


```

0000 1: ;-----
0000 2: ;      Copyright 2010 Integrated Logic Systems
0000 3: ;      Source Code is Copyright Stephen J. Car
0000 4: ;
0000 5: ;
0000 6: ;      Please do not share this source!
0000 7: ;-----
0000 8: ;      CRCcheck.com
0000 9: ;
0000 10: ; This is a program that does a 16 bit CRC that was e
0000 11: ; to verify that the com file has not had a single by
0000 12: ; it was compiled and sent out.
0000 13: ;
0000 14: ;
0000 15: ;
0000 16: ;      init  2/24/2010      sjc new design
0000 17: ;      added hex delimiter      2/25/2010
0000 18: ;      added new code for registered user 2/26/201
0000 19: ; Updated the screen Print Utility so it will work wi
0000 20: ; This change will also help "Print P:"      6/1
0000 21: ;
0000 22: ;Also added code for Control+1 pause of screen to the high
0000 23: ;handling.      8/24/2012      sjc
0000 24: ;
0000 25: ;
0000 26: ;
0000 27: ;
0000 28: ;
0000 29: ;
0000 30: ;
0000 31: ;
0000 32: ;-----
0000 33: ; Notes: o This source code MAY NOT be placed for download
0000 34: ;      o "mea" is a macro that loads the address of the
0000 35: ;      into the pointer specified by the second field.
0000 36: ;-----
0000 37: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 38: ;      XASM  st and IBM VERSIONS
0000 39: ;-----
0000 40:
0000 41: com_file_name:      .macro
0000 42: ;      dc.b  13,"filename.ext",$9b
0000 43: ;      dc.b  12,"Crccheck.com",$9b
0000 44: ;      .endm
0000 45:
0000 46: ;
0000 47: ;~~~~~
0000 48: ; File revision history. Version Number in hex
0000 49: ;
0000 50: file_ver:      equ  $14
0000 51: ;~~~~~
0000 52: ; the Month in dec for the first time this revision
0000 53: ;was compiled

```

```

0000 54: ;
0000 55: c_month:      equ 8
0000 56: ;~~~~~
0000 57: ; the day in dec for the first time this revision
0000 58: ;was compiled
0000 59: c_day:        equ 27
0000 60: ;~~~~~
0000 61: ; the year in dec for the first time this revision
0000 62: ;was compiled
0000 63: c_year:       equ 2012
0000 64: ;~~~~~
0000 65: ; Type of compiler used in program
0000 66: ;
0000 67: ; 0 = unknown Compiler
0000 68: ; 1 = xasm
0000 69: ; 2 = mac_65
0000 70: ; 3 = basic
0000 71: ; 4 = compiled basic xl
0000 72: ; 5 = C65
0000 73: ; 6 = Action
0000 74: ;
0000 75: ; We can add more as time goes on!
0000 76: ;
0000 77: ;
0000 78: ;
0000 79: xasm:         equ 1
0000 80: mac_65:       equ 2
0000 81: basic:        equ 3
0000 82: basicxl:      equ 4
0000 83: c65:         equ 5
0000 84: action:       equ 6
0000 85:
0000 86: file_compiler: equ xasm
0000 87: ;~~~~~
0000 88: ; is this relocatable code ?
0000 89: ; anyother value other than 1 or 2 would be unknown
0000 90: r..yes:       equ 1
0000 91: r..no:        equ 2
0000 92: ;
0000 93: ;~~~~~
0000 94: ; Gotta define if it can be relocated
0000 95: l_relocatable: equ r..no
0000 96: ;
0000 97: ;
0000 98: ;~~~~~
0000 99: ; Gotta define our Header control Byte
0000 100: ;$ff =bypass intro
0000 101: ;$15 = Bypass Dos check
0000 102: r..crl:      equ $7d
0000 103: r..crlf:     equ $9b
0000 104: r..space:    equ $20
0000 105: r..bypassDOS: equ $15
0000 106: r..bypass:   equ $ff

```

```

0000    107: ;
0000    108: l_frstscreenbyte: equ r..space
0000    109: ;~~~~~
0000    110: ; The language the output file is in.
0000    111: ;
0000    112: ;
0000    113: ; 0 = Undefined
0000    114: ; 1 = English
0000    115: ; 2 = German
0000    116: ;
0000    117: r..language: equ 1
0000    118: ;~~~~~
0000    119:
0000    120: ;
0000    121: ;
0000    122: buff_l: equ $9fff-copy_buff
0000    123: ;
0000    124:
0000    125:
0000    126: .include equates
0000    127: .include globals
0000    128: .include macros
0000    129:
4000    130: .org $4000
4000    131:
4000    132: win_start:
4000    133: header_info:
4000    134: .include header
42a7    135: readcli:
42a7    136:
42a7 a200 137: ldx #$00
42a9 a03f 138: ldy #LBUF
42ab b10a 139: .a: lda (comtab),y
42ad c920 140: cmp #' '
42af f00b 141: beq .gotspacechar
42b1 c99b 142: cmp #$9b
42b3 f004 143: beq .noparmerror
42b5    144:
42b5 c8 145: iny
42b6 4cab42 146: jmp .a
42b9    147:
42b9    148:
42b9 4c8343 149: .noparmerror: jmp usage_error ; based on
42bc    150:
42bc a200 151: .gotspacechar: ldx #$00
42be    152:
42be c8 153: .lop: iny
42bf e8 154: inx
42c0 b10a 155: lda (comtab),y
42c2 9ddb44 156: sta parm_1,x
42c5 8edb44 157: stx parm_1
42c8 c99b 158: cmp #$9b
42ca d0f2 159: bne .lop

```

```

42cc 4cbf43 160:          jmp    top
42cf      161:
42cf      162: ;
42cf      163: filename:
42cf 000000 164:          dc.b  0,0,0,0,0,0,0,0,0,0
42d9 000000 165:          dc.b  0,0,0,0,0,0,0,0,0,0
42e3 000000 166:          dc.b  0,0,0,0,0,0,0,0,0,0
42ed 000000 167:          dc.b  0,0,0,0,0,0,0,0,0,0
42f7 000000 168:          dc.b  0,0,0,0,0,0,0,0,0,0
4301 000000 169:          dc.b  0,0,0,0,0,0,0,0,0,0
430b 000000 170:          dc.b  0,0,0,0,0,0,0,0,0,0
4315 000000 171:          dc.b  0,0,0,0,0,0,0,0,0,0
431f 000000 172:          dc.b  0,0,0,0,0,0,0,0,0,0
4329 000000 173:          dc.b  0,0,0,0,0,0,0,0,0,0
4333 000000 174:          dc.b  0,0,0,0,0,0,0,0,0,0
433d 000000 175:          dc.b  0,0,0,0,0,0,0,0,0,0
4347 000000 176:          dc.b  0,0,0,0,0,0,0,0,0,0
4351 000000 177:          dc.b  0,0,0,0,0,0,0,0,0,0
435b 000000 178:          dc.b  0,0,0,0,0,0,0,0,0,0
4365 000000 179:          dc.b  0,0,0,0,0,0,0,0,0,0
436f 000000 180:          dc.b  0,0,0,0,0,0,0,0,0,0
4379 000000 181:          dc.b  0,0,0,0,0,0,0,0,0,0
4383      182: usage_error:
4383 206b48 183:          jsr    printsi
4386 9b9b   184:          dc.b  $9b,$9b
4388 53796e 185:          dc.b  "Syntax Error!"
4395 9b     186:          dc.b  $9b
4396 557361 187:          dc.b  "Usage: Crccheck Dn:>path>f
43bb 9b     188:          dc.b  $9b
43bc 9b     189:          dc.b  $9b
43bd ff     190:          dc.b  -1
43be 60     191:          rts
43bf      192:
43bf      193: ;
43bf      194: top:
43bf      195: ;          lda    s_node
43bf      196: ;          sta    d_node
43bf      197: ;          lda    s_node+1
43bf      198: ;          sta    d_node+1
43bf      199: ;
43bf      200: ;      Copy the file in source_path/source_file to dest_p
43bf      201: ;      -----
43bf      202: copy_file:
43bf      203: ;
43bf      204:          mea    parm_1,s_ptr
43bf a9db   ----:          lda    #low_parm_1
43c1 8582   ----:          sta    s_ptr
43c3 a944   ----:          lda    #high_parm_1
43c5 8583   ----:          sta    s_ptr + 1
43c7      205:          mea    _infile,d_ptr
43c7 a93b   ----:          lda    #low_infile
43c9 8584   ----:          sta    d_ptr
43cb a944   ----:          lda    #high_infile

```

```

43cd 8585 ----:      sta  d_ptr + 1
43cf 202b45 206:      jsr  toascii
43d2      207:      mea  _infile,fn_ptr
43d2 a93b ----:      lda  #low _infile
43d4 8586 ----:      sta  fn_ptr
43d6 a944 ----:      lda  #high _infile
43d8 8587 ----:      sta  fn_ptr + 1
43da a220 208:      ldx  #$20
43dc 209f4b 209:      jsr  open_in
43df 3057 210:      bmi  _error
43e1      211: _srcok:
43e1      212:
43e1      213:
43e1      214: _copyloop:
43e1 a220 215:      ldx  #$20
43e3 20f147 216:      jsr  blockread
43e6 0060 217:      dc.w  copy_buff
43e8 179c 218:      dc.w  buff_l
43ea 1014 219:      bpl  _readok
43ec c088 220:      cpy  #136
43ee f010 221:      beq  _readok
43f0 8c3744 222: _shutdown:  sty  _savey      ;copy error
43f3 a220 223:      ldx  #$20      ; baby down
43f5 20ea4b 224:      jsr  close
43f8 ac3744 225:      ldy  _savey
43fb 4ccf4a 226:      jmp  next_run
43fe 60 227:      rts
43ff      228: ;
43ff 00 229: first_call:  dc.b 0
4400      230: ;
4400      231: _readok:
4400 adff43 232:      lda  first_call
4403 c900 233:      cmp  #$00
4405 d006 234:      bne  _wer
4407 202c4a 235:      jsr  set_pass
440a eeff43 236:      inc  first_call
440d      237: ;
440d      238: _wer:
440d a220 239:      ldx  #$20
440f bd4803 240:      lda  icbll,x
4412 8d2f44 241:      sta  _totbytes
4415 bd4903 242:      lda  icblh,x
4418 8d3044 243:      sta  _totbytes+1
441b 204d4a 244:      jsr  wipe
441e      245:
441e      246:
441e ad2f44 247:      lda  _totbytes
4421 c917 248:      cmp  #buff_l & $ff      ;di
4423 d00c 249:      bne  _done      ;no
4425 ad3044 250:      lda  _totbytes+1
4428 c99c 251:      cmp  #buff_l / 256
442a d005 252:      bne  _done
442c 4ce143 253:      jmp  _copyloop

```

```

442f      254: ;
442f 179c 255: _totbytes:      dc.w  buff_1
4431 a001 256: _done:          ldy  #1
4433 4cf043 257:          jmp  _shutdown
4436 00    258: _idx:          dc.b  0
4437 00    259: _savey:       dc.b  0
4438 4c5245 260: _error:       jmp  system_error
443b      261:
443b      262: _infile:
443b 000000 263:          dc.b  0,0,0,0,0,0,0,0,0,0
4445 000000 264:          dc.b  0,0,0,0,0,0,0,0,0,0
444f 000000 265:          dc.b  0,0,0,0,0,0,0,0,0,0
4459 000000 266:          dc.b  0,0,0,0,0,0,0,0,0,0
4463 000000 267:          dc.b  0,0,0,0,0,0,0,0,0,0
446d 000000 268:          dc.b  0,0,0,0,0,0,0,0,0,0
4477 000000 269:          dc.b  0,0,0,0,0,0,0,0,0,0
4481 000000 270:          dc.b  0,0,0,0,0,0,0,0,0,0
448b      271: _outfile:
448b 000000 272:          dc.b  0,0,0,0,0,0,0,0,0,0
4495 000000 273:          dc.b  0,0,0,0,0,0,0,0,0,0
449f 000000 274:          dc.b  0,0,0,0,0,0,0,0,0,0
44a9 000000 275:          dc.b  0,0,0,0,0,0,0,0,0,0
44b3 000000 276:          dc.b  0,0,0,0,0,0,0,0,0,0
44bd 000000 277:          dc.b  0,0,0,0,0,0,0,0,0,0
44c7 000000 278:          dc.b  0,0,0,0,0,0,0,0,0,0
44d1 000000 279:          dc.b  0,0,0,0,0,0,0,0,0,0
44db      280:
44db      281:
44db      282:
44db      283: parm_1:      ds.b  80
452b      284:
452b      285: ;   Make an ASCII ($9b terminated) string out of a lth-b
452b      286: ;   ~~~~~
452b      287:
452b a000 288: toascii:     ldy  #0
452d b182 289:          lda  (s_ptr),y      ;ge
452f 8d5145 290:          sta  .lth
4532 f019 291:          beq  .null
4534 c8    292:          iny
4535 ee5145 293:          inc  .lth
4538 b182 294: .loop:      lda  (s_ptr),y
453a 88    295:          dey
453b 9184 296:          sta  (d_ptr),y
453d c8    297:          iny
453e c8    298:          iny
453f cc5145 299:          cpy  .lth
4542 d0f4 300:          bne  .loop
4544 ac5145 301: .done:      ldy  .lth
4547 88    302:          dey
4548 a99b 303: .rtn:      lda  #$9b
454a 9184 304:          sta  (d_ptr),y
454c 60    305:          rts
454d a000 306: .null:      ldy  #0

```

```

454f f0f7 307:          beq  .rtn
4551      308:
4551 01   309: .lth:          dc.b  1
4552      310:
4552      311: ;   Display error message of flag in Y register
4552      312: ;   ~~~~~
4552      313:
4552 8c7a45 314: system_error:  sty  .hold
4555 206b48 315:          jsr  printsi
4558 9b9b 316:          dc.b  $9b,$9b
455a 537973 317:          dc.b  "System Error: ",-1
4569 ad7a45 318:          lda  .hold
456c 207b45 319:          jsr  pr_byte
456f 206b48 320:          jsr  printsi
4572 9bff 321:          dc.b  $9b,-1
4574 a220 322:          ldx  #$20
4576 20ea4b 323:          jsr  close
4579 60   324:          rts
457a      325:
457a      326:
457a 00   327: .hold:          dc.b  0
457b      328:
457b      329:
457b      330: ;~~~~~
457b      331: ;This creates an length byte string and then
457b      332: ;prints it to the screen.
457b      333: ;
457b      334: ;
457b      335: ;
457b      336: pr_byte:
457b a200 337:          ldx  #$00
457d      338: pr_card:
457d 85d4 339:          sta  fr0
457f 86d5 340:          stx  fr0+1
4581 a900 341:          lda  #$00
4583 85f2 342:          sta  cix
4585 8dab45 343:          sta  pr_data
4588 20aad9 344:          jsr  ifp
458b 20e6d8 345:          jsr  fasc
458e a000 346:          ldy  #$00
4590      347: .loop:
4590 eeab45 348:          inc  pr_data
4593 b1f3 349:          lda  ($f3),y
4595 08   350:          php
4596 297f 351:          and  #$7f
4598 99ac45 352:          sta  pr_data+1,y
459b c8   353:          iny
459c      354: ;          jsr  echo
459c 28   355:          plp
459d 10f1 356:          bpl  .loop
459f      357:          mea  pr_data,pr_ptr
459f a9ab ----:          lda  #low pr_data
45a1 8588 ----:          sta  pr_ptr

```



```

45a3 a945 ----:      lda  #high pr_data
45a5 8589 ----:      sta  pr_ptr + 1
45a7 20f24b 358:      jsr  echos
45aa 60 359:          rts
45ab 360:
45ab 361: pr_data:      ds.b 10
45b5 362:
45b5 363:
45b5 364:
45b5 365:
45b5 366: ;      Expand wildcarded name to create an output filename
45b5 367: ;      Input is source_file, dest_file; output is out_name
45b5 368: ;      -----
45b5 369: ;
45b5 370: ;
45b5 371: ;
45b5 372: put256_out:
45b5 a000 373:          ldy  #0
45b7 8ce747 374:          sty  crc_sum
45ba 8ce847 375:          sty  crc_sum+1
45bd 8ce947 376:          sty  chk_sum
45c0 377: ;
45c0 378: crc_loop:
45c0 b188 379:          lda  (pr_ptr),y
45c2 48 380:          pha
45c3 18 381:          clc
45c4 6de947 382:          adc  chk_sum      ;calc stand
45c7 8de947 383:          sta  chk_sum
45ca 68 384:          pla
45cb 4de847 385:          eor  crc_sum+1      ;now do the
45ce aa 386:          tax
45cf bde146 387:          lda  crc_hitable,x
45d2 4de747 388:          eor  crc_sum
45d5 8de847 389:          sta  crc_sum+1
45d8 bde145 390:          lda  crc_lutable,x
45db 4900 391:          eor  #0
45dd 8de747 392:          sta  crc_sum
45e0 60 393:          rts
45e1 394: ;
45e1 395: ;
45e1 396: ;
45e1 397: crc_lutable:
45e1 002142 398:          dc.b  $00,$21,$42,$63,$84,$a5,$c6
45eb 4a6b8c 399:          dc.b  $4a,$6b,$8c,$ad,$ce,$ef,$31
45f5 b594f7 400:          dc.b  $b5,$94,$f7,$d6,$39,$18,$7b
45ff ffd6e2 401:          dc.b  $ff,$de,$62,$43,$20,$01,$e6
4609 6a4b28 402:          dc.b  $6a,$4b,$28,$09,$ee,$cf,$ac
4613 1130d7 403:          dc.b  $11,$30,$d7,$f6,$95,$b4,$5b
461d dffe9d 404:          dc.b  $df,$fe,$9d,$bc,$c4,$e5,$86
4627 0223cc 405:          dc.b  $02,$23,$cc,$ed,$8e,$af,$48
4631 f5d4b7 406:          dc.b  $f5,$d4,$b7,$96,$71,$50,$33
463b bf9e79 407:          dc.b  $bf,$9e,$79,$58,$3b,$1a,$a6
4645 220360 408:          dc.b  $22,$03,$60,$41,$ae,$8f,$ec

```

```

464f 684997 409:      dc.b  $68,$49,$97,$b6,$d5,$f4,$13
4659 9fbedd 410:      dc.b  $9f,$be,$dd,$fc,$1b,$3a,$59
4663 caeb0c 411:      dc.b  $ca,$eb,$0c,$2d,$4e,$6f,$80
466d 042546 412:      dc.b  $04,$25,$46,$67,$b9,$98,$fb
4677 7f5eb1 413:      dc.b  $7f,$5e,$b1,$90,$f3,$d2,$35
4681 eacba8 414:      dc.b  $ea,$cb,$a8,$89,$6e,$4f,$2c
468b a08166 415:      dc.b  $a0,$81,$66,$47,$24,$05,$db
4695 5f7e1d 416:      dc.b  $5f,$7e,$1d,$3c,$d3,$f2,$91
469f 15344c 417:      dc.b  $15,$34,$4c,$6d,$0e,$2f,$c8
46a9 446506 418:      dc.b  $44,$65,$06,$27,$c0,$e1,$82
46b3 3f1ef9 419:      dc.b  $3f,$1e,$f9,$d8,$bb,$9a,$75
46bd f1d0b3 420:      dc.b  $f1,$d0,$b3,$92,$2e,$0f,$6c
46c7 e8c926 421:      dc.b  $e8,$c9,$26,$07,$64,$45,$a2
46d1 1f3e5d 422:      dc.b  $1f,$3e,$5d,$7c,$9b,$ba,$d9
46db 557493 423:      dc.b  $55,$74,$93,$b2,$d1,$f0
46e1      424: crc_hitable:
46e1 001020 425:      dc.b  $00,$10,$20,$30,$40,$50,$60
46ee d1e1f1 426:      dc.b  $d1,$e1,$f1,$12,$02,$32,$22
46fb b3a3d3 427:      dc.b  $b3,$a3,$d3,$c3,$f3,$e3,$24
4708 54a5b5 428:      dc.b  $54,$a5,$b5,$85,$95,$e5,$f5
4715 766656 429:      dc.b  $76,$66,$56,$46,$b7,$a7,$97
4722 586878 430:      dc.b  $58,$68,$78,$08,$18,$28,$38
472f a9b95a 431:      dc.b  $a9,$b9,$5a,$4a,$7a,$6a,$1a
473c eb9b8b 432:      dc.b  $eb,$9b,$8b,$bb,$ab,$6c,$7c
4749 edfdcd 433:      dc.b  $ed,$fd,$cd,$dd,$ad,$bd,$8d
4756 2e1e0e 434:      dc.b  $2e,$1e,$0e,$ff,$ef,$df,$cf
4763 b1a1d1 435:      dc.b  $b1,$a1,$d1,$c1,$f1,$e1,$10
4770 608393 436:      dc.b  $60,$83,$93,$a3,$b3,$c3,$d3
477d 425262 437:      dc.b  $42,$52,$62,$72,$b5,$a5,$95
478a 241404 438:      dc.b  $24,$14,$04,$74,$64,$54,$44
4797 c7d726 439:      dc.b  $c7,$d7,$26,$36,$06,$16,$66
47a4 e99989 440:      dc.b  $e9,$99,$89,$b9,$a9,$58,$48
47b1 cbdbeb 441:      dc.b  $cb,$db,$eb,$fb,$8b,$9b,$ab
47be 1a2a3a 442:      dc.b  $1a,$2a,$3a,$fd,$ed,$dd,$cd
47cb 5c4c3c 443:      dc.b  $5c,$4c,$3c,$2c,$1c,$0c,$ef
47d8 9f6e7e 444:      dc.b  $9f,$6e,$7e,$4e,$5e,$2e,$3e
47e1      445: ;
47e1      446: ;
47e1      447: ;
47e1      448: ;
47e1 00    449: cksum_lo:      dc.b  0
47e2 00    450: cksum_hi:      dc.b  0
47e3 0000  451: crcsize:      dc.w  0
47e5 0000  452: crctemp:      dc.w  0
47e7 0000  453: crc_sum:      dc.w  0
47e9 0000  454: chk_sum:      dc.w  0
47eb      455: ;
47eb 0000  456: real_sum:      dc.w  0 ; real crc check sum
47ed 0000  457: n_split:      dc.w  0 ; split of node numbe
47ef 0000  458: n_split1:     dc.w  0 ; split of node numbe
47f1      459:
47f1      460:
47f1      461: ; Blockread from disk channel in X register.

```

```

47f1      462: ;
47f1      463: ;   call: jsr   blockread
47f1      464: ;           dc.w   buffer_address
47f1      465: ;           dc.w   number_of_bytes
47f1      466: ;   -----
47f1      467: ;
47f1      468: ;
47f1 a907  469: blockread:      lda   #ccgetch
47f3 9d4203 470:           sta   iccom,x
47f6 4c0148 471:           jmp   blockio
47f9 a90b  472: blockwrite:      lda   #ccputch
47fb 9d4203 473:           sta   iccom,x
47fe 4c0148 474:           jmp   blockio
4801 68    475: blockio:        pla
4802 8588  476:           sta   pr_ptr
4804 68    477:           pla
4805 8589  478:           sta   pr_ptr+1
4807 e688  479:           inc   pr_ptr
4809 d002  480:           bne   Anovf
480b e689  481:           inc   pr_ptr+1
480d a000  482: Anovf:          ldy   #0
480f b188  483:           lda   (pr_ptr),y
4811 9d4403 484:           sta   icbal,x
4814 c8    485:           iny
4815 b188  486:           lda   (pr_ptr),y
4817 9d4503 487:           sta   icbah,x
481a c8    488:           iny
481b b188  489:           lda   (pr_ptr),y
481d 9d4803 490:           sta   icblh,x
4820 c8    491:           iny
4821 b188  492:           lda   (pr_ptr),y
4823 9d4903 493:           sta   icblh,x
4826 a588  494:           lda   pr_ptr
4828 18    495:           clc
4829 6903  496:           adc   #3
482b 8588  497:           sta   pr_ptr
482d a589  498:           lda   pr_ptr+1
482f 6900  499:           adc   #0
4831 48    500:           pha
4832 a588  501:           lda   pr_ptr
4834 48    502:           pha
4835 bd4803 503:           lda   icblh,x      ;make sure
4838 1d4903 504:           ora   icblh,x      ; zero byte
483b f003  505:           beq   Aoops
483d 4c56e4 506:           jmp   ciov
4840 a001  507: Aoops:          ldy   #1      ;signal success and
4842 60    508:           rts
4843      509:
4843      510: ;-----
4843      511: ;prints Routine macro
4843      512: ;this is a slow print to the screen
4843      513: ;usage jsr   prints
4843      514: ;   .byte $9b," string to be printed",$ff

```

```

4843      515: ;-----
4843      516: ;
4843 68    517: sprints:      pla
4844 85b0   518:          sta  $b0
4846 68    519:          pla
4847 85b1   520:          sta  $b1
4849 a001   521:          ldy  #$01
484b b1b0   522: .a:         lda  ($b0),y
484d c9ff   523:          cmp  #$ff
484f f009   524:          beq  .b
4851 20c148 525:          jsr  sprint
4854 206448 526:          jsr  .c
4857 4c4b48 527:          jmp  .a
485a 206448 528: .b:         jsr  .c
485d a5b1   529:          lda  $b1
485f 48     530:          pha
4860 a5b0   531:          lda  $b0
4862 48     532:          pha
4863 60     533:          rts
4864 e6b0   534: .c:         inc  $b0
4866 d002   535:          bne  .d
4868 e6b1   536:          inc  $b1
486a 60     537: .d:         rts
486b      538: ;
486b      539: ;
486b      540: ;-----
486b      541: ;prints Routeen macro
486b      542: ;this is a slow print to the screen
486b      543: ;usage jsr  prints
486b      544: ; .byte $9b," string to be printed", $ff
486b      545: ;-----
486b      546: ;
486b      547: prints:
486b 68     548:          pla
486c 8d7c48 549:          sta  .pstr+1
486f 68     550:          pla
4870 8d7d48 551:          sta  .pstr+2
4873 ee7c48 552: .prsl:     inc  .pstr+1
4876 d003   553:          bne  .pstr
4878 ee7d48 554:          inc  .pstr+2
487b adffff 555: .pstr:     lda  $ffff
487e c9ff   556:          cmp  #$ff
4880 f006   557:          beq  .estri
4882 20c148 558:          jsr  fast_output
4885 4c7348 559:          jmp  .prsl
4888 ad7d48 560: .estri:    lda  .pstr+2
488b 48     561:          pha
488c ad7c48 562:          lda  .pstr+1
488f 48     563:          pha
4890 60     564:          rts
4891      565: ;
4891      566: ;
4891      567: ;

```

```

4891      568: ;
4891      569: ;-----
4891      570: ;actual screen handler
4891      571: ;screen offset definitions
4891      572: ;-----
4891      573: scr_offset:
4891 0000 574:          dc.w 0
4893 2800 575:          dc.w 40
4895 5000 576:          dc.w 80
4897 7800 577:          dc.w 120
4899 a000 578:          dc.w 160
489b c800 579:          dc.w 200
489d f000 580:          dc.w 240
489f 1801 581:          dc.w 280
48a1 4001 582:          dc.w 320
48a3 6801 583:          dc.w 360
48a5 9001 584:          dc.w 400
48a7 b801 585:          dc.w 440
48a9 e001 586:          dc.w 480
48ab 0802 587:          dc.w 520
48ad 3002 588:          dc.w 560
48af 5802 589:          dc.w 600
48b1 8002 590:          dc.w 640
48b3 a802 591:          dc.w 680
48b5 d002 592:          dc.w 720
48b7 f802 593:          dc.w 760
48b9 2003 594:          dc.w 800
48bb 4803 595:          dc.w 840
48bd 7003 596:          dc.w 880
48bf 9803 597:          dc.w 920
48c1      598:
48c1      599:
48c1      600: ;
48c1      601: ;-----
48c1      602: ; PRINT BYTE
48c1      603: ;-----
48c1      604: echo
48c1      605: sprint:
48c1      606: fast_output:
48c1 855a 607:          sta $5a
48c3 ad0407 608:          lda sc.redirect
48c6 c902 609:          cmp #print_p
48c8 f01f 610:          beq .go_cio
48ca      611: .leo:
48ca adff02 612:          lda $02ff      ; CONTROL 1
48cd c900 613:          cmp #$00
48cf d0f9 614:          bne .leo
48d1      615:
48d1 a55a 616:          lda $5a
48d3 c920 617:          cmp #32
48d5 9004 618:          bcc .do_look
48d7 c97d 619:          cmp #125
48d9 9013 620:          bcc .do_here

```

```

48db a000 621: .do_look:      ldy  #0
48dd b9e449 622: .lookup:      lda  .char_table,y
48e0 f00c 623:              beq  .do_here
48e2 c55a 624:              cmp  $5a
48e4 f003 625:              beq  .go_cio
48e6 c8 626:          iny
48e7 d0f4 627:              bne  .lookup
48e9      628:
48e9      629:
48e9 a55a 630: .go_cio:      lda  $5a
48eb 4cf449 631:              jmp  putlocal
48ee      632:
48ee      633:
48ee a000 634: .do_here:      ldy  #0
48f0 a55d 635:              lda  $5d
48f2 915e 636:              sta  ($5e),y
48f4 a55a 637:              lda  $5a
48f6 c99b 638:              cmp  #$9b
48f8 f044 639:              beq  .docr
48fa 20c349 640: .notcr:       jsr  .get_adr
48fd a55a 641:              lda  $5a
48ff 297f 642:          and  #$7f
4901 c920 643:              cmp  #32
4903 9007 644:              bcc  .add64
4905 c960 645:              cmp  #96
4907 9009 646:              bcc  .sub32
4909 4c1549 647:              jmp  .asis
490c 18 648: .add64:       clc
490d 6940 649:              adc  #64
490f 4c1549 650:              jmp  .asis
4912 38 651: .sub32:       sec
4913 e920 652:              sbc  #32
4915 245a 653: .asis:        bit  $5a
4917 1002 654:              bpl  .xxlate
4919 0980 655:              ora  #$80
491b a000 656: .xxlate:      ldy  #0
491d 915e 657:              sta  ($5e),y
491f e655 658:              inc  $55
4921 e663 659:              inc  $63
4923 a555 660:              lda  $55
4925 c928 661:              cmp  #40
4927 b019 662:              bcs  .next_row
4929 c8 663:          iny
492a b15e 664:              lda  ($5e),y
492c 855d 665:              sta  $5d
492e 0980 666:              ora  #$80
4930 aef002 667:              ldx  752
4933 d002 668:              bne  .nocurs1
4935 915e 669:              sta  ($5e),y
4937 e65e 670: .nocurs1:     inc  $5e
4939 d002 671:              bne  .nooav
493b e65f 672:              inc  $5e+1
493d 60 673: .nooav:       rts

```

```

493e a900 674: .docr:      lda  #0
4940 8563 675:      sta  $63
4942 a552 676: .next_row:      lda  $52
4944 8555 677:      sta  $55
4946 e654 678:      inc  $54
4948 a454 679:      ldy  $54
494a c018 680:      cpy  #24
494c b013 681:      bcs  .scroll
494e 20c349 682:      jsr  .get_adr
4951 a000 683:      ldy  #0
4953 b15e 684:      lda  ($5e),y
4955 855d 685:      sta  $5d
4957 aef002 686:      ldx  752
495a d004 687:      bne  .nocurs2
495c 0980 688:      ora  #$80
495e 915e 689:      sta  ($5e),y
4960 60 690: .nocurs2:      rts
4961      691:
4961 c654 692: .scroll:      dec  $54
4963 a558 693:      lda  $58
4965 8568 694:      sta  $68
4967 18 695:      clc
4968 6928 696:      adc  #40
496a 855e 697:      sta  $5e
496c a559 698:      lda  $58+1
496e 8569 699:      sta  $68+1
4970 6900 700:      adc  #0
4972 855f 701:      sta  $5e+1
4974 a000 702:      ldy  #0
4976 a203 703:      ldx  #3
4978 b15e 704: .scloop:      lda  ($5e),y
497a 9168 705:      sta  ($68),y
497c c8 706:      iny
497d d0f9 707:      bne  .scloop
497f e65f 708:      inc  $5e+1
4981 e669 709:      inc  $68+1
4983 ca 710:      dex
4984 d0f2 711:      bne  .scloop
4986 b15e 712: .scloop2:      lda  ($5e),y
4988 9168 713:      sta  ($68),y
498a c8 714:      iny
498b c098 715:      cpy  #152
498d 90f7 716:      bcc  .scloop2
498f a558 717:      lda  $58
4991 18 718:      clc
4992 6998 719:      adc  #920&$ff
4994 855e 720:      sta  $5e
4996 a559 721:      lda  $58+1
4998 6903 722:      adc  #920/256
499a 855f 723:      sta  $5e+1
499c a000 724:      ldy  #0
499e 98 725:      tya
499f 915e 726: .clear:      sta  ($5e),y

```

```

49a1 c8 727:      iny
49a2 c028 728:    cpy #40
49a4 90f9 729:    bcc .clear
49a6 aef002 730:    ldx 752
49a9 d006 731:    bne .nocurs3
49ab a980 732:    lda #$80
49ad a455 733:    ldy $55
49af 915e 734:    sta ($5e),y
49b1 a900 735: .nocurs3:    lda #0
49b3 855d 736:    sta $5d
49b5 a55e 737:    lda $5e
49b7 18 738:      clc
49b8 6555 739:    adc $55
49ba 855e 740:    sta $5e
49bc a55f 741:    lda $5e+1
49be 6900 742:    adc #0
49c0 855f 743:    sta $5e+1
49c2 60 744:      rts
49c3 a554 745: .get_adr:    lda $54
49c5 0a 746:      asl
49c6 a8 747:      tay
49c7 a558 748:    lda $58
49c9 18 749:      clc
49ca 799148 750:    adc scr_offset,y
49cd 855e 751:    sta $5e
49cf a559 752:    lda $58+1
49d1 799248 753:    adc scr_offset+1,y
49d4 855f 754:    sta $5e+1
49d6 a55e 755:    lda $5e
49d8 18 756:      clc
49d9 6555 757:    adc $55
49db 855e 758:    sta $5e
49dd a55f 759:    lda $5e+1
49df 6900 760:    adc #0
49e1 855f 761:    sta $5e+1
49e3 60 762:      rts
49e4 763: .char_table:
49e4 1b1c1d 764:    dc.b 27,28,29,30,31,125,126,127
49ec 9c9d9e 765:    dc.b 156,157,158,159,253,254,255
49f4 766:
49f4 767:
49f4 768: putlocal:
49f4 a20b 769:    ldx #11
49f6 8e4203 770:    stx $0342
49f9 a200 771:    ldx #0
49fb 8e4803 772:    stx $0348
49fe 8e4903 773:    stx $0349
4a01 4c56e4 774:    jmp $e456
4a04 775: ;
4a04 776:
4a04 777: ;-----
4a04 778: ; ERROR NUMBER
4a04 779: ;-----

```



```

4a04      780: ;
4a04 00    781: h_cardd: dc.b $00
4a05      782: ;
4a05      783: ; print a hex word value
4a05      784: ; low byte in a reg x is high byte
4a05      785: ;
4a05 8d044a 786: h_card:          sta  h_cardd
4a08 8a    787:          txa
4a09 20134a 788:          jsr  d_error
4a0c ad044a 789:          lda  h_cardd
4a0f 20134a 790:          jsr  d_error
4a12 60    791:          rts
4a13      792: ;
4a13      793: ;
4a13 48    794: d_error:      pha
4a14 4a    795:          lsr
4a15 4a    796:          lsr
4a16 4a    797:          lsr
4a17 4a    798:          lsr
4a18 201e4a 799:          jsr  .a
4a1b 68    800:          pla
4a1c 290f   801:          and  #$0f
4a1e c90a   802: .a:          cmp  #$0a
4a20 b004   803:          bcs  .b
4a22 0930   804:          ora  #$30
4a24 d002   805:          bne  .c
4a26 6936   806: .b:          adc  #$36
4a28 20c148 807: .c:          jsr  echo
4a2b 60    808:          rts
4a2c      809:
4a2c      810:
4a2c      811:
4a2c      812: ;ff ff 00 00 00 00 4c ff ff
4a2c      813: ; ver month day year 0 0
4a2c      814: set_pass:
4a2c ad0d60 815:          lda  irf.c_crc          ;fi
4a2f 8d494a 816:          sta  inst_crc
4a32 ad0e60 817:          lda  irf.c_crc+1
4a35 8d4a4a 818:          sta  inst_crc+1
4a38 ad474a 819:          lda  d_node
4a3b 8d0d60 820:          sta  irf.c_crc
4a3e ad484a 821:          lda  d_node+1
4a41 8d0e60 822:          sta  irf.c_crc+1        ;fi
4a44 60    823:          rts
4a45      824:
4a45      825:
4a45      826: ;
4a45 1f00   827: s_node:          dc.w  l_dos_build
4a47      828:
4a47 1f00   829: d_node:          dc.w  l_dos_build
4a49      830:
4a49 0000   831: inst_crc:        dc.w  0
4a4b      832:

```

```

4a4b 0000 833: block_w:      dc.w 0
4a4d      834:
4a4d      835:
4a4d      836:
4a4d      837: wipe:
4a4d      838:      mea copy_buff,pr_ptr
4a4d a900 ----:      lda #low copy_buff
4a4f 8588 ----:      sta pr_ptr
4a51 a960 ----:      lda #high copy_buff
4a53 8589 ----:      sta pr_ptr + 1
4a55      839:      mea copy_buff,fr0
4a55 a900 ----:      lda #low copy_buff
4a57 85d4 ----:      sta fr0
4a59 a960 ----:      lda #high copy_buff
4a5b 85d5 ----:      sta fr0 + 1
4a5d 20aad9 840:      jsr ifp
4a60 20b6dd 841:      jsr fmove
4a63 ad2f44 842:      lda _totbytes
4a66 85d4 843:      sta fr0
4a68 ad3044 844:      lda _totbytes+1
4a6b 85d5 845:      sta fr0+1
4a6d 20aad9 846:      jsr ifp
4a70 2066da 847:      jsr fadd
4a73 20d2d9 848:      jsr fpi
4a76      849: ;
4a76 a5d4 850:      lda fr0
4a78 8d4b4a 851:      sta block_w
4a7b a5d5 852:      lda fr0+1
4a7d 8d4c4a 853:      sta block_w+1
4a80      854: ;
4a80      855: ;~~~~~debug
4a80 206b48 856:      JSR PRINTSI
4a83 9b2053 857:      dc.b $9B," START $",-1
4a8d a900 858:      LDA # low COPY_BUFF
4a8f a260 859:      LDX # high COPY_BUFF
4a91 20054a 860:      JSR H_CARD
4a94 206b48 861:      JSR PRINTSI
4a97 20454e 862:      dc.b " END $",-1
4a9e ad4b4a 863:      LDA BLOCK_W
4aa1 ae4c4a 864:      LDX BLOCK_W+1
4aa4 20054a 865:      JSR H_CARD
4aa7      866: ;      RTS
4aa7 20174c 867:      JSR CR_LF
4aaa      868: ;
4aaa      869: ;      JSR GET_KEY
4aaa      870: ;~~~~~debug
4aaa      871:
4aaa      872: ;
4aaa a000 873:      ldy #$00
4aac      874: ;
4aac      875: ;      $FFFF
4aac b188 876: wp1:      lda (pr_ptr),y
4aae 20c045 877:      jsr crc_loop

```

```

4ab1 c8 878:      iny
4ab2 c000 879:      cpy  #$00
4ab4 d0f6 880:      bne  wp1
4ab6 e689 881:      inc  pr_ptr+1 ;WP1+2
4ab8 a000 882:      ldy  #$00
4aba a589 883:      lda  pr_ptr+1 ;WP1+2
4abc cd4c4a 884:      cmp  block_w+1
4abf 90eb 885:      bcc  wp1
4ac1 a000 886:      ldy  #$00
4ac3 b188 887: wp2:      lda  (pr_ptr),y
4ac5 20c045 888:      jsr  crc_loop
4ac8 c8 889:      iny
4ac9 cc4b4a 890:      cpy  block_w
4acc 90f5 891:      bcc  wp2
4ace 60 892:      rts
4acf      893: ;
4acf      894: next_run:
4acf      895:
4acf a900 896:      lda  #$00
4ad1 cd494a 897:      cmp  INST_CRC
4ad4 d020 898:      bne  .ok
4ad6 cd4a4a 899:      cmp  INST_CRC+1
4ad9      900:
4ad9 d01b 901:      bne  .ok
4adb 206b48 902:      JSR  printsi
4ade 9b2049 903:      dc.b  $9B," Internal crc not set"
4af5 60 904:      rts
4af6      905:
4af6      906:
4af6      907: .ok:
4af6      908: ;~~~~~debug
4af6 206b48 909:      JSR  printsi
4af9 9b9b20 910:      dc.b  $9B,$9B,"      Crc = $"
4b0d ade747 911:      LDA  CRC_SUM
4b10 aee847 912:      LDX  CRC_SUM+1
4b13 20054a 913:      JSR  H_CARD
4b16 206b48 914:      JSR  printsi
4b19 9b2049 915:      dc.b  $9B," Internal crc = $",-1
4b2c ad494a 916:      LDA  INST_CRC
4b2f ae4a4a 917:      LDX  INST_CRC+1
4b32 20054a 918:      JSR  H_CARD
4b35 20174c 919:      JSR  CR_LF
4b38 20174c 920:      JSR  CR_LF
4b3b      921: ;      RTS
4b3b      922: ;~~~~~debug
4b3b ade747 923:      lda  crc_sum
4b3e cd494a 924:      cmp  inst_crc
4b41 d032 925:      bne  oh_Damn
4b43 ade847 926:      lda  crc_sum+1
4b46 cd4a4a 927:      cmp  inst_crc+1
4b49 d02a 928:      bne  oh_Damn
4b4b 206b48 929:      jsr  printsi
4b4e 9b9b20 930:      dc.b  $9b,$9b," Command PASSED in

```

```

4b74 60 931:          rts
4b75      932: ;
4b75      933: oh_Damn:
4b75 206b48 934:      jsr  printsi
4b78 9b9b20 935:      dc.b  $9b,$9b," Command "
4b83 c6e1e9 936:      dc.b  <+128>,"Failed"
4b89 20696e 937:      dc.b  " internal Crc Test",$9B,$9
4b9e 60 938:          rts
4b9f      939: ;
4b9f      940:
4b9f a904 941: open_in:      lda  #4
4ba1 4cbd4b 942:      jmp  do_open
4ba4      943:
4ba4 a908 944: open_out:     lda  #8
4ba6 4cbd4b 945:      jmp  do_open
4ba9      946:
4ba9 a909 947: open_app:     lda  #9
4bab 4cbd4b 948:      jmp  do_open
4bae      949:
4bae a90c 950: open_updt:    lda  #12
4bb0 4cbd4b 951:      jmp  do_open
4bb3      952:
4bb3 a906 953: open_dir:     lda  #6
4bb5 4cbd4b 954:      jmp  do_open
4bb8      955:
4bb8 a914 956: open_raw:     lda  #20      ;open raw d
4bba 4cbd4b 957:      jmp  do_open
4bbd      958:
4bbd      959: do_open:
4bbd 48 960:      pha          ;open files
4bbe 8a 961:      txa          ;
4bbf 48 962:      pha
4bc0 20ea4b 963:      jsr  close      ;make sure
4bc3 68 964:      pla
4bc4 aa 965:      tax
4bc5 68 966:      pla
4bc6 9d4a03 967:      sta  icaux1,x
4bc9 a903 968:      lda  #ccopen
4bcb 9d4203 969:      sta  iccom,x
4bce a586 970:      lda  fn_ptr
4bd0 9d4403 971:      sta  icba1,x
4bd3 a587 972:      lda  fn_ptr+1
4bd5 9d4503 973:      sta  icbah,x
4bd8 a950 974:      lda  #80
4bda 9d4803 975:      sta  icbll,x
4bdd a900 976:      lda  #0
4bdf 9d4903 977:      sta  icblh,x
4be2 a980 978:      lda  #128      ;for long d
4be4 9d4b03 979:      sta  icaux2,x
4be7 4c56e4 980:      jmp  ciov
4bea      981:
4bea      982:
4bea a90c 983: close:      lda  #cccclose

```

```

4bec 9d4203 984:          sta  iccom,x
4bef 4c56e4 985:          jmp  ciov
4bf2      986:
4bf2      987: ;
4bf2      988: ;
4bf2      989: ; prints: prints string with address in PR_PTR, length by
4bf2      990: ; ~~~~~
4bf2      991:
4bf2      992: echos:
4bf2 a000 993:          ldy  #0
4bf4 b188 994:          lda  (pr_ptr),y      ;ge
4bf6 f01b 995:          beq  .xpri          ;ex
4bf8 8d154c 996:          sta  .maxlth
4bfb c8 997:          iny
4bfc 8c144c 998:          sty  .tempy
4bff      999:
4bff ac144c 1000: .pril:      ldy  .tempy
4c02 b188 1001:          lda  (pr_ptr),y
4c04 c900 1002:          cmp  #$00
4c06 f00b 1003:          beq  .xpri
4c08 20c148 1004:          jsr  echo
4c0b ee144c 1005: .prnext:    inc  .tempy
4c0e ce154c 1006:          dec  .maxlth
4c11 d0ec 1007:          bne  .pril
4c13 60 1008: .xpri:      rts
4c14      1009:
4c14 01 1010: .tempy:      dc.b  1
4c15 01 1011: .maxlth:     dc.b  1
4c16      1012:
4c16 00 1013: pr_mode:     dc.b  0
4c17      1014:
4c17      1015:
4c17      1016:
4c17      1017:
4c17 a99b 1018: cr_lf:      lda  #$9b
4c19 20c148 1019:          jsr  echo
4c1c 60 1020:          rts
4c1d      1021:
4c1d      1022: ;-----
4c1d      1023: ; GET KEY ROUTEEN
4c1d      1024: ;this routeen get's one key and
4c1d      1025: ;return it to the accumulator
4c1d      1026: ;-----
4c1d      1027: ;
4c1d 86b6 1028: get_key:     stx  $b6
4c1f 84b7 1029:          sty  $b7
4c21 20294c 1030:          jsr  g_fff
4c24 a6b6 1031:          ldx  $b6
4c26 a4b7 1032:          ldy  $b7
4c28 60 1033:          rts
4c29 ad25e4 1034: g_fff:      lda  $e420+5
4c2c 48 1035:          pha
4c2d ad24e4 1036:          lda  $e420+4

```

```
4c30 48 1037:          pha
4c31 60 1038:          rts
4c32    1039:
4c32    1040:
4c32    1041:
4c32    1042:
4c32    1043:
4c32    1044: win_end:      ds.b  0
4c32    1045: end_file:
4c32    1046:
6000    1047:          .org  $6000
6000    1048: copy_buff:
6000 0000 1049:          dc.w  0          ; this is a
6002 0000 1050:          dc.w  0          ; File orgs
6004 0000 1051:          dc.w  0          ; length of
6006    1052:
6006 000000 1053: BUFFER3:      dc.b  0,0,0      ; $4c jmp l
6009    1054:
6009 00    1055: irf.c_ver:    dc.b  0          ; Current V
600a 00    1056: irf.c_month:  dc.b  0          ; First Com
600b 00    1057: irf.c_day:    dc.b  0          ; First Com
600c 00    1058: irf.c_year_short: dc.b  0          ; single by
600d    1059:
600d    1060: buffer4:
600d 0000 1061: irf.c_crc:    dc.w  0          ; file CRC
600f    1062: ;
600f    1063: ;
600f    1064:
```

End assembly: no errors