
39c8 0e 769:          dc.b  $0E
39c9    770:
39c9    771:
39c9 7c 772: lotta_ram:      dc.b  $7C
39ca    773:
39ca    774:
39ca    775: rd_buffer:      ds.b  256
3aca    776:
3aca    777: zend:          ds.b   0
3aca    778: ;~~~~~
3aca    779: ;End of handler
3aca    780: ;
3aca    781: ;~~~~~
3aca    782: ram_ref:          ds.b   32
3aea    783: Reference:        ds.b   32
3b0a    784: ;~~~~~
3b0a    785: ;
3b0a    786: ;          Relocation data table
3b0a    787: ;          -----
3b0a    788: ;          We resolve Word locations First!
3b0a    789: ;
3b0a    790: ;  Things like  lda  $5000
3b0a    791: ;           sta  $5000

```



```

3b0a 792: ;          lda  $5000,y
3b0a 793: ;          jsr  $5000
3b0a 794: ;          jmp  $5000
3b0a 795: ;
3b0a 796: ;    Well you get the Idea..
3b0a 797: ;
3b0a 798: ;~~~~~
3b0a 799: ;
3b0a 800: rtable:
3b0a 801:
3b0a ac36 802:          dc.w  w01+1
3b0c b636 803:          dc.w  w02+1
3b0e bd36 804:          dc.w  w03+1
3b10      805: ;          dc.w  w04+1
3b10 3339 806:          dc.w  w05+1
3b12 6339 807:          dc.w  w06+1
3b14 8139 808:          dc.w  w07+1
3b16 7439 809:          dc.w  w08+1
3b18 8a38 810:          dc.w  w09+1
3b1a 9138 811:          dc.w  w10+1
3b1c ab38 812:          dc.w  w11+1
3b1e bd38 813:          dc.w  w12+1
3b20 c438 814:          dc.w  w13+1
3b22 c738 815:          dc.w  w14+1
3b24 df38 816:          dc.w  w15+1
3b26 e438 817:          dc.w  w16+1
3b28 1139 818:          dc.w  w17+1
3b2a 4139 819:          dc.w  w18+1
3b2c 4c39 820:          dc.w  W19+1
3b2e f538 821:          dc.w  w20+1
3b30 3e39 822:          dc.w  w21+1
3b32 4f39 823:          dc.w  w22+1
3b34 0000 824:          dc.w  0          ; o
3b36      825:
3b36      826:
3b36      827: ;
3b36 828: ;Ok these are low Byte Locations.. only one byte changed!
3b36 829: ; OK Resolving Low Byte's First
3b36 830: ;
3b36 9336 831:          dc.w  lb01+1
3b38 af36 832:          dc.w  lb02+1
3b3a 5338 833:          dc.w  lb03+1
3b3c 0000 834:          dc.w  0          ; o
3b3e      835: ;~~~~~
3b3e 836: ;Now to resolve High Bytes..... Only one byte to change!
3b3e 837: ; Here is what is needed... location of high byte in .wor
3b3e 838: ; Second thing needed is the low byte of the address you a
3b3e 839: ;to adjust!
3b3e 840: ;
3b3e 841: ;
3b3e 9736 842:          dc.w  hb01+1
3b40 4f 843:          dc.b  initz ;This will store ju
3b41 b936 844:          dc.w  Hb02+1

```

```

3b43 5d 845:          dc.b  ram_siov
3b44 5838 846:          dc.w  hb03+1
3b46 ca 847:          dc.b  zend
3b47 0000 848:          dc.w  0          ; o
3b49 00 849:          dc.b  0          ; o
3b4a 850:
3b4a 851:
3b4a 852: mepoint:      ds.b  0
3b4a 853:
3b4a 854:
3b4a 855: ;~~~~~
3b4a 856: ;
3b4a 857: ;          *** END OF HANDLER *
3b4a 858: ;~~~~~
3b4a 859: ;
3b4a 860: ;
3b4a 861: ; Relocater: main entry
3b4a 862: ; -----
3b4a 863: ; in:
3b4a 864: ; segtab = table of segment descriptors
3b4a 865: ; +00 = relocater table address
3b4a 866: ; +02 = originate address of block
3b4a 867: ; +04 = destination originate of block
3b4a 868: ; +06 = address of block
3b4a 869: ; +08 = number of bytes in block
3b4a 870: ; +10 = destination address of block
3b4a 871: ;
3b4a 872: ; reldab = relocater table
3b4a 873: ; list of address of words to adjust 2
3b4a 874: ; list of address low bytes to adjust 2
3b4a 875: ; list of address high bytes to adjust 3
3b4a 876: ; followed by their low bytes
3b4a 877: ;~~~~~
3b4a a900 878: rlocate:      lda  #0
3b4c 8d0e3c 879:          sta  segment
3b4f 880:
3b4f ae0e3c 881: segloop:      ldx  segment
3b52 bd113c 882:          lda  segtab+rettad,x
3b55 8df43b 883:          sta  relget+1
3b58 bd123c 884:          lda  segtab+rettad+1,x
3b5b 8df53b 885:          sta  relget+2
3b5e 0df43b 886:          ora  relget+1
3b61 d001 887:          bne  havseg
3b63 60 888:          rts
3b64 889:
3b64 38 890: havseg:      sec
3b65 bd153c 891:          lda  segtab+dorgadr,x
3b68 fd133c 892:          sbc  segtab+orgadr,x
3b6b 8d0f3c 893:          sta  zoffset
3b6e bd163c 894:          lda  segtab+dorgadr+1,x
3b71 fd143c 895:          sbc  segtab+orgadr+1,x
3b74 8d103c 896:          sta  zoffset+1
3b77 897:

```

```

3b77 20ff3b 898: zwordlp:      jsr  getzwp
3b7a f013 899:                beq  zlbytlp
3b7c      900:
3b7c b1d7 901:                lda  (zwptr),y
3b7e 18 902:                clc
3b7f 6d0f3c 903:                adc  zoffset
3b82 91d7 904:                sta  (zwptr),y
3b84 c8 905:                iny
3b85 b1d7 906:                lda  (zwptr),y
3b87 6d103c 907:                adc  zoffset+1
3b8a 91d7 908:                sta  (zwptr),y
3b8c 4c773b 909:                jmp  zwordlp
3b8f      910:
3b8f 20ff3b 911: zlbytlp:      jsr  getzwp
3b92 f00b 912:                beq  zhbytlp
3b94      913:
3b94 b1d7 914:                lda  (zwptr),y
3b96 18 915:                clc
3b97 6d0f3c 916:                adc  zoffset
3b9a 91d7 917:                sta  (zwptr),y
3b9c 4c8f3b 918:                jmp  zlbytlp
3b9f      919:
3b9f 20ff3b 920: zhbytlp:      jsr  getzwp
3ba2 f011 921:                beq  zmovlp
3ba4      922:
3ba4 20f33b 923:                jsr  relget
3ba7 18 924:                clc
3ba8 6d0f3c 925:                adc  zoffset
3bab b1d7 926:                lda  (zwptr),y
3bad 6d103c 927:                adc  zoffset+1
3bb0 91d7 928:                sta  (zwptr),y
3bb2 4c9f3b 929:                jmp  zhbytlp
3bb5      930:
3bb5 ae0e3c 931: zmovlp:      ldx  segment
3bb8 bd173c 932:                lda  segtab+blkadr,x
3bbb 8dd63b 933:                sta  zmovfr+1
3bbe bd183c 934:                lda  segtab+blkadr+1,x
3bc1 8dd73b 935:                sta  zmovfr+2
3bc4      936:
3bc4 bd1b3c 937:                lda  segtab+blkdes,x
3bc7 8dd93b 938:                sta  zmovto+1
3bca bd1c3c 939:                lda  segtab+blkdes+1,x
3bcd 8dda3b 940:                sta  zmovto+2
3bd0      941:
3bd0      942:
3bd0 bc1a3c 943:                ldy  segtab+blkbyt+1,x
3bd3      944:
3bd3 a200 945:                ldx  #0
3bd5 bfffff 946: zmovfr:      lda  $ffff,x
3bd8 9dffff 947: zmovto:      sta  $ffff,x
3bdb e8 948:                inx
3bdc d0f7 949:                bne  zmovfr
3bde eed73b 950:                inc  zmovfr+2

```

```

3be1 eeda3b 951:      inc    zmovto+2
3be4 88    952:      dey
3be5 10ee  953:      bpl    zmovfr
3be7      954:
3be7 ad0e3c 955:      lda    segment
3bea 18    956:      clc
3beb 690c  957:      adc    #seglen
3bed 8d0e3c 958:      sta    segment
3bf0 4c4f3b 959:      jmp    segloop
3bf3      960:
3bf3      961:
3bf3 adffff 962: relget:      lda    $ffff
3bf6 eef43b 963:      inc    relget+1
3bf9 d003   964:      bne    nc1
3bfb eef53b 965:      inc    relget+2
3bfe 60    966: nc1:      rts
3bff      967:
3bff 20f33b 968: getzwp:      jsr    relget
3c02 85d7   969:      sta    zwptr
3c04 20f33b 970:      jsr    relget
3c07 a000   971:      ldy    #0
3c09 85d8   972:      sta    zwptr+1
3c0b 05d7   973:      ora    zwptr
3c0d 60    974:      rts
3c0e      975:
3c0e 00    976: segment:      dc.b    0
3c0f 0000   977: zoffset:      dc.w    0
3c11      978: ;
3c11      979: ;
3c11      980: ;~~~~~
3c11      981: ;End of relocation Table!!!
3c11      982: ;
3c11      983: ;
3c11      984: segtab:
3c11 0a3b   985:      dc.w    rtable
3c13 4f38   986:      dc.w    initz
3c15 0000   987:      dc.w    0
3c17 4f38   988:      dc.w    initz
3c19 7b02   989:      dc.w    zend-initz
3c1b 0000   990:      dc.w    0
3c1d 0000   991:      dc.w    0
3c1f      992:
3c1f      993:
3c1f      994: ;
3c1f      995: ;~~~~~
3c1f      996: ;going to see if the wedge is present, if it is present th
3c1f      997: ;map out the extended ram
3c1f      998: ;~~~~~
3c1f      999: check_for_wedge:
3c1f 202c3d 1000:      jsr    os_off
3c22      1001:      mea    tsr,s_ptr
3c22 a9f5   ----:      lda    #low tsr
3c24 8582   ----:      sta    s_ptr

```

```

3c26 a9ff ----:      lda  #high tsr
3c28 8583 ----:      sta  s_ptr + 1
3c2a a903 1002:      lda  #wedge
3c2c 20893c 1003:      jsr  seclvl          ; c
3c2f 302d 1004:      bmi  h.a
3c31 a904 1005:      LDA   #4
3c33 8d9c39 1006:      STA   reserve_ram
3c36 201938 1007:      jsr  printsi
3c39 20 1008:      dc.b  " "
3c3a bdbdbd 1009:      dc.b  <+128>,"=====>"
3c44 d7e5e4 1010:      dc.b  <+128>,"Wedge detected"
3c52 bcbdbd 1011:      dc.b  <+128>,"<====="
3c5c 1012:
3c5c 9bff 1013:      dc.b  $9b,-1
3c5e 1014: h.a:
3c5e 20373d 1015:      jsr  os_on
3c61 60 1016:      rts
3c62 1017:
3c62 1018:
3c62 1019:
3c62 1020:
3c62 1021: ;
3c62 1022: ;~~~~~
3c62 1023: ;
3c62 1024: ;
3c62 1025: ;~~~~~
3c62 1026: set_ramdisk:
3c62 202c3d 1027:      jsr  os_off
3c65 1028:      mea  tsr,s_ptr
3c65 a9f5 ----:      lda  #low tsr
3c67 8582 ----:      sta  s_ptr
3c69 a9ff ----:      lda  #high tsr
3c6b 8583 ----:      sta  s_ptr + 1
3c6d a901 1029:      lda  #ramdisk
3c6f 20ea3c 1030:      jsr  setlvl
3c72 ad9c39 1031: w31:      lda  reserve_ram
3c75 c904 1032:      cmp  #$04
3c77 d00d 1033:      bne  .a
3c79 1034:
3c79 1035:      mea  tsr,s_ptr
3c79 a9f5 ----:      lda  #low tsr
3c7b 8582 ----:      sta  s_ptr
3c7d a9ff ----:      lda  #high tsr
3c7f 8583 ----:      sta  s_ptr + 1
3c81 a904 1036:      lda  #ram_ex
3c83 20ea3c 1037:      jsr  setlvl
3c86 1038: .a:
3c86 20373d 1039:      jsr  os_on
3c89 1040:
3c89 1041: ;~~~~~
3c89 1042: ;checks user security level pointed to by S_PTR to see if
3c89 1043: ;bit specified in the A register is ON. Returns MInus if
3c89 1044: ;user doesn't have the specified security level.

```

```

3c89      1045: ;~~~~~
3c89      1046:
3c89 8ce43c 1047: seclvl:          sty  .tempy
3c8c 8ee53c 1048:          stx  .tempx
3c8f c900  1049:          cmp  #0          ;se
3c91 f042  1050:          beq  .passed      ;ye
3c93 a8     1051:          tay          ;:-1
3c94 a980  1052:          lda  #128
3c96 8de63c 1053:          sta  .temp        ;in
3c99 a900  1054:          lda  #0
3c9b 8de73c 1055:          sta  .temp+1
3c9e 8de83c 1056:          sta  .temp+2
3ca1 8de93c 1057:          sta  .temp+3
3ca4 88     1058:          dey
3ca5 f011  1059:          beq  .mask        ;if
3ca7 4ee63c 1060: .shloop:      lsr  .temp        ;sh
3caa 6ee73c 1061:          ror  .temp+1
3cad 6ee83c 1062:          ror  .temp+2
3cb0 6ee93c 1063:          ror  .temp+3
3cb3 88     1064:          dey
3cb4 d0f1  1065:          bne  .shloop
3cb6 a000  1066:          ldy  #0
3cb8 b9e63c 1067: .mask:        lda  .temp,y        ;no
3cbb 3182  1068:          and  (s_ptr),y
3cbd 99e63c 1069:          sta  .temp,y
3cc0 c8     1070:          iny
3cc1 c004  1071:          cpy  #4
3cc3 d0f3  1072:          bne  .mask
3cc5 ade63c 1073: .check:        lda  .temp
3cc8 0de73c 1074:          ora  .temp+1        ;i
3ccb 0de83c 1075:          ora  .temp+2
3cce 0de93c 1076:          ora  .temp+3
3cd1 c900  1077:          cmp  #0
3cd3 f004  1078:          beq  .failed
3cd5 a900  1079: .passed:      lda  #0
3cd7 f002  1080:          beq  .return
3cd9 a980  1081: .failed:      lda  #$80
3cdb 08     1082: .return:      php
3cdc ace43c 1083:          ldy  .tempy
3cdf aee53c 1084:          ldx  .tempx
3ce2 28     1085:          plp
3ce3 60     1086:          rts
3ce4      1087:
3ce4 00     1088: .tempy:      dc.b  0
3ce5 00     1089: .tempx:      dc.b  0
3ce6 000000 1090: .temp:      dc.b  0,0,0,0
3cea      1091:
3cea      1092:
3cea      1093: ; sets the bit specified (in the A register) of the se
3cea      1094: ; level pointed to by S_PTR to a TRUE value.
3cea      1095: ; ~~~~~
3cea      1096:
3cea 8c263d 1097: setlvl:      sty  .tempy

```

```

3ced 8e273d 1098:      stx  .temp
3cf0 c900  1099:      cmp  #0          ;se
3cf2 f031  1100:      beq  .return     ;ye
3cf4 a8    1101:      tay              ;-
3cf5 a980  1102:      lda  #128
3cf7 8d283d 1103:      sta  .temp       ;in
3cfa a900  1104:      lda  #0
3cfc 8d293d 1105:      sta  .temp+1
3cff 8d2a3d 1106:      sta  .temp+2
3d02 8d2b3d 1107:      sta  .temp+3
3d05 88    1108:      dey
3d06 f011  1109:      beq  .mask       ;if
3d08 4e283d 1110: .shloop:  lsr  .temp     ;sh
3d0b 6e293d 1111:      ror  .temp+1
3d0e 6e2a3d 1112:      ror  .temp+2
3d11 6e2b3d 1113:      ror  .temp+3
3d14 88    1114:      dey
3d15 d0f1  1115:      bne  .shloop
3d17 a000  1116:      ldy  #0
3d19 b9283d 1117: .mask:    lda  .temp,y   ;no
3d1c 1182  1118:      ora  (s_ptr),y
3d1e 9182  1119:      sta  (s_ptr),y
3d20 c8    1120:      iny
3d21 c004  1121:      cpy  #4
3d23 d0f4  1122:      bne  .mask
3d25 60    1123: .return:  rts
3d26      1124:
3d26 00    1125: .tempy:   dc.b  0
3d27 00    1126: .tempx:   dc.b  0
3d28 000000 1127: .temp:    dc.b  0,0,0,0
3d2c      1128:
3d2c      1129: ;
3d2c      1130: ;~~~~~
3d2c      1131: ; OS on and off ROUTEEN
3d2c      1132: ;~~~~~
3d2c      1133: ;
3d2c      1134:
3d2c      1135: os_off:
3d2c 78    1136:      SEI
3d2d ad01d3 1137:      LDA  PORTB
3d30 29fe  1138:      AND  #$FE
3d32 8d01d3 1139:      STA  PORTB
3d35 58    1140:      CLI
3d36 60    1141:      RTS
3d37      1142: ;
3d37      1143: os_on:
3d37 78    1144:      SEI
3d38 ad01d3 1145:      LDA  PORTB
3d3b 0901  1146:      ORA  #1
3d3d 8d01d3 1147:      STA  PORTB
3d40 58    1148:      CLI
3d41 60    1149:      RTS
3d42      1150:

```

```
3d42    1151:
3d42    1152:
3d42    1153:
3d42    1154:
3d42    1155: win_end:      ds.b  0
3d42    1156: ;
3d42    1157: ;
3d42    1158:
```

End assembly: no errors