


```

0000 1: ;-----
0000 2: ;
0000 3: ;      Copyright 2006 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: ;      hyp_r (utility for real.dos v1.0)
0000 10: ;      Real.dos hpy_r.com
0000 11: ;
0000 12: ;      Code History.  Init design, 1994
0000 13: ;      last updated 11/04/2006
0000 14: ;      converted to xasm
0000 15: ;      combined both mio code and black
0000 16: ;      code and installed a routine to c
0000 17: ;      for each device.
0000 18: ;
0000 19: ;
0000 20: ;
0000 21: ;      ~~~~~
0000 22: ;-----
0000 23: ;Optimized R: handler
0000 24: ;for Multi-I/O and Black Box
0000 25: ;
0000 26: ;Multi-I/O
0000 27: ;Black Box tm CSS
0000 28: ;
0000 29: ;This is a replacement handler for the ROM handlers in the
0000 30: ;Black Box, in an effort to provide a more efficient hand
0000 31: ;high-speed modems. The problem is not so much the handle
0000 32: ;rather the Atari XL/XE operating system, and its overhead
0000 33: ;what is called the "generic handler just to access the RO
0000 34: ;There is also an excess of overhead code in the interrupt
0000 35: ;get to the ROM interrupt service routines. This memory-r
0000 36: ;handler has its own IRQ routine that is patched into the
0000 37: ;$0216, so it can check the hardware directly.
0000 38: ;
0000 39: ;This source code assembles into two versions, one for the
0000 40: ;other for the Black Box, depending on how the variable MI
0000 41: ;MIO=0, a Black Box version will be assembled. If MIO=non
0000 42: ;version will result.
0000 43: ;
0000 44: ;This handler is optimized for using the RS-232 port stric
0000 45: ;operations, so do not attempt to use it for a serial prin
0000 46: ;will lock up.
0000 47: ;
0000 48: ;-----
0000 49: ; Notes: o This source code MAY NOT be placed for download
0000 50: ;      o "mea" is a macro that loads the address of the
0000 51: ;      into the pointer specified by the second field.
0000 52: ;-----
0000 53: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)

```

```

0000 54: ; XASM st and IBM VERSIONS
0000 55: ;-----
0000 56: ;
0000 57: com_file_name: .macro
0000 58: ; dc.b 13,"filename.ext",$9b
0000 59: ; dc.b 12,"HYP_r.com ", $9b
0000 60: ; .endm
0000 61:
0000 62: ;
0000 63: ;~~~~~
0000 64: ; File revision history. Version Number in hex
0000 65: ;
0000 66: file_ver: equ $10
0000 67: ;~~~~~
0000 68: ; the Month in dec for the first time this revision
0000 69: ;was compiled
0000 70: ;
0000 71: c_month: equ 3
0000 72: ;~~~~~
0000 73: ; the day in dec for the first time this revision
0000 74: ;was compiled
0000 75: c_day: equ 31
0000 76: ;~~~~~
0000 77: ; the year in dec for the first time this revision
0000 78: ;was compiled
0000 79: c_year: equ 2010
0000 80: ;~~~~~
0000 81: ; Type of compiler used in program
0000 82: ;
0000 83: ; 0 = unknown Compiler
0000 84: ; 1 = xasm
0000 85: ; 2 = mac_65
0000 86: ; 3 = basic
0000 87: ; 4 = compiled basic xl
0000 88: ; 5 = C65
0000 89: ; 6 = Action
0000 90: ;
0000 91: ; We can add more as time goes on!
0000 92: ;
0000 93: ;
0000 94: xasm: equ 1
0000 95: mac_65: equ 2
0000 96: basic: equ 3
0000 97: basicxl: equ 4
0000 98: c65: equ 5
0000 99: action: equ 6
0000 100:
0000 101: file_compiler: equ xasm
0000 102: ;~~~~~
0000 103: ; is this relocatable code ?
0000 104: ; anyother value other than 1 or 2 would be unknown
0000 105: r.yes: equ 1
0000 106: r.no: equ 2

```

```

0000 107: ;
0000 108: ;~~~~~
0000 109: ; Gotta define if it can be relocated
0000 110: l_relocatable: equ r..no
0000 111: ;
0000 112: ;
0000 113: ;~~~~~
0000 114: ; Gotta define if it can be relocated
0000 115: r..crl: equ $7d
0000 116: r..crlf: equ $9b
0000 117: r..space: equ $20
0000 118:
0000 119: l_frstscreenbyte: equ r..space
0000 120: ;~~~~~
0000 121: ; The language the output file is in.
0000 122: ;
0000 123: ;
0000 124: ; 0 = Undefined
0000 125: ; 1 = English
0000 126: ; 2 = German
0000 127: ;
0000 128: r..language: equ 1
0000 129: ;~~~~~
0000 130: ;
0000 131:
0000 132:
0000 133: .include equates
0000 134: .include globals
0000 135: .include macros
0000 136: ;
0000 137:
0000 138:
0000 139:
0000 140:
0000 141:
0000 142: .org $4000
0000 143:
0000 144:
0000 145:
0000 146: win_start:
0000 147: header_info:
0000 148: .include header
0000 149:
42a7 150: jsr os_off
42aa 151: mea tsr,s_ptr
42aa a9f5 ----: lda #low tsr
42ac 8582 ----: sta s_ptr
42ae a9ff ----: lda #high tsr
42b0 8583 ----: sta s_ptr + 1
42b2 a91c 152: lda #Hyp_r
42b4 203652 153: jsr seclvl
42b7 304d 154: bmi .aa
42b9 206f4a 155: jsr printsi

```

```

42bc 9b 156:          dc.b  $9b
42bd 20 157:          dc.b  " "
42be bdbdbe 158:       dc.b  <+128>,"==>"
42c1 c8f9f0 159:       dc.b  <+128>,"Hyper R is
42db bcbdbd 160:       dc.b  <+128>,"<=="
42de 9b 161:          dc.b  $9b
42df 20 162:          dc.b  " "
42e0 bdbdbe 163:       dc.b  <+128>,"==>"
42e3 c1eee4 164:       dc.b  <+128>,"And Registe
42fd bcbdbd 165:       dc.b  <+128>,"<=="
4300 9bff 166:          dc.b  $9b,-1
4302 202b52 167:       jsr   os_on
4305 60 168:          rts
4306      169:
4306      170:
4306      171: .aa:
4306 202b52 172:       jsr   os_on
4309      173:
4309 20c54a 174:       jsr   test_pbi
430c ad6743 175:       lda   pbi_device
430f c901 176:       cmp   #$01
4311 f007 177:       beq   go_mio
4313 c902 178:       cmp   #$02
4315 f006 179:       beq   go_blackbox
4317 4c2043 180:       jmp   no_comp_pbi
431a      181:
431a 4ceb47 182: go_mio:          jmp   relocate
431d 4c3e50 183: go_blackbox:      jmp   relocate.bb
4320      184:
4320      185: no_comp_pbi:
4320 206f4a 186:       jsr   printsi
4323 9b9b 187:       dc.b  $9b,$9b
4325 20596f 188:       dc.b  " You must have a B
4347 20746f 189:       dc.b  " to use the Hyper
4365 ff 190:          dc.b  -1
4366 60 191:          rts
4367      192:
4367      193:
4367 00 194: pbi_device:      dc.b  $00
4368      195:
4368      196:
4368      197:
4368      198:
4368      199:
4368      200:
4368      201: start:
4368      202:
4368      203: ;Needed by relocater
4368 7a44 204: hatab:          dc.w  open-1
436a 9244 205: w01:          dc.w  close-1
436c f443 206: w02:          dc.w  get-1
436e 1d44 207: w03:          dc.w  put-1
4370 aa44 208: w04:          dc.w  status-1

```

```

4372 e844 209: w05:          dc.w  special-1
4374 00 210:          dc.b  0
4375      211:
4375 00 212: staterr:      dc.b  0
4376 00 213: nextin:          dc.b  0
4377 00 214: lastin:          dc.b  0
4378 00 215: nextout:         dc.b  0
4379 00 216: lastout:        dc.b  0
437a 00 217: concur:         dc.b  0
437b 00 218: mode:          dc.b  0
437c 00 219: trans:         dc.b  0
437d      220: ;
437d      221: ;  "INTERRUPT ROUTINE"
437d      222: ;
437d 48 223: irq:          pha
437e 8a 224:          txa
437f aec1d1 225:          ldx  $d1c1
4382 3005 226:          bmi  irq2
4384 aa 227:          tax
4385 68 228:          pla
4386 4c30c0 229: sysirq:      jmp  $c030
4389 48 230: irq2:          pha
438a 98 231:          tya
438b 48 232:          pha
438c 8a 233:          txa
438d 2908 234:          and  #8
438f f01d 235:          beq  trysend
4391 ac7743 236: w06:          ldy  lastin
4394 adc0d1 237:          lda  $d1c0
4397 999e45 238: w07:          sta  mio.inbuff,y
439a c8 239:          iny
439b 8c7743 240: w08:          sty  lastin
439e cc7643 241: w09:          cpy  nextin
43a1 d02b 242:          bne  irqexit
43a3 ee7643 243: w10:          inc  nextin
43a6 ad7543 244: w11:          lda  staterr
43a9 0910 245:          ora  #$10
43ab 8d7543 246: w12:          sta  staterr
43ae 8a 247: trysend:      txa
43af 2910 248:          and  #$10
43b1 f01b 249:          beq  irqexit
43b3 adc2d1 250:          lda  $d1c2
43b6 290c 251:          and  #$0c
43b8 c904 252:          cmp  #4
43ba d012 253:          bne  irqexit
43bc adffd1 254: send02:      lda  $d1ff
43bf 2904 255:          and  #$04
43c1 f0f9 256:          beq  send02
43c3 ac7843 257: w13:          ldy  nextout
43c6 cc7943 258: w14:          cpy  lastout
43c9 d00c 259:          bne  sendit
43cb 20e443 260: w15:          jsr  dstxirq
43ce 68 261: irqexit:      pla

```

```

43cf a8 262:          tay
43d0 263:
43d0 204344 264: w16:      jsr  enirq
43d3 68 265:          pla
43d4 aa 266:          tax
43d5 68 267:          pla
43d6 40 268:          rti
43d7 c8 269: sendit:      iny
43d8 b99e46 270: w17:      lda  outbuff,y
43db 8dc0d1 271:          sta  $d1c0
43de 8c7843 272: w18:      sty  nextout
43e1 4cce43 273: w19:      jmp  irqexit
43e4 a908 274: dstxirq:    lda  #8
43e6 2c 275:          dc.b $2c
43e7 a904 276: entxirq:    lda  #4
43e9 4dc2d1 277:          eor  $d1c2
43ec 290c 278:          and  #$0c
43ee 4dc2d1 279:          eor  $d1c2
43f1 8dc2d1 280:          sta  $d1c2
43f4 60 281:          rts
43f5 282: ;
43f5 283: ;   "GET BYTE ROUTINE"
43f5 284: ;
43f5 2411 285: get:      bit  brkkey
43f7 101c 286:          bpl  get10
43f9 ae7643 287: w20:      ldx  nextin
43fc ec7743 288: w21:      cpx  lastin
43ff f0f4 289:          beq  get
4401 bc9e45 290: w22:      ldy  mio.inbuff,x
4404 e8 291:          inx
4405 8e7643 292: w23:      stx  nextin
4408 ad7c43 293: w24:      lda  trans
440b 2920 294:          and  #$20
440d d006 295:          bne  get10
440f c00d 296:          cpy  #$0c+1
4411 d002 297:          bne  get10
4413 a09b 298:          ldy  #$9b
4415 98 299: get10:    tya
4416 48 300:          pha
4417 203944 301: w25:      jsr  setbreak
441a 68 302:          pla
441b c000 303:          cpy  #0
441d 60 304:          rts
441e 305: ;
441e 306: ;   "PUT BYTE, SET BREAK"
441e 307: ;
441e a8 308: put:      tay
441f ad7c43 309: w26:      lda  trans
4422 2920 310:          and  #$20
4424 d010 311:          bne  put03
4426 c09b 312:          cpy  #$9b
4428 d00c 313:          bne  put03
442a a00d 314:          ldy  #$0c+1

```

```

442c 206144 315: w27:          jsr  put10
442f 2c7c43 316: w28:          bit  trans
4432 5005 317:          bvc  setbreak
4434 a00a 318:          ldy  #$0a
4436 206144 319: put03:        jsr  put10
4439 a001 320: setbreak:    ldy  #1
443b 2411 321:          bit  brkkey
443d 3004 322:          bmi  enirq
443f a080 323:          ldy  #$80
4441 8411 324:          sty  brkkey
4443      325: ENIRQ:
4443 78 326:          sei
4444 a200 327:          ldx  #0
4446 a920 328:          lda  #$20
4448 8ee0d1 329:          stx  $d1e0
444b 8de2d1 330:          sta  $d1e2
444e adfdd6 331:          lda  $d6fc+1
4451 aefcd6 332:          ldx  $d6fc
4454 0980 333:          ora  #$80
4456 8ee0d1 334:          stx  $d1e0
4459 8de2d1 335:          sta  $d1e2
445c 58 336:          cli
445d c000 337:          cpy  #0
445f 38 338:          sec
4460 60 339:          rts
4461 ae7943 340: put10:        ldx  lastout
4464 e8 341:          inx
4465 2411 342: put11:        bit  brkkey
4467 1011 343:          bpl  putexit
4469 ec7843 344: w29:          cpx  nextout
446c f0f7 345:          beq  put11
446e 78 346:          sei
446f 98 347:          tya
4470 9d9e46 348: w30:          sta  outbuff,x
4473 8e7943 349: w31:          stx  lastout
4476 20e743 350: w32:          jsr  entxirq
4479 58 351:          cli
447a 60 352: putexit: rts
447b      353: ;
447b      354: ;   "OPEN, CLOSE R: CHANNEL"
447b      355: ;
447b a52a 356: open:          lda  icax1z
447d 8d7b43 357: w33:          sta  mode
4480 78 358:          sei
4481 a205 359:          ldx  #5
4483 a900 360:          lda  #0
4485 9d7543 361: op01:          sta  staterr,x
4488 ca 362:          dex
4489 10fa 363:          bpl  op01
448b ad7b43 364: open03:        lda  mode
448e 852a 365:          sta  icax1z
4490 4c3944 366: w34:          jmp  setbreak
4493      367: ;

```



```

4493      368: ; CLOSE R: HANDLER CHANNEL
4493      369: ;
4493 a900  370: close:          lda  #0
4495 8d7a43 371: w35:          sta  concur
4498 209e44 372: xio32:        jsr  forceout
449b 4c8b44 373: w36:          jmp  open03
449e ad7943 374: forceout:      lda  lastout
44a1 2411  375: f01:          bit  brkkey
44a3 1005  376:          bpl  f02
44a5 cd7843 377: w37:          cmp  nextout
44a8 d0f7  378:          bne  f01
44aa 60    379: f02:          rts
44ab      380: ;
44ab      381: ; "GET STATUS OF R: CHANNEL"
44ab      382: ;
44ab adffd1 383: status:          lda  $d1ff
44ae 2907  384:          and  #7
44b0 aa    385:          tax
44b1 bde144 386: w38:          lda  hsflags,x
44b4 8deb02 387:          sta  dvstat+1
44b7 2c7a43 388: w39:          bit  concur
44ba 1014  389:          bpl  stat02
44bc 38    390:          sec
44bd ad7743 391: w40:          lda  lastin
44c0 ed7643 392: w41:          sbc  nextin
44c3 8deb02 393:          sta  dvstat+1
44c6 38    394:          sec
44c7 ad7943 395: w42:          lda  lastout
44ca ed7843 396: w43:          sbc  nextout
44cd 8ded02 397:          sta  dvstat+3
44d0 ad7543 398: stat02:        lda  staterr
44d3 8dea02 399:          sta  dvstat
44d6 a900  400:          lda  #0
44d8 8d7543 401: w44:          sta  staterr
44db 8dec02 402:          sta  dvstat+2
44de 4c3944 403: w45:          jmp  setbreak
44e1      404:
44e1      405:
44e1      406: hSFLAGS:
44e1 010dc1 407:          dc.b $01,$0d,$c1,$cd
44e5 313df1 408:          dc.b $31,$3d,$f1,$fd
44e9      409: ;
44e9      410: ; "XIO 40, 38, 36 CALLS"
44e9      411: ;
44e9 a622  412: special:        ldx  iccomz
44eb a52a  413:          lda  icax1z
44ed e028  414:          cpx  #$28
44ef d00b  415:          bne  xio38
44f1 ad7a43 416: w46:          lda  concur
44f4 0980  417:          ora  #$80
44f6 8d7a43 418: w47:          sta  concur
44f9 4c8b44 419: w48:          jmp  open03
44fc e026  420: xio38:         cpx  #$26

```

```

44fe d01a 421:          bne xio36
4500 8d7c43 422: w49:      sta trans
4503 4a 423:          lsr
4504 4a 424:          lsr
4505 2903 425:          and #3
4507 a8 426:          tay
4508 adc2d1 427:          lda $d1c2
450b 291f 428:          and #$1f
450d 191645 429: w50:      ora parity,y
4510 8dc2d1 430:          sta $d1c2
4513 4c8b44 431: w51:      jmp open03
4516 432:
4516 002060 433: parity:      dc.b 0,$20,$60,$
451a 434:
451a 435:
451a e024 436: xio36:      cpx #$24
451c d038 437:          bne xio34
451e a8 438:          tay
451f 29d0 439:          and #$d0
4521 0a 440:          asl
4522 08 441:          php
4523 0a 442:          asl
4524 28 443:          plp
4525 6a 444:          ror
4526 48 445:          pha
4527 98 446:          tya
4528 290f 447:          and #$0f
452a f00d 448:          beq xio36c
452c a208 449:          ldx #8
452e c905 450:          cmp #5
4530 f008 451:          beq xio36a
4532 38 452:          sec
4533 c908 453:          cmp #8
4535 9016 454:          bcc err132
4537 2907 455:          and #7
4539 aa 456: xio36c:      tax
453a 68 457: xio36a:      pla
453b 1d4445 458: w52:      ora baud,x
453e 8dc3d1 459:          sta $d1c3
4541 4c8b44 460: xio36b:      jmp open03
4544 161718 461: baud:      dc.b $16,$17,$18,$19
4548 1a1c1e 462:          dc.b $1a,$1c,$1e,$1f,$13
454d 463: ; "ERROR 132
454d 464: ; XIO 34, 32"
454d 203944 465: err132:      jsr setbreak
4550 98 466:          tya
4551 3002 467:          bmi er132a
4553 a084 468:          ldy #$84
4555 60 469: er132a:      rts
4556 e022 470: xio34:      cpx #$22
4558 d03d 471:          bne xio32a
455a aa 472:          tax
455b 100e 473:          bpl xio34a

```

```

455d 2940 474:          and    #$40
455f 0a    475:          asl
4560 0a    476:          asl
4561 08    477:          php
4562 adc2d1 478:          lda    $d1c2
4565 4a    479:          lsr
4566 28    480:          plp
4567 2a    481:          rol
4568 8dc2d1 482:          sta    $d1c2
456b 8a    483: xio34a:          txa
456c 2930 484:          and    #$30
456e c930 485:          cmp    #$30
4570 f00f 486:          beq    xio34d
4572 c920 487:          cmp    #$20
4574 f011 488:          beq    xio34c
4576 8a    489:          txa
4577 2903 490:          and    #3
4579 c903 491:          cmp    #3
457b f007 492:          beq    xio34b
457d c902 493:          cmp    #2
457f d013 494:          bne    w53
4581 a908 495: xio34d:          lda    #8
4583 2c    496:          dc.b   $2c
4584 a90c 497: xio34b:          lda    #$0c
4586 2c    498:          dc.b   $2c
4587 a900 499: xio34c:          lda    #0
4589 4dc2d1 500:          eor    $d1c2
458c 290c 501:          and    #$0c
458e 4dc2d1 502:          eor    $d1c2
4591 8dc2d1 503:          sta    $d1c2
4594 4c8b44 504: w53:          jmp    open03
4597 e020 505: xio32a:          cpx    #$20
4599 d0b2 506:          bne    err132
459b 4c9844 507: w54:          jmp    xio32
459e      508: mio.inbuff:      ds.b   256
469e      509: outbuff:         ds.b   256
479e      510: ;
479e      511: ; "RESET INITIALIZATION"
479e 20ffff 512: init:          jsr    $ffff
47a1 a9eb 513: init02:          lda    # low reloc
47a3 8de702 514:          sta    memlo
47a6 a947 515: h03:          lda    # high relocate
47a8 8de802 516:          sta    memlo+1
47ab a200 517:          ldx    #0
47ad bd1a03 518: init03:          lda    $031a,x
47b0 f009 519:          beq    sethand
47b2 c952 520:          cmp    #'R'
47b4 f00a 521:          beq    set02
47b6 e8    522:          inx
47b7 e8    523:          inx
47b8 e8    524:          inx
47b9 d0f2 525:          bne    init03
47bb a952 526: sethand:        lda    #'R'

```

```

47bd 9d1a03 527:          sta  $031a,x
47c0 a968 528: set02:      lda  # low hatab
47c2 9d1b03 529:          sta  $031b,x
47c5 a943 530: h01:       lda  # high hatab
47c7 9d1c03 531:          sta  $031c,x
47ca ad1602 532:          lda  $0216
47cd 8d8743 533: w55:      sta  sysirq+1
47d0 ad1702 534:          lda  $0216+1
47d3 8d8843 535: w56:      sta  sysirq+2
47d6 78 536:          sei
47d7 a97d 537: l02:      lda  # low irq
47d9 8d1602 538:          sta  $0216
47dc a943 539: h02:      lda  # high irq
47de 8d1702 540:          sta  $0216+1
47e1 58 541:          cli
47e2 adc2d1 542:          lda  $d1c2
47e5 29fe 543:          and  #$fe
47e7 8dc2d1 544:          sta  $d1c2
47ea 60 545: inout:    rts
47eb 546: ;
47eb 547: ;  GENERIC RELOCATOR -this is my early relocater
47eb 548: ;
47eb 549: relocate:
47eb 20d952 550:          jsr  set_handler
47ee 551:
47ee 552:
47ee a200 553:          ldx  #0
47f0 bd1a03 554: rel01:    lda  $031a,x
47f3 f00a 555:          beq  rel01a
47f5 e8 556:          inx
47f6 e8 557:          inx
47f7 e8 558:          inx
47f8 e021 559:          cpx  #32+1
47fa 90f4 560:          bcc  rel01
47fc 561:
47fc 4c1249 562:          jmp  no_instal_error
47ff 563:
47ff 564: rel01a:
47ff ad6743 565:          lda  pbi_device
4802 c901 566:          cmp  #$01
4804 f003 567:          beq  .a
4806 4c1249 568:          jmp  no_instal_error
4809 569: .a:
4809 a968 570:          lda  # low start
480b 38 571:          sec
480c ede702 572:          sbc  memlo
480f 8d4749 573:          sta  offset
4812 a943 574:          lda  # high start
4814 ede802 575:          sbc  memlo+1
4817 8d4849 576:          sta  offset+1
481a a200 577:          ldx  #0
481c a000 578:          ldy  #0
481e bd4949 579: rel02:    lda  tablo,x

```

```

4821 85e0 580:          sta  $e0
4823 bddb49 581: rel03:      lda  tabhi,x
4826 85e2 582:          sta  $e2
4828 e8 583:          inx
4829 bd4949 584: rel04:      lda  tablo,x
482c 85e1 585:          sta  $e0+1
482e bddb49 586: rel05:      lda  tabhi,x
4831 85e3 587:          sta  $e2+1
4833 05e2 588:          ora  $e2
4835 f021 589:          beq  moveit
4837 b1e0 590:          lda  ($e0),y
4839 38 591:          sec
483a ed4749 592:          sbc  offset
483d 91e0 593:          sta  ($e0),y
483f b1e2 594:          lda  ($e2),y
4841 ed4849 595:          sbc  offset+1
4844 91e2 596:          sta  ($e2),y
4846 e8 597:          inx
4847 d0d5 598:          bne  rel02
4849 ee2048 599:          inc  rel02+2
484c ee2548 600:          inc  rel03+2
484f ee2b48 601:          inc  rel04+2
4852 ee3048 602:          inc  rel05+2
4855 4c1e48 603:          jmp  rel02
4858      604:
4858      605:
4858      606:
4858      607: ; "MOVE CODE TO MEMLO"
4858      608:
4858 a968 609: moveit:      lda  # low start
485a 85e0 610:          sta  $e0
485c a943 611:          lda  # high start
485e 85e1 612:          sta  $e0+1
4860 ade702 613:          lda  memlo
4863 85e2 614:          sta  $e2
4865 ade802 615:          lda  memlo+1
4868 85e3 616:          sta  $e2+1
486a a983 617:          lda  # low relocate-star
486c 85e4 618:          sta  $e4
486e a904 619:          lda  # high relocate-sta
4870 85e5 620:          sta  $e4+1
4872 b1e0 621: move02:      lda  ($e0),y
4874 91e2 622:          sta  ($e2),y
4876 e6e0 623:          inc  $e0
4878 d002 624:          bne  move03
487a e6e1 625:          inc  $e0+1
487c e6e2 626: move03:      inc  $e2
487e d002 627:          bne  move04
4880 e6e3 628:          inc  $e2+1
4882 a5e4 629: move04:      lda  $e4
4884 d002 630:          bne  move05
4886 c6e5 631:          dec  $e4+1
4888 c6e4 632: move05:      dec  $e4

```

```

488a d0e6 633:      bne  move02
488c a5e5 634:      lda  $e4+1
488e d0e2 635:      bne  move02
4890 a50c 636: relinit:      lda  $0c
4892 8d9f47 637: m06:      sta  init+1
4895 a50d 638:      lda  $0c+1
4897 8da047 639: m07:      sta  init+2
489a a99e 640: m08:      lda  # low init
489c 850c 641:      sta  $0c
489e a947 642: m09:      lda  # high init
48a0 850d 643:      sta  $0c+1
48a2 20a147 644: m10:      jsr  init02
48a5      645:
48a5      646: prtmsg:
48a5 206f4a 647:      jsr  printsi
48a8      648:
48a8 487970 649:      dc.b  "HyperSpeed RS232 a
48c5 566572 650:      dc.b  "Version",-1
48ce ad0340 651:      lda  f.c_ver ;cmd_ve
48d1 20d34a 652:      jsr  ver_num
48d4 ad6743 653:      lda  pbi_device
48d7 c902 654:      cmp  #$02
48d9 f015 655:      beq  .got_bb
48db      656:
48db 206f4a 657:      jsr  printsi
48de 20666f 658:      dc.b  " for Multi I/O",-1
48ed 4c0249 659:      jmp  .gotendmsg
48f0      660:
48f0 206f4a 661: .got_bb:      jsr  printsi
48f3 20666f 662:      dc.b  " for Black Box",-1
4902      663:
4902 206f4a 664: .gotendmsg:      jsr  printsi
4905 9b 665:      dc.b  $9b
4906 c9eef3 666:      dc.b  <+128>,"Installed"
490f 9bff 667:      dc.b  $9b,-1
4911 60 668:      rts
4912      669:
4912      670:
4912 206f4a 671: no_instal_error:      jsr  printsi
4915 9b9b 672:      dc.b  $9b,$9b
4917 c8e1ee 673:      dc.b  <+128>,"Handler tab
492a 9b 674:      dc.b  $9B
492b c8f9f0 675:      dc.b  <+128>,"HyperSpeed
4944 9bff 676:      dc.b  $9b,-1
4946 60 677:      rts
4947      678: errend:
4947      679:
4947 0000 680: offset:      dc.w 0
4949      681:
4949      682:
4949      683: ; "RELOCATOR TABLE 1"
4949      684: ;
4949      685: ;Each TABLO entry MUST have a matching

```

```

4949      686: ;TABHI entry. TABLO is for the low
4949      687: ;bytes to be offset, TABHI is for the
4949      688: ;high bytes. TABHI MUST END WITH ZERO
4949      689: ;OR THE RELOCATOR WILL NOT KNOW WHEN
4949      690: ;TO STOP!
4949      691: ;
4949 9348  692: tablo:          dc.w  m06+1
494b 9848  693:          dc.w  m07+1
494d 9b48  694:          dc.w  m08+1
494f a348  695:          dc.w  m10+1
4951 6843  696:          dc.w  hatab
4953 6a43  697:          dc.w  w01
4955 6c43  698:          dc.w  w02
4957 6e43  699:          dc.w  w03
4959 7043  700:          dc.w  w04
495b 7243  701:          dc.w  w05
495d 9243  702:          dc.w  w06+1
495f 9843  703:          dc.w  w07+1
4961 9c43  704:          dc.w  w08+1
4963 9f43  705:          dc.w  w09+1
4965 a443  706:          dc.w  w10+1
4967 a743  707:          dc.w  w11+1
4969 ac43  708:          dc.w  w12+1
496b c443  709:          dc.w  w13+1
496d c743  710:          dc.w  w14+1
496f cc43  711:          dc.w  w15+1
4971 d143  712:          dc.w  w16+1
4973 d943  713:          dc.w  w17+1
4975 df43  714:          dc.w  w18+1
4977 e243  715:          dc.w  w19+1
4979 fa43  716:          dc.w  w20+1
497b fd43  717:          dc.w  w21+1
497d 0244  718:          dc.w  w22+1
497f 0644  719:          dc.w  w23+1
4981 0944  720:          dc.w  w24+1
4983 1844  721:          dc.w  w25+1
4985 2044  722:          dc.w  w26+1
4987 2d44  723:          dc.w  w27+1
4989 3044  724:          dc.w  w28+1
498b 6a44  725:          dc.w  w29+1
498d 7144  726:          dc.w  w30+1
498f 7444  727:          dc.w  w31+1
4991 7744  728:          dc.w  w32+1
4993 7e44  729:          dc.w  w33+1
4995 9144  730:          dc.w  w34+1
4997 9644  731:          dc.w  w35+1
4999 9c44  732:          dc.w  w36+1
499b a644  733:          dc.w  w37+1
499d b244  734:          dc.w  w38+1
499f b844  735:          dc.w  w39+1
49a1 be44  736:          dc.w  w40+1
49a3 c144  737:          dc.w  w41+1
49a5 c844  738:          dc.w  w42+1

```

```

49a7 cb44 739:          dc.w  w43+1
49a9 d944 740:          dc.w  w44+1
49ab df44 741:          dc.w  w45+1
49ad f244 742:          dc.w  w46+1
49af f744 743:          dc.w  w47+1
49b1 fa44 744:          dc.w  w48+1
49b3 0145 745:          dc.w  w49+1
49b5 0e45 746:          dc.w  w50+1
49b7 1445 747:          dc.w  w51+1
49b9 3c45 748:          dc.w  w52+1
49bb 9545 749:          dc.w  w53+1
49bd 9c45 750:          dc.w  w54+1
49bf ce47 751:          dc.w  w55+1
49c1 d447 752:          dc.w  w56+1
49c3 c147 753:          dc.w  set02+1
49c5 d847 754:          dc.w  l02+1
49c7 3744 755:          dc.w  put03+1
49c9 6244 756:          dc.w  put10+1
49cb 8644 757:          dc.w  op01+1
49cd 8c44 758:          dc.w  open03+1
49cf 9944 759:          dc.w  xio32+1
49d1 a247 760:          dc.w  init02+1
49d3 9f44 761:          dc.w  forceout+1
49d5 d144 762:          dc.w  stat02+1
49d7 4245 763:          dc.w  xio36b+1
49d9 4e45 764:          dc.w  err132+1
49db      765:
49db      766:
49db      767:
49db      768: ; "RELOCATOR TABLE 2"
49db      769: tabhi:
49db 9448 770:          dc.w  m06+2
49dd 9948 771:          dc.w  m07+2
49df 9f48 772:          dc.w  m09+1
49e1 a448 773:          dc.w  m10+2
49e3 6943 774:          dc.w  hatab+1
49e5 6b43 775:          dc.w  w01+1
49e7 6d43 776:          dc.w  w02+1
49e9 6f43 777:          dc.w  w03+1
49eb 7143 778:          dc.w  w04+1
49ed 7343 779:          dc.w  w05+1
49ef 9343 780:          dc.w  w06+2
49f1 9943 781:          dc.w  w07+2
49f3 9d43 782:          dc.w  w08+2
49f5 a043 783:          dc.w  w09+2
49f7 a543 784:          dc.w  w10+2
49f9 a843 785:          dc.w  w11+2
49fb ad43 786:          dc.w  w12+2
49fd c543 787:          dc.w  w13+2
49ff c843 788:          dc.w  w14+2
4a01 cd43 789:          dc.w  w15+2
4a03 d243 790:          dc.w  w16+2
4a05 da43 791:          dc.w  w17+2

```

4a07	e043	792:	dc.w	w18+2
4a09	e343	793:	dc.w	w19+2
4a0b	fb43	794:	dc.w	w20+2
4a0d	fe43	795:	dc.w	w21+2
4a0f	0344	796:	dc.w	w22+2
4a11	0744	797:	dc.w	w23+2
4a13	0a44	798:	dc.w	w24+2
4a15	1944	799:	dc.w	w25+2
4a17	2144	800:	dc.w	w26+2
4a19	2e44	801:	dc.w	w27+2
4a1b	3144	802:	dc.w	w28+2
4a1d	6b44	803:	dc.w	w29+2
4a1f	7244	804:	dc.w	w30+2
4a21	7544	805:	dc.w	w31+2
4a23	7844	806:	dc.w	w32+2
4a25	7f44	807:	dc.w	w33+2
4a27	9244	808:	dc.w	w34+2
4a29	9744	809:	dc.w	w35+2
4a2b	9d44	810:	dc.w	w36+2
4a2d	a744	811:	dc.w	w37+2
4a2f	b344	812:	dc.w	w38+2
4a31	b944	813:	dc.w	w39+2
4a33	bf44	814:	dc.w	w40+2
4a35	c244	815:	dc.w	w41+2
4a37	c944	816:	dc.w	w42+2
4a39	cc44	817:	dc.w	w43+2
4a3b	da44	818:	dc.w	w44+2
4a3d	e044	819:	dc.w	w45+2
4a3f	f344	820:	dc.w	w46+2
4a41	f844	821:	dc.w	w47+2
4a43	fb44	822:	dc.w	w48+2
4a45	0245	823:	dc.w	w49+2
4a47	0f45	824:	dc.w	w50+2
4a49	1545	825:	dc.w	w51+2
4a4b	3d45	826:	dc.w	w52+2
4a4d	9645	827:	dc.w	w53+2
4a4f	9d45	828:	dc.w	w54+2
4a51	cf47	829:	dc.w	w55+2
4a53	d547	830:	dc.w	w56+2
4a55	c647	831:	dc.w	h01+1
4a57	dd47	832:	dc.w	h02+1
4a59	3844	833:	dc.w	put03+2
4a5b	6344	834:	dc.w	put10+2
4a5d	8744	835:	dc.w	op01+2
4a5f	8d44	836:	dc.w	open03+2
4a61	9a44	837:	dc.w	xio32+2
4a63	a747	838:	dc.w	h03+1
4a65	a044	839:	dc.w	forceout+2
4a67	d244	840:	dc.w	stat02+2
4a69	4345	841:	dc.w	xio36b+2
4a6b	4f45	842:	dc.w	err132+2
4a6d	0000	843:	dc.w	0
4a6f		844: ;		

```

4a6f      845:
4a6f      846: ;-----
4a6f      847: ;prints Routeen macro
4a6f      848: ;this is a slow print to the screen
4a6f      849: ;usage jsr  prints
4a6f      850: ; .byte $9b," string to be printed",$ff
4a6f      851: ;-----
4a6f      852: ;
4a6f 68    853: prints:          pla
4a70 85b0   854:                  sta  $b0
4a72 68    855:                  pla
4a73 85b1   856:                  sta  $b1
4a75 a001   857:                  ldy  #$01
4a77 b1b0   858: .a:              lda  ($b0),y
4a79 c9ff   859:                  cmp  #$ff
4a7b f009   860:                  beq  .b
4a7d 20974a 861:                  jsr  sprint
4a80 20904a 862:                  jsr  .c
4a83 4c774a 863:                  jmp  .a
4a86 20904a 864: .b:              jsr  .c
4a89 a5b1   865:                  lda  $b1
4a8b 48     866:                  pha
4a8c a5b0   867:                  lda  $b0
4a8e 48     868:                  pha
4a8f 60     869:                  rts
4a90 e6b0   870: .c:              inc  $b0
4a92 d002   871:                  bne  .d
4a94 e6b1   872:                  inc  $b1
4a96 60     873: .d:              rts
4a97      874: ;
4a97      875: ;
4a97      876: ;-----
4a97      877: ; PRINT BYTE
4a97      878: ;-----
4a97      879: echo:
4a97 85b8   880: sprint:          sta  $b8
4a99 a904   881:                  lda  #$04
4a9b 8d4a03 882:                  sta  $034a
4a9e 86b6   883:                  stx  $b6
4aa0 84b7   884:                  sty  $b7
4aa2 a206   885:                  ldx  #$06
4aa4 bd01e4 886:                  lda  $e400+1,x
4aa7 85b3   887:                  sta  $b3
4aa9 bd00e4 888:                  lda  $e400,x
4aac 85b2   889:                  sta  $b2
4aae a5b8   890:                  lda  $b8
4ab0 20bf4a 891:                  jsr  .a
4ab3 a6b6   892:                  ldx  $b6
4ab5 a4b7   893:                  ldy  $b7
4ab7 a90c   894:                  lda  #$0c
4ab9 8d4a03 895:                  sta  $034a
4abc a5b8   896:                  lda  $b8
4abe 60     897:                  rts

```

```

4abf      898: ;
4abf e6b2 899: .a:          inc  $b2
4ac1 6cb200 900:          jmp  ($b2)
4ac4 60    901:          rts
4ac5      902:
4ac5      903: test_pbi:
4ac5 20764b 904:          jsr  BLACK_BOX
4ac8 ad6743 905:          lda  pbi_device
4acb c902   906:          cmp  #$02      ; b
4acd f003   907:          beq  .done
4acf 20f14a 908:          jsr  mio_test
4ad2 60    909: .done:          rts
4ad3      910:
4ad3      911:
4ad3      912: ;
4ad3      913: ver_num:
4ad3 48    914:          pha
4ad4 4a    915:          lsr
4ad5 4a    916:          lsr
4ad6 4a    917:          lsr
4ad7 4a    918:          lsr
4ad8 20e34a 919:          jsr  .a
4adb a92e   920:          lda  #'.'
4add 20974a 921:          jsr  echo
4ae0 68    922:          pla
4ae1 290f   923:          and  #$0f
4ae3 c90a   924: .a:          cmp  #$0a
4ae5 b004   925:          bcs  .b
4ae7 0930   926:          ora  #$30
4ae9 d002   927:          bne  .c
4aeb 6936   928: .b:          adc  #$36
4aed 20974a 929: .c:          jsr  echo
4af0 60    930:          rts
4af1      931:
4af1      932: ;
4af1      933:
4af1      934:
4af1      935:
4af1      936:
4af1      937:
4af1      938:
4af1      939: ;;
4af1      940: ;-----
4af1      941: ; This Routen Tests for a MIO in system
4af1      942: ;
4af1      943: ;
4af1      944: MIO_TEST:
4af1 8d714b 945:          sta  mio_acc
4af4 8c724b 946:          sty  mio_y
4af7 8e734b 947:          stx  mio_x
4afa ba    948:          tsx
4afb 8e744b 949:          stx  mio_stack
4afe 78    950:          sei

```

```

4aff      951:
4aff a514  952:          lda    $14
4b01 c514  953: tammy:      cmp    $14
4b03 f0fc  954:          beq    tammy
4b05 38    955:          SEC      ; D
4b06 ade0d1 956:          LDa    Ld1e0
4b09 8d6f4b 957:          STa    M_HOLD
4b0c ade2d1 958:          LDa    Ld1e2
4b0f 8d704b 959:          STa    M_HOLD+1
4b12 a900  960:          LDa    #$00
4b14 8de0d1 961:          STa    Ld1e0
4b17 a920  962:          LDa    #$20
4b19 8de2d1 963:          STa    Ld1e2
4b1c a943  964:          LDa    #'C'
4b1e cd01d6 965:          cmp    pbi_ram+1
4b21 d00a  966:          bne    NO_MIO
4b23 cd0ad6 967:          cmp    pbi_ram+10
4b26 d005  968:          bne    NO_MIO
4b28 a901  969:          lda    #$01
4b2a 8d754b 970:          sta    mio_there
4b2d      971:
4b2d      972: NO_MIO
4b2d a900  973:          lda    #$00      ; t
4b2f 8de0d1 974:          sta    $d1e0
4b32 8de2d1 975:          sta    $d1e2
4b35 58    976:          CLI
4b36      977:
4b36 ae744b 978:          ldx    mio_stack
4b39 9a    979:          txs
4b3a ad714b 980:          lda    mio_acc
4b3d ac724b 981:          ldy    mio_y
4b40 ae734b 982:          ldx    mio_x
4b43 ad754b 983:          lda    mio_there
4b46 c900  984:          cmp    #$00
4b48 f024  985:          beq    _exit
4b4a      986:
4b4a      987:
4b4a 206f4a 988:          jsr    printsi
4b4d 494344 989:          dc.b   "ICD MIO Present in
4b69 a901  990:          lda    #$01
4b6b 8d6743 991:          sta    pbi_device
4b6e      992:
4b6e 60    993: _exit:      rts
4b6f      994:
4b6f      995:
4b6f 0000  996: M_HOLD:      dc.b   0,0
4b71      997:
4b71 00    998: mio_acc:      dc.b   0      ; accumulato
4b72 00    999: mio_y:        dc.b   0      ; Y registe
4b73 00   1000: mio_x:        dc.b   0      ; x registe
4b74 00   1001: mio_stack:    dc.b   0      ; hold the
4b75 00   1002: mio_there:    dc.b   0      ; holding r
4b76      1003: ;

```

```

4b76      1004: ;
4b76      1005: ; Checking on the black box if it is present!
4b76      1006: ;
4b76      1007: BLACK_BOX:
4b76 78    1008:          SEI
4b77 a902  1009:          LDa  #2
4b79 8dc0d1 1010:          STa  BB_Sensel
4b7c a010  1011:          ldy  #$10
4b7e      1012: .a
4b7e a202  1013:          idx  #2
4b80 b900d8 1014:          LDa  bb_rom,Y
4b83 ddcf4b 1015:          cmp  .e,X
4b86 f007  1016:          beq  .b
4b88 c8     1017:          INY
4b89 d0f3  1018:          bne  .a
4b8b 4cc74b 1019:          jmp  .d
4b8e 60     1020:          RTS
4b8f      1021: ;
4b8f      1022: .b
4b8f c8     1023:          INY
4b90 ca     1024:          DEX
4b91 300a  1025:          bmi  .c
4b93 b900d8 1026:          LDa  bb_rom,Y
4b96 ddcf4b 1027:          cmp  .e,X
4b99 d0e3  1028:          bne  .a
4b9b f0f2  1029:          beq  .b
4b9d      1030: .c
4b9d 206f4a 1031:          jsr  printsi
4ba0 9b     1032:          dc.b  $9b
4ba1 426c61 1033:          dc.b  "Black Box IS prese
4bc2 a902  1034:          lda  #$02
4bc4 8d6743 1035:          sta  pbi_device
4bc7      1036: ;
4bc7      1037: .d
4bc7 d8     1038:          CLD
4bc8 a900  1039:          LDa  #0
4bca 8dc0d1 1040:          STa  BB_Sensel
4bcd 58     1041:          CLI
4bce 60     1042:          RTS
4bcf      1043: ;
4bcf d00829 1044: .e          dc.b  $D0,$08,$29,$00
4bd3      1045: ;
4bd3      1046: ;
4bd3      1047: ;
4bd3      1048: ;
4bd3      1049: ; this is stuff for a Hyper R for the black box
4bd3      1050: ;
4bd3      1051: ;
4bd3      1052: ;
4bd3      1053: start.bb:
4bd3      1054:
4bd3      1055: ;Needed by relocater
4bd3 c84c  1056: hatab.bb:          dc.w  open.bb-1

```

```

4bd5 e04c 1057: w01.bb:          dc.w  close.bb-1
4bd7 5c4c 1058: w02.bb:          dc.w  get.bb-1
4bd9 854c 1059: w03.bb:          dc.w  put.bb-1
4bdb f84c 1060: w04.bb:          dc.w  status.bb-1
4bdd 3b4d 1061: w05.bb:          dc.w  special.bb-
4bdf 00 1062:          dc.b  0
4be0 1063:
4be0 00 1064: staterr.bb:        dc.b  0
4be1 00 1065: nextin.bb:          dc.b  0
4be2 00 1066: lastin.bb:          dc.b  0
4be3 00 1067: nextout.bb:         dc.b  0
4be4 00 1068: lastout.bb:         dc.b  0
4be5 00 1069: concur.bb:         dc.b  0
4be6 00 1070: mode.bb:          dc.b  0
4be7 00 1071: trans.bb:         dc.b  0
4be8 1072: ;
4be8 1073: ; "INTERRUPT ROUTINE"
4be8 1074: ;
4be8 48 1075: irq.bb:          pha
4be9 8a 1076:          txa
4bea ae31d1 1077:          ldx  $d131
4bed 3005 1078:          bmi  irq2.bb
4bef aa 1079:          tax
4bf0 68 1080:          pla
4bf1 4c30c0 1081: sysirq.bb:          jmp  $c030
4bf4 48 1082: irq2.bb:          pha
4bf5 98 1083:          tya
4bf6 48 1084:          pha
4bf7 8a 1085:          txa
4bf8 2908 1086:          and  #8
4bfa f01d 1087:          beq  trysend.bb
4bfc ace24b 1088: w06.bb:          ldy  lastin.bb
4bff ad30d1 1089:          lda  $d130
4c02 99f14d 1090: w07.bb:          sta  inbuff.bb,y
4c05 c8 1091:          iny
4c06 8ce24b 1092: w08.bb:          sty  lastin.bb
4c09 cce14b 1093: w09.bb:          cpy  nextin.bb
4c0c d02b 1094:          bne  irqexit.bb
4c0e eee14b 1095: w10.bb:          inc  nextin.bb
4c11 ade04b 1096: w11.bb:          lda  staterr.bb
4c14 0910 1097:          ora  #$10
4c16 8de04b 1098: w12.bb:          sta  staterr.bb
4c19 8a 1099: trysend.bb:        txa
4c1a 2910 1100:          and  #$10
4c1c f01b 1101:          beq  irqexit.bb
4c1e ad32d1 1102:          lda  $d132
4c21 290c 1103:          and  #$0c
4c23 c904 1104:          cmp  #4
4c25 d012 1105:          bne  irqexit.bb
4c27 adc0d1 1106: send02.bb:        lda  $d1c0
4c2a 2940 1107:          and  #$40
4c2c f0f9 1108:          beq  send02.bb
4c2e ace34b 1109: w13.bb:          ldy  nextout.bb

```

```

4c31 cce44b 1110: w14.bb:          cpy  lastout.bb
4c34 d009  1111:          bne  sendit.bb
4c36 204c4c 1112: w15.bb:          jsr  dstxirq.bb
4c39 68    1113: irqexit.bb:      pla
4c3a a8    1114:          tay
4c3b      1115:
4c3b 68    1116:          pla
4c3c aa    1117:          tax
4c3d 68    1118:          pla
4c3e 40    1119:          rti
4c3f c8    1120: sendit.bb:      iny
4c40 b9f14e 1121: w17.bb:          lda  outbuff.bb,
4c43 8d30d1 1122:          sta  $d130
4c46 8ce34b 1123: w18.bb:          sty  nextout.bb
4c49 4c394c 1124: w19.bb:          jmp  irqexit.bb
4c4c a908  1125: dstxirq.bb:      lda  #8
4c4e 2c    1126:          dc.b $2c
4c4f a904  1127: entxirq.bb:      lda  #4
4c51 4d32d1 1128:          eor  $d132
4c54 290c  1129:          and  #$0c
4c56 4d32d1 1130:          eor  $d132
4c59 8d32d1 1131:          sta  $d132
4c5c 60    1132:          rts
4c5d      1133: ;
4c5d      1134: ;    "GET BYTE ROUTINE"
4c5d      1135: ;
4c5d 2411  1136: get.bb:          bit  brkkey
4c5f 101c  1137:          bpl  get10.bb
4c61 aee14b 1138: w20.bb:          ldx  nextin.bb
4c64 ece24b 1139: w21.bb:          cpx  lastin.bb
4c67 f0f4  1140:          beq  get.bb
4c69 bcf14d 1141: w22.bb:          ldy  inbuff.bb,x
4c6c e8    1142:          inx
4c6d 8ee14b 1143: w23.bb:          stx  nextin.bb
4c70 ade74b 1144: w24.bb:          lda  trans.bb
4c73 2920  1145:          and  #$20
4c75 d006  1146:          bne  get10.bb
4c77 c00d  1147:          cpy  #$0d
4c79 d002  1148:          bne  get10.bb
4c7b a09b  1149:          ldy  #$9b
4c7d 98    1150: get10.bb:      tya
4c7e 48    1151:          pha
4c7f 20a14c 1152: w25.bb:          jsr  setbreak.bb
4c82 68    1153:          pla
4c83 c000  1154:          cpy  #0
4c85 60    1155:          rts
4c86      1156: ;
4c86      1157: ;    "PUT BYTE, SET BREAK"
4c86      1158: ;
4c86 a8    1159: put.bb:          tay
4c87 ade74b 1160: w26.bb:          lda  trans.bb
4c8a 2920  1161:          and  #$20
4c8c d010  1162:          bne  put03.bb

```

```

4c8e c09b 1163:      cpy  #$9b
4c90 d00c 1164:      bne  put03.bb
4c92 a00d 1165:      ldy  #$0d
4c94 20af4c 1166: w27.bb:      jsr  put10.bb
4c97 2ce74b 1167: w28.bb:      bit  trans.bb
4c9a 5005 1168:      bvc  setbreak.bb
4c9c a00a 1169:      ldy  #$0a
4c9e 20af4c 1170: put03.bb:      jsr  put10.bb
4ca1 a001 1171: setbreak.bb:      ldy  #1
4ca3 2411 1172:      bit  brkkey
4ca5 3004 1173:      bmi  enirq.bb
4ca7 a080 1174:      ldy  #$80
4ca9 8411 1175:      sty  brkkey
4cab      1176: ENIRQ.bb:
4cab c000 1177:      cpy  #0
4cad 38 1178:      sec
4cae 60 1179:      rts
4caf aee44b 1180: put10.bb:      ldx  lastout.bb
4cb2 e8 1181:      inx
4cb3 2411 1182: put11.bb:      bit  brkkey
4cb5 1011 1183:      bpl  putexit.bb
4cb7 ece34b 1184: w29.bb:      cpx  nextout.bb
4cba f0f7 1185:      beq  put11.bb
4cbc 78 1186:      sei
4cbd 98 1187:      tya
4cbe 9df14e 1188: w30.bb:      sta  outbuff.bb,
4cc1 8ee44b 1189: w31.bb:      stx  lastout.bb
4cc4 204f4c 1190: w32.bb:      jsr  entxirq.bb
4cc7 58 1191:      cli
4cc8 60 1192: putexit.bb:      rts
4cc9      1193: ;
4cc9      1194: ; "OPEN, CLOSE R.bb: CHANNEL"
4cc9      1195: ;
4cc9 a52a 1196: open.bb:      lda  icax1z
4ccb 8de64b 1197: w33.bb:      sta  mode.bb
4cce 78 1198:      sei
4ccf a205 1199:      ldx  #5
4cd1 a900 1200:      lda  #0
4cd3 9de04b 1201: op01.bb:      sta  staterr.bb,x
4cd6 ca 1202:      dex
4cd7 10fa 1203:      bpl  op01.bb
4cd9 ade64b 1204: open03.bb:      lda  mode.bb
4cdc 852a 1205:      sta  icax1z
4cde 4ca14c 1206: w34.bb:      jmp  setbreak.bb
4ce1      1207: ;
4ce1      1208: ; CLOSE R: HANDLER CHANNEL
4ce1      1209: ;
4ce1 a900 1210: close.bb:      lda  #0
4ce3 8de54b 1211: w35.bb:      sta  concur.bb
4ce6 20ec4c 1212: xio32.bb:      jsr  forceout.bb
4ce9 4cd94c 1213: w36.bb:      jmp  open03.bb
4cec ade44b 1214: forceout.bb:      lda  lastout.bb
4cef 2411 1215: f01.bb:      bit  brkkey

```



```

4cf1 1005 1216:          bpl  f02.bb
4cf3 cde34b 1217: w37.bb:          cmp  nexttout.bb
4cf6 d0f7 1218:          bne  f01.bb
4cf8 60 1219: f02.bb:          rts
4cf9 1220: ;
4cf9 1221: ; "GET STATUS OF R: CHANNEL"
4cf9 1222: ;
4cf9 adc0d1 1223: status.bb:          lda  $d1c0
4cfc 4a 1224:          lsr
4cfd 4a 1225:          lsr
4cfe 4a 1226:          lsr
4cff 4a 1227:          lsr
4d00 4a 1228:          lsr
4d01 2907 1229:          and  #7
4d03 aa 1230:          tax
4d04 bd344d 1231: w38.bb:          lda  hsflags.bb,
4d07 8deb02 1232:          sta  dvstat+1
4d0a 2ce54b 1233: w39.bb:          bit  concur.bb
4d0d 1014 1234:          bpl  stat02.bb
4d0f 38 1235:          sec
4d10 ade24b 1236: w40.bb:          lda  lastin.bb
4d13 ede14b 1237: w41.bb:          sbc  nextin.bb
4d16 8deb02 1238:          sta  dvstat+1
4d19 38 1239:          sec
4d1a ade44b 1240: w42.bb:          lda  lastout.bb
4d1d ede34b 1241: w43.bb:          sbc  nextout.bb
4d20 8ded02 1242:          sta  dvstat+3
4d23 ade04b 1243: stat02.bb:          lda  staterr.bb
4d26 8dea02 1244:          sta  dvstat
4d29 a900 1245:          lda  #0
4d2b 8de04b 1246: w44.bb:          sta  staterr.bb
4d2e 8dec02 1247:          sta  dvstat+2
4d31 4ca14c 1248: w45.bb:          jmp  setbreak.bb
4d34 1249:
4d34 1250:
4d34 1251: HSFLAGS.bb:
4d34 01310d 1252:          dc.b $01,$31,$0d,$3d
4d38 c1f1cd 1253:          dc.b $c1,$f1,$cd,$fd
4d3c 1254:
4d3c 1255: ; "XIO 40, 38, 36 CALLS"
4d3c 1256: ;
4d3c a622 1257: special.bb:          ldx  iccomz
4d3e a52a 1258:          lda  icax1z
4d40 e028 1259:          cpx  #$28
4d42 d00b 1260:          bne  xio38.bb
4d44 ade54b 1261: w46.bb:          lda  concur.bb
4d47 0980 1262:          ora  #$80
4d49 8de54b 1263: w47.bb:          sta  concur.bb
4d4c 4cd94c 1264: w48.bb:          jmp  open03.bb
4d4f e026 1265: xio38.bb:          cpx  #$26
4d51 d01a 1266:          bne  xio36.bb
4d53 8de74b 1267: w49.bb:          sta  trans.bb
4d56 4a 1268:          lsr

```

```

4d57 4a 1269:      lsr
4d58 2903 1270:     and  #3
4d5a a8 1271:      tay
4d5b ad32d1 1272:    lda  $d132
4d5e 291f 1273:     and  #$1f
4d60 19694d 1274: w50.bb:      ora  parity.bb,y
4d63 8d32d1 1275:     sta  $d132
4d66 4cd94c 1276: w51.bb:      jmp  open03.bb
4d69      1277:
4d69 002060 1278: parity.bb:      dc.b  0,$20,$60,$a0
4d6d      1279:
4d6d      1280:
4d6d e024 1281: xio36.bb:      cpx  #$24
4d6f d038 1282:      bne  xio34.bb
4d71 a8 1283:      tay
4d72 29d0 1284:     and  #$d0
4d74 0a 1285:      asl
4d75 08 1286:      php
4d76 0a 1287:      asl
4d77 28 1288:      plp
4d78 6a 1289:      ror
4d79 48 1290:      pha
4d7a 98 1291:      tya
4d7b 290f 1292:     and  #$0f
4d7d f00d 1293:     beq  xio36c.bb
4d7f a208 1294:     ldx  #8
4d81 c905 1295:     cmp  #5
4d83 f008 1296:     beq  xio36a.bb
4d85 38 1297:     sec
4d86 c908 1298:     cmp  #8
4d88 9016 1299:     bcc  err132.bb
4d8a 2907 1300:     and  #7
4d8c aa 1301: xio36c.bb:      tax
4d8d 68 1302: xio36a.bb:      pla
4d8e 1d974d 1303: w52.bb:      ora  baud.bb,x
4d91 8d33d1 1304:     sta  $d133
4d94 4cd94c 1305: xio36b.bb:      jmp  open03.bb
4d97 161718 1306: baud.bb:      dc.b  $16,$17,$18,$19
4d9b 1a1c1e 1307:     dc.b  $1a,$1c,$1e,$1f,$13
4da0      1308: ;      "ERROR 132
4da0      1309: ;      XIO 34, 32"
4da0 20a14c 1310: err132.bb:      jsr  setbreak.bb
4da3 98 1311:      tya
4da4 3002 1312:     bmi  er132a.bb
4da6 a084 1313:     ldy  #$84
4da8 60 1314: er132a.bb:      rts
4da9 e022 1315: xio34.bb:      cpx  #$22
4dab d03d 1316:     bne  xio32a.bb
4dad aa 1317:      tax
4dae 100e 1318:     bpl  xio34a.bb
4db0 2940 1319:     and  #$40
4db2 0a 1320:      asl
4db3 0a 1321:      asl

```

```

4db4 08 1322:      php
4db5 ad32d1 1323:    lda  $d132
4db8 4a 1324:      lsr
4db9 28 1325:      plp
4dba 2a 1326:      rol
4dbb 8d32d1 1327:    sta  $d132
4dbe 8a 1328: xio34a.bb:  txa
4dbf 2930 1329:    and  #$30
4dc1 c930 1330:    cmp  #$30
4dc3 f00f 1331:    beq  xio34d.bb
4dc5 c920 1332:    cmp  #$20
4dc7 f011 1333:    beq  xio34c.bb
4dc9 8a 1334:      txa
4dca 2903 1335:    and  #3
4dcc c903 1336:    cmp  #3
4dce f007 1337:    beq  xio34b.bb
4dd0 c902 1338:    cmp  #2
4dd2 d013 1339:    bne  w53.bb
4dd4 a908 1340: xio34d.bb:    lda  #8
4dd6 2c 1341:      dc.b $2c
4dd7 a90c 1342: xio34b.bb:    lda  #$0c
4dd9 2c 1343:      dc.b $2c
4dda a900 1344: xio34c.bb:    lda  #0
4ddc 4d32d1 1345:    eor  $d132
4ddf 290c 1346:    and  #$0c
4de1 4d32d1 1347:    eor  $d132
4de4 8d32d1 1348:    sta  $d132
4de7 4cd94c 1349: w53.bb:      jmp  open03.bb
4dea e020 1350: xio32a.bb:    cpx  #$20
4dec d0b2 1351:    bne  err132.bb
4dee 4ce64c 1352: w54.bb:      jmp  xio32.bb
4df1 1353: inbuff.bb:  ds.b 256
4ef1 1354: outbuff.bb: ds.b 256
4ff1 1355: ;
4ff1 1356: ; "RESET INITIALIZATION"
4ff1 20ffff 1357: init.bb:      jsr  $ffff
4ff4 a93e 1358: init02.bb:    lda  # low relocate.bb
4ff6 8de702 1359:      sta  memlo
4ff9 a950 1360: h03.bb:      lda  # high relo
4ffb 8de802 1361:      sta  memlo+1
4ffe a200 1362:      ldx  #0
5000 bd1a03 1363: init03.bb:    lda  $031a,x
5003 f009 1364:      beq  sethand.bb
5005 c952 1365:      cmp  #'R'
5007 f00a 1366:      beq  set02.bb
5009 e8 1367:      inx
500a e8 1368:      inx
500b e8 1369:      inx
500c d0f2 1370:      bne  init03.bb
500e a952 1371: sethand.bb:   lda  #'R'
5010 9d1a03 1372:      sta  $031a,x
5013 a9d3 1373: set02.bb:     lda  # low hatab.bb
5015 9d1b03 1374:      sta  $031b,x

```

```

5018 a94b 1375: h01.bb:          lda  # high hata
501a 9d1c03 1376:          sta  $031c,x
501d ad1602 1377:          lda  $0216
5020 8df24b 1378: w55.bb:          sta  sysirq.bb+1
5023 ad1702 1379:          lda  $0216+1
5026 8df34b 1380: w56.bb:          sta  sysirq.bb+2
5029 78 1381:          sei
502a a9e8 1382: i02.bb:          lda  # low irq.b
502c 8d1602 1383:          sta  $0216
502f a94b 1384: h02.bb:          lda  # high irq.
5031 8d1702 1385:          sta  $0216+1
5034 58 1386:          cli
5035 ad32d1 1387:          lda  $d132
5038 29fe 1388:          and  #$fe
503a 8d32d1 1389:          sta  $d132
503d 60 1390: inout.bb:        rts
503e 1391: ;
503e 1392: ;   GENERIC RELOCATOR - written for use with other prog
503e 1393: ;
503e 1394: relocate.bb:
503e 20d952 1395:          jsr  set_handler
5041 1396:
5041 a200 1397:          ldx  #0
5043 bd1a03 1398: rel01.bb:        lda  $031a,x
5046 f00a 1399:          beq  rel01a.bb
5048 e8 1400:          inx
5049 e8 1401:          inx
504a e8 1402:          inx
504b e021 1403:          cpx  #32+1
504d 90f4 1404:          bcc  rel01.bb
504f 1405:
504f 4c1249 1406:          jmp  no_instal_error
5052 1407: rel01a.bb:
5052 ad6743 1408:          lda  pbi_device
5055 c902 1409:          cmp  #$02
5057 f003 1410:          beq  .a
5059 4c1249 1411:          jmp  no_instal_error
505c 1412: .a:
505c a9d3 1413:          lda  # low start.bb
505e 38 1414:          sec
505f ede702 1415:          sbc  memlo
5062 8dfc50 1416:          sta  offset.bb
5065 a94b 1417:          lda  # high start.bb
5067 ede802 1418:          sbc  memlo+1
506a 8dfd50 1419:          sta  offset.bb+1
506d a200 1420:          ldx  #0
506f a000 1421:          ldy  #0
5071 bdfc50 1422: rel02.bb:        lda  tablo.bb,x
5074 85e0 1423:          sta  $e0
5076 bd8e51 1424: rel03.bb:        lda  tabhi.bb,x
5079 85e2 1425:          sta  $e2
507b e8 1426:          inx
507c bdfc50 1427: rel04.bb:        lda  tablo.bb,x

```

```

507f 85e1 1428:          sta  $e1
5081 bd8e51 1429: rel05.bb:      lda  tabhi.bb,x
5084 85e3 1430:          sta  $e3
5086 05e2 1431:          ora  $e2
5088 f021 1432:          beq  moveit.bb
508a b1e0 1433:          lda  ($e0),y
508c 38 1434:          sec
508d edfc50 1435:          sbc  offset.bb
5090 91e0 1436:          sta  ($e0),y
5092 b1e2 1437:          lda  ($e2),y
5094 edfd50 1438:          sbc  offset.bb+1
5097 91e2 1439:          sta  ($e2),y
5099 e8 1440:          inx
509a d0d5 1441:          bne  rel02.bb
509c ee7350 1442:          inc  rel02.bb+2
509f ee7850 1443:          inc  rel03.bb+2
50a2 ee7e50 1444:          inc  rel04.bb+2
50a5 ee8350 1445:          inc  rel05.bb+2
50a8 4c7150 1446:          jmp  rel02.bb
50ab 1447:
50ab 1448:
50ab 1449:
50ab 1450: ; "MOVE CODE TO MEMLO"
50ab 1451:
50ab a9d3 1452: moveit.bb:      lda  # low start.bb
50ad 85e0 1453:          sta  $e0
50af a94b 1454:          lda  # high start.bb
50b1 85e1 1455:          sta  $e1
50b3 ade702 1456:          lda  memlo
50b6 85e2 1457:          sta  $e2
50b8 ade802 1458:          lda  memlo+1
50bb 85e3 1459:          sta  $e3
50bd a96b 1460:          lda  # low relocate.bb-s
50bf 85e4 1461:          sta  $e4
50c1 a904 1462:          lda  # high relocate.bb-
50c3 85e5 1463:          sta  $e5
50c5 b1e0 1464: move02.bb:      lda  ($e0),y
50c7 91e2 1465:          sta  ($e2),y
50c9 e6e0 1466:          inc  $e0
50cb d002 1467:          bne  move03.bb
50cd e6e1 1468:          inc  $e1
50cf e6e2 1469: move03.bb:      inc  $e2
50d1 d002 1470:          bne  move04.bb
50d3 e6e3 1471:          inc  $e3
50d5 a5e4 1472: move04.bb:      lda  $e4
50d7 d002 1473:          bne  move05.bb
50d9 c6e5 1474:          dec  $e5
50db c6e4 1475: move05.bb:      dec  $e4
50dd d0e6 1476:          bne  move02.bb
50df a5e5 1477:          lda  $e5
50e1 d0e2 1478:          bne  move02.bb
50e3 a50c 1479: relinit.bb:      lda  $0c
50e5 8df24f 1480: m06.bb:          sta  init.bb+1

```

```

50e8 a50d 1481:          lda  $0d
50ea 8df34f 1482: m07.bb:          sta  init.bb+2
50ed a9f1 1483: m08.bb:          lda  # low init.
50ef 850c 1484:          sta  $0c
50f1 a94f 1485: m09.bb:          lda  # high init
50f3 850d 1486:          sta  $0d
50f5 20f44f 1487: m10.bb:          jsr  init02.bb
50f8      1488:
50f8      1489:
50f8      1490:
50f8 20a548 1491: prtmsg.bb:          jsr  prtmsg
50fb      1492:
50fb 60 1493:          rts
50fc      1494:
50fc      1495:
50fc      1496:
50fc 0000 1497: offset.bb:          dc.w 0
50fe      1498:
50fe      1499:
50fe      1500:
50fe      1501:
50fe      1502:
50fe      1503: ; "RELOCATOR TABLE 1"
50fe      1504: ;
50fe      1505: ;Each TABLO entry MUST have a matching
50fe      1506: ;TABHI entry. TABLO is for the low
50fe      1507: ;bytes to be offset, TABHI is for the
50fe      1508: ;high bytes. TABHI MUST END WITH ZERO
50fe      1509: ;OR THE RELOCATOR WILL NOT KNOW WHEN
50fe      1510: ;TO STOP!
50fe      1511: ;
50fe e650 1512: tablo.bb:          dc.w  m06.bb+1
5100 eb50 1513:          dc.w  m07.bb+1
5102 ee50 1514:          dc.w  m08.bb+1
5104 f650 1515:          dc.w  m10.bb+1
5106 d34b 1516:          dc.w  hatab.bb
5108 d54b 1517:          dc.w  w01.bb
510a d74b 1518:          dc.w  w02.bb
510c d94b 1519:          dc.w  w03.bb
510e db4b 1520:          dc.w  w04.bb
5110 dd4b 1521:          dc.w  w05.bb
5112 fd4b 1522:          dc.w  w06.bb+1
5114 034c 1523:          dc.w  w07.bb+1
5116 074c 1524:          dc.w  w08.bb+1
5118 0a4c 1525:          dc.w  w09.bb+1
511a 0f4c 1526:          dc.w  w10.bb+1
511c 124c 1527:          dc.w  w11.bb+1
511e 174c 1528:          dc.w  w12.bb+1
5120 2f4c 1529:          dc.w  w13.bb+1
5122 324c 1530:          dc.w  w14.bb+1
5124 374c 1531:          dc.w  w15.bb+1
5126 414c 1532:          dc.w  w17.bb+1
5128 474c 1533:          dc.w  w18.bb+1

```

512a	4a4c	1534:	dc.w	w19.bb+1
512c	624c	1535:	dc.w	w20.bb+1
512e	654c	1536:	dc.w	w21.bb+1
5130	6a4c	1537:	dc.w	w22.bb+1
5132	6e4c	1538:	dc.w	w23.bb+1
5134	714c	1539:	dc.w	w24.bb+1
5136	804c	1540:	dc.w	w25.bb+1
5138	884c	1541:	dc.w	w26.bb+1
513a	954c	1542:	dc.w	w27.bb+1
513c	984c	1543:	dc.w	w28.bb+1
513e	b84c	1544:	dc.w	w29.bb+1
5140	bf4c	1545:	dc.w	w30.bb+1
5142	c24c	1546:	dc.w	w31.bb+1
5144	c54c	1547:	dc.w	w32.bb+1
5146	cc4c	1548:	dc.w	w33.bb+1
5148	df4c	1549:	dc.w	w34.bb+1
514a	e44c	1550:	dc.w	w35.bb+1
514c	ea4c	1551:	dc.w	w36.bb+1
514e	f44c	1552:	dc.w	w37.bb+1
5150	054d	1553:	dc.w	w38.bb+1
5152	0b4d	1554:	dc.w	w39.bb+1
5154	114d	1555:	dc.w	w40.bb+1
5156	144d	1556:	dc.w	w41.bb+1
5158	1b4d	1557:	dc.w	w42.bb+1
515a	1e4d	1558:	dc.w	w43.bb+1
515c	2c4d	1559:	dc.w	w44.bb+1
515e	324d	1560:	dc.w	w45.bb+1
5160	454d	1561:	dc.w	w46.bb+1
5162	4a4d	1562:	dc.w	w47.bb+1
5164	4d4d	1563:	dc.w	w48.bb+1
5166	544d	1564:	dc.w	w49.bb+1
5168	614d	1565:	dc.w	w50.bb+1
516a	674d	1566:	dc.w	w51.bb+1
516c	8f4d	1567:	dc.w	w52.bb+1
516e	e84d	1568:	dc.w	w53.bb+1
5170	ef4d	1569:	dc.w	w54.bb+1
5172	2150	1570:	dc.w	w55.bb+1
5174	2750	1571:	dc.w	w56.bb+1
5176	1450	1572:	dc.w	set02.bb+1
5178	2b50	1573:	dc.w	l02.bb+1
517a	9f4c	1574:	dc.w	put03.bb+1
517c	b04c	1575:	dc.w	put10.bb+1
517e	d44c	1576:	dc.w	op01.bb+1
5180	da4c	1577:	dc.w	open03.bb+1
5182	e74c	1578:	dc.w	xio32.bb+1
5184	f54f	1579:	dc.w	init02.bb+1
5186	ed4c	1580:	dc.w	forceout.bb+1
5188	244d	1581:	dc.w	stat02.bb+1
518a	954d	1582:	dc.w	xio36b.bb+1
518c	a14d	1583:	dc.w	err132.bb+1
518e		1584:		
518e		1585:		
518e		1586:		

```

518e      1587: ; "RELOCATOR TABLE 2"
518e      1588: tabhi.bb:
518e e750  1589:          dc.w  m06.bb+2
5190 ec50  1590:          dc.w  m07.bb+2
5192 f250  1591:          dc.w  m09.bb+1
5194 f750  1592:          dc.w  m10.bb+2
5196 d44b  1593:          dc.w  hatab.bb+1
5198 d64b  1594:          dc.w  w01.bb+1
519a d84b  1595:          dc.w  w02.bb+1
519c da4b  1596:          dc.w  w03.bb+1
519e dc4b  1597:          dc.w  w04.bb+1
51a0 de4b  1598:          dc.w  w05.bb+1
51a2 fe4b  1599:          dc.w  w06.bb+2
51a4 044c  1600:          dc.w  w07.bb+2
51a6 084c  1601:          dc.w  w08.bb+2
51a8 0b4c  1602:          dc.w  w09.bb+2
51aa 104c  1603:          dc.w  w10.bb+2
51ac 134c  1604:          dc.w  w11.bb+2
51ae 184c  1605:          dc.w  w12.bb+2
51b0 304c  1606:          dc.w  w13.bb+2
51b2 334c  1607:          dc.w  w14.bb+2
51b4 384c  1608:          dc.w  w15.bb+2
51b6 424c  1609:          dc.w  w17.bb+2
51b8 484c  1610:          dc.w  w18.bb+2
51ba 4b4c  1611:          dc.w  w19.bb+2
51bc 634c  1612:          dc.w  w20.bb+2
51be 664c  1613:          dc.w  w21.bb+2
51c0 6b4c  1614:          dc.w  w22.bb+2
51c2 6f4c  1615:          dc.w  w23.bb+2
51c4 724c  1616:          dc.w  w24.bb+2
51c6 814c  1617:          dc.w  w25.bb+2
51c8 894c  1618:          dc.w  w26.bb+2
51ca 964c  1619:          dc.w  w27.bb+2
51cc 994c  1620:          dc.w  w28.bb+2
51ce b94c  1621:          dc.w  w29.bb+2
51d0 c04c  1622:          dc.w  w30.bb+2
51d2 c34c  1623:          dc.w  w31.bb+2
51d4 c64c  1624:          dc.w  w32.bb+2
51d6 cd4c  1625:          dc.w  w33.bb+2
51d8 e04c  1626:          dc.w  w34.bb+2
51da e54c  1627:          dc.w  w35.bb+2
51dc eb4c  1628:          dc.w  w36.bb+2
51de f54c  1629:          dc.w  w37.bb+2
51e0 064d  1630:          dc.w  w38.bb+2
51e2 0c4d  1631:          dc.w  w39.bb+2
51e4 124d  1632:          dc.w  w40.bb+2
51e6 154d  1633:          dc.w  w41.bb+2
51e8 1c4d  1634:          dc.w  w42.bb+2
51ea 1f4d  1635:          dc.w  w43.bb+2
51ec 2d4d  1636:          dc.w  w44.bb+2
51ee 334d  1637:          dc.w  w45.bb+2
51f0 464d  1638:          dc.w  w46.bb+2
51f2 4b4d  1639:          dc.w  w47.bb+2

```



```

51f4 4e4d 1640:      dc.w  w48.bb+2
51f6 554d 1641:      dc.w  w49.bb+2
51f8 624d 1642:      dc.w  w50.bb+2
51fa 684d 1643:      dc.w  w51.bb+2
51fc 904d 1644:      dc.w  w52.bb+2
51fe e94d 1645:      dc.w  w53.bb+2
5200 f04d 1646:      dc.w  w54.bb+2
5202 2250 1647:      dc.w  w55.bb+2
5204 2850 1648:      dc.w  w56.bb+2
5206 1950 1649:      dc.w  h01.bb+1
5208 3050 1650:      dc.w  h02.bb+1
520a a04c 1651:      dc.w  put03.bb+2
520c b14c 1652:      dc.w  put10.bb+2
520e d54c 1653:      dc.w  op01.bb+2
5210 db4c 1654:      dc.w  open03.bb+2
5212 e84c 1655:      dc.w  xio32.bb+2
5214 fa4f 1656:      dc.w  h03.bb+1
5216 ee4c 1657:      dc.w  forceout.bb+2
5218 254d 1658:      dc.w  stat02.bb+2
521a 964d 1659:      dc.w  xio36b.bb+2
521c a24d 1660:      dc.w  err132.bb+2
521e 0000 1661:      dc.w  0
5220      1662: ;
5220      1663:
5220      1664:
5220      1665:
5220      1666: ;~~~~~
5220      1667: ; OS on and off ROUTEEN
5220      1668: ;~~~~~
5220      1669: ;
5220      1670:
5220      1671: os_off:
5220 78 1672:      SEI
5221 ad01d3 1673:      LDA  PORTB
5224 29fe 1674:      AND  #$FE
5226 8d01d3 1675:      STA  PORTB
5229 58 1676:      CLI
522a 60 1677:      RTS
522b      1678: ;
522b      1679: os_on:
522b 78 1680:      SEI
522c ad01d3 1681:      LDA  PORTB
522f 0901 1682:      ORA  #1
5231 8d01d3 1683:      STA  PORTB
5234 58 1684:      CLI
5235 60 1685:      RTS
5236      1686: ;~~~~~
5236      1687: ;checks user security level pointed to by S_PTR to see if
5236      1688: ;bit specified in the A register is ON. Returns MINus if
5236      1689: ;user doesn't have the specified security level.
5236      1690: ;~~~~~
5236      1691:
5236 8c9152 1692: seclvl:      sty  .tempy

```

```

5239 8e9252 1693:      stx  .tempx
523c c900  1694:      cmp  #0
523e f042  1695:      beq  .passed
5240 a8     1696:      tay
5241 a980  1697:      lda  #128
5243 8d9352 1698:      sta  .temp
5246 a900  1699:      lda  #0
5248 8d9452 1700:      sta  .temp+1
524b 8d9552 1701:      sta  .temp+2
524e 8d9652 1702:      sta  .temp+3
5251 88     1703:      dey
5252 f011  1704:      beq  .mask
5254 4e9352 1705: .shloop:      lsr  .temp
5257 6e9452 1706:      ror  .temp+1
525a 6e9552 1707:      ror  .temp+2
525d 6e9652 1708:      ror  .temp+3
5260 88     1709:      dey
5261 d0f1  1710:      bne  .shloop
5263 a000  1711:      ldy  #0
5265 b99352 1712: .mask:      lda  .temp,y
5268 3182  1713:      and  (s_ptr),y
526a 999352 1714:      sta  .temp,y
526d c8     1715:      iny
526e c004  1716:      cpy  #4
5270 d0f3  1717:      bne  .mask
5272 ad9352 1718: .check:      lda  .temp
5275 0d9452 1719:      ora  .temp+1
5278 0d9552 1720:      ora  .temp+2
527b 0d9652 1721:      ora  .temp+3
527e c900  1722:      cmp  #0
5280 f004  1723:      beq  .failed
5282 a900  1724: .passed:      lda  #0
5284 f002  1725:      beq  .return
5286 a980  1726: .failed:      lda  #$80
5288 08     1727: .return:      php
5289 ac9152 1728:      ldy  .tempy
528c ae9252 1729:      ldx  .tempx
528f 28     1730:      plp
5290 60     1731:      rts
5291      1732:
5291 00     1733: .tempy:      dc.b  0
5292 00     1734: .tempx:      dc.b  0
5293 000000 1735: .temp:      dc.b  0,0,0,0
5297      1736:
5297      1737:
5297      1738: ; sets the bit specified (in the A register) of the se
5297      1739: ; level pointed to by S_PTR to a TRUE value.
5297      1740: ; ~~~~~
5297      1741:
5297 8cd352 1742: setlvl:      sty  .tempy
529a 8ed452 1743:      stx  .tempx
529d c900  1744:      cmp  #0
529f f031  1745:      beq  .return

```

```

52a1 a8 1746:          tay
52a2 a980 1747:         lda #128
52a4 8dd552 1748:         sta .temp
52a7 a900 1749:         lda #0
52a9 8dd652 1750:         sta .temp+1
52ac 8dd752 1751:         sta .temp+2
52af 8dd852 1752:         sta .temp+3
52b2 88 1753:          dey
52b3 f011 1754:         beq .mask
52b5 4ed552 1755: .shloop:      lsr .temp
52b8 6ed652 1756:         ror .temp+1
52bb 6ed752 1757:         ror .temp+2
52be 6ed852 1758:         ror .temp+3
52c1 88 1759:          dey
52c2 d0f1 1760:         bne .shloop
52c4 a000 1761:         ldy #0
52c6 b9d552 1762: .mask:      lda .temp,y
52c9 1182 1763:         ora (s_ptr),y
52cb 9182 1764:         sta (s_ptr),y
52cd c8 1765:          iny
52ce c004 1766:         cpy #4
52d0 d0f4 1767:         bne .mask
52d2 60 1768: .return:     rts
52d3 1769:
52d3 00 1770: .tempy:      dc.b 0
52d4 00 1771: .tempx:      dc.b 0
52d5 000000 1772: .temp:      dc.b 0,0,0,0
52d9 1773:
52d9 1774: ; ~~~~~
52d9 1775: set_handler:
52d9 202052 1776:          jsr os_off
52dc 1777:          mea tsr,s_ptr
52dc a9f5 ----:      lda #low tsr
52de 8582 ----:      sta s_ptr
52e0 a9ff ----:      lda #high tsr
52e2 8583 ----:      sta s_ptr + 1
52e4 a91c 1778:          lda #hyp_r
52e6 209752 1779:          jsr setlvl
52e9 202b52 1780:          jsr os_on
52ec 60 1781:          rts
52ed 1782:
52ed 1783:
52ed 1784:
52ed 1785: win_end:      ds.b 0
52ed 1786: ;
52ed 1787: ;

```

End assembly: no errors