


```

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: ;      to_ascii utility for real.dos v1.0)
0000 10: ;-----
0000 11: ;      to_ascii.com
0000 12: ;
0000 13: ;
0000 14: ;      Code History.  Init design,      4/22/2006
0000 15: ;      Cleaning up labels and formating      6/6/2006
0000 16: ;      Updating the cli portion of the file  6/14/2006
0000 17: ;      added the syntaxing ken wanted!      6/29/2006
0000 18: ;
0000 19: ;Updated the screen Print Utility so it will work with the
0000 20: ; This change will also help "Print P:"      8/1
0000 21: ;
0000 22: ;
0000 23: ;Also added code for Control+1 pause of screen to the high
0000 24: ;handling.      8/24/2012      sjc
0000 25: ;
0000 26: ;
0000 27: ;
0000 28: ;
0000 29: ;      ~~~~~
0000 30: ;-----
0000 31: ; Notes: o This source code MAY NOT be placed for download
0000 32: ;      o "mea" is a macro that loads the address of the
0000 33: ;      into the pointer specified by the second field.
0000 34: ;-----
0000 35: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 36: ;      XASM  st and IBM VERSIONS
0000 37: ;-----
0000 38: ;
0000 39: com_file_name:      .macro
0000 40: ;      dc.b  13,"filename.ext",$9b
0000 41: ;      dc.b  12,"To_ascii.com",$9b
0000 42: ;      .endm
0000 43: ;
0000 44: ;
0000 45: ;~~~~~
0000 46: ; File revision history. Version Number in hex
0000 47: ;
0000 48: file_ver:      equ  $13
0000 49: ;~~~~~
0000 50: ; the Month in dec for the first time this revision
0000 51: ;was compiled
0000 52: ;
0000 53: c_month:      equ  8

```

```
0000 54: ;~~~~~
0000 55: ; the day in dec for the first time this revision
0000 56: ;was compiled
0000 57: c_day:          equ 24
0000 58: ;~~~~~
0000 59: ; the year in dec for the first time this revision
0000 60: ;was compiled
0000 61: c_year:          equ 2012
0000 62: ;~~~~~
0000 63: ; Type of compiler used in program
0000 64: ;
0000 65: ; 0 = unknown Compiler
0000 66: ; 1 = xasm
0000 67: ; 2 = mac_65
0000 68: ; 3 = basic
0000 69: ; 4 = compiled basic xl
0000 70: ; 5 = C65
0000 71: ; 6 = Action
0000 72: ;
0000 73: ; We can add more as time goes on!
0000 74: ;
0000 75: ;
0000 76: xasm:            equ 1
0000 77: mac_65:          equ 2
0000 78: basic:           equ 3
0000 79: basicxl:         equ 4
0000 80: c65:            equ 5
0000 81: action:          equ 6
0000 82:
0000 83: file_compiler:   equ xasm
0000 84: ;~~~~~
0000 85: ; is this relocatable code ?
0000 86: ; anyother value other than 1 or 2 would be unknown
0000 87: r..yes:          equ 1
0000 88: r..no:           equ 2
0000 89: ;
0000 90: ;~~~~~
0000 91: ; Gotta define if it can be relocated
0000 92: l_relocatable:   equ r..no
0000 93: ;
0000 94: ;~~~~~
0000 95: ; Gotta define our Header control Byte
0000 96: ;$ff=bypass intro
0000 97: ;$15 = Bypass Dos check
0000 98: r..crl:          equ $7d
0000 99: r..crlf:         equ $9b
0000 100: r..space:        equ $20
0000 101: r..bypassDOS:    equ $15
0000 102: r..bypass:       equ $ff
0000 103:
0000 104: l_frstscreenbyte: equ r..space
0000 105: ;
0000 106: ;~~~~~
```

```

0000    107: ; The language the output file is in.
0000    108: ;
0000    109: ;
0000    110: ; 0 =   Undefined
0000    111: ; 1 =   English
0000    112: ; 2 =   German
0000    113: ;
0000    114: r..language:      equ   1
0000    115: ;~~~~~
0000    116:
0000    117:
0000    118:
0000    119:          .include      equates
0000    120:          .include      globals
0000    121:          .include      macros
0000    122:
0000    123:          .6502
4000    124:          .org   $4000
4000    125:
4000    126: win_start:
4000    127: header_info:
4000    128:          .include      header
42a7    129:
42a7    130: ;
42a7    131: .real_dos:
42a7    132:
42a7 a210  133:      ldx #disk1
42a9 20774b 134:      jsr close
42ac a220  135:      ldx #disk2
42ae 20774b 136:      jsr close
42b1    137:
42b1 a50a  138:      LDA comtab
42b3 18    139:      CLC
42b4 6903  140:      ADC # low ZCRNAME      ;3
42b6 8dc142 141:      STA .a+1
42b9 a50b  142:      LDA comtab+1
42bb 6900  143:      ADC # high ZCRNAME     ;0
42bd 8dc242 144:      STA .a+2
42c0 200000 145: .a    JSR $0000
42c3    146:
42c3 d035  147:      bne got_something      ; nothing there
42c5    148:
42c5 209349 149:      JSR printsi
42c8 9b9b  150:      dc.b $9B,$9B
42ca 205379 151:      dc.b " Syntax for use:",$9b
42db 20746f 152:      dc.b " to_ascii Dn:>dir>fname.ext",$9b,$9b,-1
42f9 60    153:      RTS
42fa    154:
42fa    155:
42fa    156: got_something:
42fa    157:
42fa a50a  158:      LDA comtab
42fc 18    159:      CLC

```

```

42fd 6921 160:    ADC # low COMFNAM      ;3
42ff 8582 161:    STA s_ptr
4301 a50b 162:    LDA comtab+1
4303 6900 163:    ADC # high COMFNAM      ;0
4305 8583 164:    STA s_ptr+1
4307      165:    mea parm_1,d_ptr
4307 a941 ----:    lda   #low parm_1
4309 8584 ----:    sta   d_ptr
430b a945 ----:    lda   #high parm_1
430d 8585 ----:    sta   d_ptr + 1
430f 209f50 166:    jsr tostr
4312      167:
4312      168: ;    jsr cr_lf
4312      169: ;    mea parm_1,pr_ptr
4312      170: ;    jsr echos
4312      171: ;    jsr echos
4312      172:
4312      173:
4312      174: ;
4312      175: GO_ON
4312 a900 176:    LDA #$00
4314      177: ;
4314 209349 178:    JSR printsi
4317 202020 179:    dc.b "  Atascii --> Ascii Converter",$9
433a      180:
433a      181:
433a      182: ;
433a a220 183:    LDX #$20
433c 20774b 184:    JSR CLOSE
433f a240 185:    LDX #$40
4341 20774b 186:    JSR CLOSE
4344 209349 187:    JSR printsi
4347 9b9b20 188:    dc.b $9B,$9B," Processing file! ",-1
435d      189: ;
435d 207a44 190:    JSR BUILD_NME
4360      191: ;
4360 20bb43 192:    JSR PROCESS
4363      193: ;
4363 a220 194:    LDX #$20
4365 20774b 195:    JSR CLOSE
4368 a240 196:    LDX #$40
436a 20774b 197:    JSR CLOSE
436d      198:    MEA IN+1,FN_PTR
436d a92d ----:    lda   #low IN+1
436f 8586 ----:    sta   FN_PTR
4371 a94e ----:    lda   #high IN+1
4373 8587 ----:    sta   FN_PTR + 1
4375 a220 199:    LDX #$20
4377 20cf47 200:    JSR ERASE
437a a220 201:    LDX #$20
437c 20774b 202:    JSR CLOSE
437f      203:    MEA SPACE,S_PTR
437f a95d ----:    lda   #low SPACE

```

```

4381 8582 ----:      sta  S_PTR
4383 a94f ----:      lda  #high SPACE
4385 8583 ----:      sta  S_PTR + 1
4387 204:      MEA  OUT,D_PTR
4387 a990 ----:      lda  #low OUT
4389 8584 ----:      sta  D_PTR
438b a94e ----:      lda  #high OUT
438d 8585 ----:      sta  D_PTR + 1
438f 20764f 205:     JSR CONCAT
4392 206:      MEA  FILENAME,S_PTR
4392 a96b ----:      lda  #low FILENAME
4394 8582 ----:      sta  S_PTR
4396 a94d ----:      lda  #high FILENAME
4398 8583 ----:      sta  S_PTR + 1
439a 207:      MEA  OUT,D_PTR
439a a990 ----:      lda  #low OUT
439c 8584 ----:      sta  D_PTR
439e a94e ----:      lda  #high OUT
43a0 8585 ----:      sta  D_PTR + 1
43a2 20764f 208:     JSR CONCAT
43a5 209:      MEA  OUT+1,FN_PTR
43a5 a991 ----:      lda  #low OUT+1
43a7 8586 ----:      sta  FN_PTR
43a9 a94e ----:      lda  #high OUT+1
43ab 8587 ----:      sta  FN_PTR + 1
43ad a220 210:      LDX #$20
43af 20c947 211:      JSR RENAME
43b2 a220 212:      LDX #$20
43b4 20774b 213:      JSR CLOSE
43b7 214:      ;
43b7 215:      ;
43b7 208d49 216:      JSR CR_LF
43ba 60 217:      RTS
43bb 218:      ;
43bb 219:      ;
43bb 220:      ;
43bb 221:      PROCESS
43bb 222:      MEA  IN+1,FN_PTR
43bb a92d ----:      lda  #low IN+1
43bd 8586 ----:      sta  FN_PTR
43bf a94e ----:      lda  #high IN+1
43c1 8587 ----:      sta  FN_PTR + 1
43c3 a220 223:      LDX #$20
43c5 202c4b 224:      JSR OPEN_IN
43c8 3013 225:      BMI OH_DAMN
43ca 226:      MEA  OUT+1,FN_PTR
43ca a991 ----:      lda  #low OUT+1
43cc 8586 ----:      sta  FN_PTR
43ce a94e ----:      lda  #high OUT+1
43d0 8587 ----:      sta  FN_PTR + 1
43d2 a240 227:      LDX #$40
43d4 20314b 228:      JSR OPEN_OUT
43d7 3004 229:      BMI OH_DAMN

```

```
43d9 200144 230:    JSR READ
43dc 60    231:    RTS
43dd      232:    ;
43dd      233:    OH_DAMN
43dd 8c3246 234:    STY RC
43e0 209349 235:    JSR printsi
43e3 9b9b20 236:    dc.b $9B,$9B," Error Opening file",$9B,$9B,-1
43fb a901   237:    LDA #$01
43fd 8d3146 238:    STA ABORTED
4400 60    239:    RTS
4401      240:    ;
4401      241:    ;
4401      242:    READ
4401 a220   243:    LDX #$20
4403      244:    MEA TEMP1,IN_PTR
4403 a980 ----:    lda  #low TEMP1
4405 8580 ----:    sta  IN_PTR
4407 a94b ----:    lda  #high TEMP1
4409 8581 ----:    sta  IN_PTR + 1
440b a220   245:    LDX #$20
440d 20b74f 246:    JSR INPUTSD
4410      247:    ;
4410 300c   248:    BMI .ER
4412      249:
4412 202045 250:    jsr  paused
4415      251:
4415 203544 252:    JSR SPLICE
4418 206144 253:    JSR WRITE
441b      254:    ;
441b 4c0144 255:    JMP READ
441e      256:    ;
441e      257:    .ER
441e 209349 258:    JSR printsi
4421 9b9b20 259:    dc.b $9B,$9B," all Done",$9B,$FF
442e      260:    ;
442e 203544 261:    JSR SPLICE
4431 206144 262:    JSR WRITE
4434 60    263:    RTS
4435      264:    ;
4435      265:    SPLICE
4435 ac804b 266:    LDY TEMP1
4438 c000   267:    CPY #$00
443a f011   268:    BEQ .SPL
443c b9804b 269:    LDA TEMP1,Y
443f c920   270:    CMP #$20
4441 d00a   271:    BNE .SPL
4443 ac804b 272:    LDY TEMP1
4446 88     273:    DEY
4447 8c804b 274:    STY TEMP1
444a 4c3544 275:    JMP SPLICE
444d      276:    .SPL
444d      277:    MEA TEMP1,D_PTR
444d a980 ----:    lda  #low TEMP1
```

```

444f 8584 ----:      sta  D_PTR
4451 a94b ----:      lda  #high TEMP1
4453 8585 ----:      sta  D_PTR + 1
4455 278:      MEA  LF,S_PTR
4455 a94e ----:      lda  #low LF
4457 8582 ----:      sta  S_PTR
4459 a94f ----:      lda  #high LF
445b 8583 ----:      sta  S_PTR + 1
445d 20764f 279:     JSR CONCAT
4460 60 280:      RTS
4461 281: ;
4461 282: WRITE
4461 283:      MEA  TEMP1+1,PR_PTR
4461 a981 ----:      lda  #low TEMP1+1
4463 8588 ----:      sta  PR_PTR
4465 a94b ----:      lda  #high TEMP1+1
4467 8589 ----:      sta  PR_PTR + 1
4469 a240 284:      LDX #$40
446b ad804b 285:      LDA TEMP1
446e 8d9348 286:      STA LNPTR
4471 a900 287:      LDA #$00
4473 8d9448 288:      STA LNPTR+1
4476 201d49 289:      JSR BPUT
4479 290: ; JSR PRINTSD
4479 60 291:      RTS
447a 292: ;
447a 293: ;
447a 294: BUILD_NME
447a 295:      MEA  PARM_1,S_PTR
447a a941 ----:      lda  #low PARM_1
447c 8582 ----:      sta  S_PTR
447e a945 ----:      lda  #high PARM_1
4480 8583 ----:      sta  S_PTR + 1
4482 296:      MEA  IN,D_PTR
4482 a92c ----:      lda  #low IN
4484 8584 ----:      sta  D_PTR
4486 a94e ----:      lda  #high IN
4488 8585 ----:      sta  D_PTR + 1
448a 205f4f 297:     JSR MOVE_STR
448d 298:      MEA  PARM_1,S_PTR
448d a941 ----:      lda  #low PARM_1
448f 8582 ----:      sta  S_PTR
4491 a945 ----:      lda  #high PARM_1
4493 8583 ----:      sta  S_PTR + 1
4495 299:      MEA  DIR_MASK,D_PTR
4495 a980 ----:      lda  #low DIR_MASK
4497 8584 ----:      sta  D_PTR
4499 a94d ----:      lda  #high DIR_MASK
449b 8585 ----:      sta  D_PTR + 1
449d 205f4f 300:     JSR MOVE_STR
44a0 20e644 301:     JSR STRIP_PATH
44a3 302: ;
44a3 303:      MEA  PARM_1,S_PTR

```



```

44a3 a941 ----:      lda  #low PARM_1
44a5 8582 ----:      sta  S_PTR
44a7 a945 ----:      lda  #high PARM_1
44a9 8583 ----:      sta  S_PTR + 1
44ab 304:      MEA  OUT,D_PTR
44ab a990 ----:      lda  #low OUT
44ad 8584 ----:      sta  D_PTR
44af a94e ----:      lda  #high OUT
44b1 8585 ----:      sta  D_PTR + 1
44b3 205f4f 305:     JSR MOVE_STR
44b6 306: ;
44b6 307:      MEA  OUT,S_PTR
44b6 a990 ----:      lda  #low OUT
44b8 8582 ----:      sta  S_PTR
44ba a94e ----:      lda  #high OUT
44bc 8583 ----:      sta  S_PTR + 1
44be 308:      MEA  DOT,D_PTR
44be a951 ----:      lda  #low DOT
44c0 8584 ----:      sta  D_PTR
44c2 a94f ----:      lda  #high DOT
44c4 8585 ----:      sta  D_PTR + 1
44c6 201b50 309:     JSR SUBSTR
44c9 c900 310:      CMP #$00
44cb f005 311:      BEQ .BN
44cd a8 312:      TAY
44ce 88 313:      DEY
44cf 8c904e 314:     STY OUT
44d2 315: .BN
44d2 316:      MEA  EXT,S_PTR
44d2 a953 ----:      lda  #low EXT
44d4 8582 ----:      sta  S_PTR
44d6 a94f ----:      lda  #high EXT
44d8 8583 ----:      sta  S_PTR + 1
44da 317:      MEA  OUT,D_PTR
44da a990 ----:      lda  #low OUT
44dc 8584 ----:      sta  D_PTR
44de a94e ----:      lda  #high OUT
44e0 8585 ----:      sta  D_PTR + 1
44e2 20764f 318:     JSR CONCAT
44e5 319: ;
44e5 320: ; JSR CR_LF
44e5 321: ; MEA IN,PR_PTR
44e5 322: ; JSR ECHOS
44e5 323: ;
44e5 324: ; JSR CR_LF
44e5 325: ; MEA OUT,PR_PTR
44e5 326: ; JSR ECHOS
44e5 327: ;
44e5 328: ; JSR CR_LF
44e5 60 329:      RTS
44e6 330: ;
44e6 331: ; dc.b 0,"dx:filename"
44e6 332: STRIP_PATH

```

```
44e6 ac804d 333:    LDY DIR_MASK
44e9      334: PATH_W
44e9 b9804d 335:    LDA DIR_MASK,Y
44ec c93a 336:    CMP #':'
44ee f014 337:    BEQ W_D
44f0 c93e 338:    CMP #'>'
44f2 f010 339:    BEQ W_D
44f4 c93c 340:    CMP #'<'
44f6 f00c 341:    BEQ W_D
44f8 c000 342:    CPY #$00
44fa f008 343:    BEQ W_D
44fc c050 344:    CPY #80
44fe b004 345:    BCS W_D
4500 88 346:    DEY
4501 4ce944 347:    JMP PATH_W
4504      348: ;
4504      349: W_D
4504 8c804d 350:    STY DIR_MASK
4507      351: W_N
4507      352: ;
4507      353: ;
4507 a200 354:    LDX #$00
4509 ac804d 355:    LDY DIR_MASK
450c      356: W_ZZ
450c 8e6b4d 357:    STX FILENAME
450f b9804d 358:    LDA DIR_MASK,Y
4512 9d6b4d 359:    STA FILENAME,X
4515 cc2c4e 360:    CPY IN
4518 b005 361:    BCS ALLW
451a c8 362:    INY
451b e8 363:    INX
451c 4c0c45 364:    JMP W_ZZ
451f      365: ALLW
451f      366: ; JSR CR_LF
451f      367: ; MEA DIR_MASK,PR_PTR
451f      368: ; JSR PRINTS
451f      369: ; JSR CR_LF
451f      370: ; MEA FILENAME,PR_PTR
451f      371: ; JSR PRINTS
451f      372: ; JSR CR_LF
451f 60 373:    RTS
4520      374: ;
4520      375: paused:
4520 209349 376:    jsr printsi
4523 7e 377:    dc.b 126 ;''
4524 ff 378:    dc.b -1
4525 ac3645 379:    ldy .hold
4528 c8 380:    iny
4529 8c3645 381:    sty .hold
452c c004 382:    cpy #$04
452e 9005 383:    bcc .tammy
4530 a000 384:    ldy #$00
4532 8c3645 385:    sty .hold
```

```

4535      386: .tammy:
4535 60    387:          rts
4536      388:
4536 00    389: .hold:      dc.b 0
4537 7c0612 390: .tab1:      dc.b $7c,$06,$12,$07,"*"
453c 7c2f2d 391: .tab:      dc.b "|/*"
4541      392:
4541      393:
4541      394: AS_IS          equ 0
4541      395: parm_1:          ds.b 40          ;passed by
4569      396: parm_2:          ds.b 40          ;passed by DOSshel
4591      397: in_str:          ds.b 80          ;
45e1      398: instr:          ds.b 80          ;
4631      399: ;
4631 00    400: aborted:        dc.b 0
4632      401:
4632      402: rc:             ds.b 1           ;return code from
4633      403:
4633 50    404: u_vw:           dc.b 80          ;video width
4634      405: u_flags:        ds.b 2           ; 16 individual fl
4636      406:
4636      407: hide:           ds.b 1           ;PRIVATE 1 means echo '*'
4637 00    408: trans:          dc.b 0
4638      409: i_bufoff:        ds.b 1           ;current positio
4639      410: word:           ds.b 30          ;damn backward com
4657      411: u_handle:        ds.b 21          ;handle
466c      412: u_name:          ds.b 21          ;real name
4681      413: cur_date:        ds.b 15          ; Mon 13-May-87 +
4690      414: cur_time:        ds.b 15          ; 11:00:00am +
469f      415: cur_month:       ds.b 1           ;1-12
46a0      416: cur_day:         ds.b 1           ;1-31
46a1      417: cur_year:        ds.b 1           ;1-99
46a2      418: cur_hour:         ds.b 1           ;0-23
46a3      419: cur_minute:      ds.b 1           ;0-59
46a4      420: cur_second:      ds.b 1           ;0-59
46a5      421: cur_weekday:     ds.b 1           ;0=sunday, 1=monda
46a6 00    422: in_clock:        dc.b 0           ;set by the Z: handler functions
46a7      423: tdbuf:          ds.b 30          ;time/date buffer area
46c5      424: s_today:         ds.b 14          ;today's date
46d3      425: s_calltoday:     ds.w 1           ;total calls today
46d5      426:
46d5      427: ; Steve's siov source code file
46d5      428:
46d5      429: ; MY CIOV MACROS
46d5      430: ; THE CHANEL NUMBER #$20 AND SO ON
46d5      431: ; LOADED INTO THE X REGISTER FILE
46d5      432: ; PARM'S ARE LOADED LOW BYTE INTO
46d5      433: ; FN_PTR HIGH BYTE FN_PTR+1 THEN A JSR
46d5      434: ; TO THE LABEL YOU WISH TO OPEN BY
46d5      435: ;
46d5      436: OPENIN
46d5 a904 437: LDA #$04
46d7 9d4a03 438: STA $034A,X

```

```
46da 206c47 439:    JSR OPEN_WORK
46dd 60 440:    RTS
46de 441: ;
46de 442: ;
46de 443: OPENOUT
46de a908 444:    LDA #$08
46e0 9d4a03 445:    STA $034A,X
46e3 206c47 446:    JSR OPEN_WORK
46e6 60 447:    RTS
46e7 448: OPENAPP
46e7 a909 449:    LDA #$09
46e9 9d4a03 450:    STA $034A,X
46ec 206c47 451:    JSR OPEN_WORK
46ef 60 452:    RTS
46f0 453: OPENUPDT
46f0 a90c 454:    LDA #$0C
46f2 9d4a03 455:    STA $034A,X
46f5 206c47 456:    JSR OPEN_WORK
46f8 60 457:    RTS
46f9 458: OPENDIR
46f9 a906 459:    LDA #$06
46fb 9d4a03 460:    STA $034A,X
46fe 206c47 461:    JSR OPEN_WORK
4701 60 462:    RTS
4702 463: GET
4702 a905 464:    LDA #$05
4704 9d4203 465:    STA $0342,X
4707 a588 466:    LDA PR_PTR
4709 9d4403 467:    STA $0344,X
470c a589 468:    LDA PR_PTR+1
470e 9d4503 469:    STA $0345,X
4711 ad9348 470:    LDA LNPTR
4714 9d4803 471:    STA $0348,X
4717 ad9448 472:    LDA LNPTR+1
471a 9d4903 473:    STA $0349,X
471d a903 474:    LDA #$03
471f 9d4a03 475:    STA $034A,X
4722 2056e4 476:    JSR $E456
4725 60 477:    RTS
4726 478: ;
4726 479: BOOT ; XIO 45
4726 204247 480:    JSR XIO_CLEAR
4729 a926 481:    LDA #$26
472b 206c47 482:    JSR OPEN_WORK
472e 60 483:    RTS
472f 484: ;
472f 485: ;
472f 00 486: STATUS1 dc.b $00
4730 487: STATUS
4730 8d2f47 488:    STA STATUS1
4733 204247 489:    JSR XIO_CLEAR
4736 ad2f47 490:    LDA STATUS1
4739 9d4a03 491:    STA $034A,X
```

```
473c a90d 492: LDA #$0D
473e 20d948 493: JSR XIO_WORK
4741 60 494: RTS
4742 495: ;
4742 496: XIO_CLEAR
4742 a900 497: LDA #$00
4744 8588 498: STA PR_PTR
4746 8589 499: STA PR_PTR+1
4748 8d9348 500: STA LNPTR
474b 8d9448 501: STA LNPTR+1
474e 60 502: RTS
474f 503: ;
474f 504: OPENRAW
474f a903 505: LDA #$03
4751 9d4203 506: STA $0342,X
4754 a906 507: LDA #$06
4756 9d4a03 508: STA $034A,X
4759 a980 509: LDA #128
475b 9d4b03 510: STA $034B,X
475e a586 511: LDA FN_PTR
4760 9d4403 512: STA $0344,X
4763 a587 513: LDA FN_PTR+1
4765 9d4503 514: STA $0345,X
4768 2056e4 515: JSR $E456
476b 60 516: RTS
476c 517: ;
476c 518: ;
476c 519: OPEN_WORK
476c a903 520: LDA #$03
476e 9d4203 521: STA $0342,X
4771 bd4a03 522: LDA $034A,X
4774 8d8b47 523: STA AUX
4777 bd4b03 524: LDA $034B,X
477a 8d8c47 525: STA AUX+1
477d a586 526: LDA FN_PTR
477f 9d4403 527: STA $0344,X
4782 a587 528: LDA FN_PTR+1
4784 9d4503 529: STA $0345,X
4787 204249 530: JSR C_IOV
478a 60 531: RTS
478b 0000 532: AUX dc.b 0,0
478d 533: ;-----
478d 534: ;you must jsr to this vector for point, note, file lenght
478d 535: ;
478d 44783a 536: PNFDATA dc.b "Dx:",$00
4791 0000 537: PNFDATA1 dc.b $00,$00
4793 538: PNF_WORK
4793 8e9147 539: STX PNFDATA1
4796 a586 540: LDA FN_PTR
4798 8da647 541: STA PNF_1+1
479b a587 542: LDA FN_PTR+1
479d 8da747 543: STA PNF_1+2
47a0 8c9247 544: STY PNFDATA1+1
```

```
47a3 a001 545: LDY #$01
47a5 b9ffff 546: PNF_1 LDA $FFFF,Y
47a8 8d8e47 547: STA PNFDATA+1
47ab ac9247 548: LDY PNFDATA1+1
47ae ae9147 549: LDX PNFDATA1
47b1 a900 550: LDA #$00
47b3 9d4a03 551: STA $034A,X
47b6 a900 552: LDA #$00
47b8 9d4b03 553: STA $034B,X
47bb a98d 554: lda # low PNFDATA
47bd 9d4403 555: STA $0344,X
47c0 a947 556: lda # high PNFDATA
47c2 9d4503 557: STA $0345,X
47c5 204249 558: JSR C_IOV
47c8 60 559: RTS
47c9 560: ;
47c9 561: RENAME ; XIO #32 RENAME FILE
47c9 a920 562: LDA #$20
47cb 20d948 563: JSR XIO_WORK
47ce 60 564: RTS
47cf 565: ;
47cf 566: ERASE ; XIO #33 ERASE FILE'S
47cf 204247 567: JSR XIO_CLEAR
47d2 a921 568: LDA #$21
47d4 20d948 569: JSR XIO_WORK
47d7 60 570: RTS
47d8 571: ;
47d8 572: LOCK ; XIO #34 LOCK DISKETTE
47d8 204247 573: JSR XIO_CLEAR
47db a922 574: LDA #$22
47dd 20d948 575: JSR XIO_WORK
47e0 60 576: RTS
47e1 577: ;
47e1 578: PROTECT ; XIO #35 PROTECT FILE'S
47e1 204247 579: JSR XIO_CLEAR
47e4 a923 580: LDA #$23
47e6 20d948 581: JSR XIO_WORK
47e9 60 582: RTS
47ea 583: UNPROTECT ; XIO #36 UNPROTECT FILE'S
47ea 204247 584: JSR XIO_CLEAR
47ed a924 585: LDA #$24
47ef 20d948 586: JSR XIO_WORK
47f2 60 587: RTS
47f3 588: ;
47f3 589: POINT ; XIO 37 POINT
47f3 a925 590: LDA #$25
47f5 8e8b48 591: STX XIO_39_HOLD
47f8 9d4203 592: STA $0342,X
47fb ad1648 593: LDA OFFSET
47fe 9d4c03 594: STA $034C,X
4801 ad1448 595: LDA f.sector
4804 9d4d03 596: STA $034D,X
4807 ad1548 597: LDA f.sector+1
```

```
480a 9d4e03 598: STA $034E,X
480d 209347 599: JSR PNF_WORK
4810 ae8b48 600: LDX XIO_39_HOLD
4813 60 601: RTS
4814 0000 602: f.sector dc.w 0
4816 00 603: OFFSET dc.b 0
4817 604: NOTE ; XIO_38 NOTE
4817 8e8b48 605: STX XIO_39_HOLD
481a a926 606: LDA #$26
481c 9d4203 607: STA $0342,X
481f 209347 608: JSR PNF_WORK
4822 ae8b48 609: LDX XIO_39_HOLD
4825 bd4c03 610: LDA $034C,X
4828 8d1648 611: STA OFFSET
482b bd4d03 612: LDA $034D,X
482e 8d1448 613: STA f.sector
4831 bd4e03 614: LDA $034E,X
4834 8d1548 615: STA f.sector+1
4837 60 616: XIO_38E RTS
4838 000000 617: NPSAVE1 dc.b 0,0,0
483b 618: NPSAVE
483b ad1648 619: LDA OFFSET
483e 8d3848 620: STA NPSAVE1
4841 ad1448 621: LDA f.sector
4844 8d3948 622: STA NPSAVE1+1
4847 ad1548 623: LDA f.sector+1
484a 8d3a48 624: STA NPSAVE1+2
484d 60 625: RTS
484e 626: NPLOAD
484e ad3848 627: LDA NPSAVE1
4851 8d1648 628: STA OFFSET
4854 ad3948 629: LDA NPSAVE1+1
4857 8d1448 630: STA f.sector
485a ad3a48 631: LDA NPSAVE1+2
485d 8d1548 632: STA f.sector+1
4860 60 633: RTS
4861 634: ;
4861 635: F_LENGTH ; XIO#39 FILE LENGTH SPARTA ONLY
4861 8e8b48 636: STX XIO_39_HOLD
4864 a927 637: LDA #$27
4866 9d4203 638: STA $0342,X
4869 209347 639: JSR PNF_WORK
486c ae8b48 640: LDX XIO_39_HOLD
486f bd4c03 641: LDA $034C,X
4872 8d8d48 642: STA FNLNG
4875 8d9048 643: STA F_LNG
4878 bd4d03 644: LDA $034D,X
487b 8d8e48 645: STA FNLNG+1
487e 8d9148 646: STA F_LNG+1
4881 bd4e03 647: LDA $034E,X
4884 8d8f48 648: STA FNLNG+2
4887 8d9248 649: STA F_LNG+2
488a 60 650: RTS
```

```
488b 0000 651: XIO_39_HOLD dc.b $00,$00
488d 000000 652: FNLNG dc.b 0,0,0
4890 000000 653: F_LNG dc.b 0,0,0
4893 654: ;FN_PTR dc.w 0
4893 0000 655: LNPTR dc.w 0
4895 656: ;
4895 657: BLOAD ; AND RUN _XIO_40_
4895 a928 658: LDA #$28
4897 20d948 659: JSR XIO_WORK
489a 60 660: RTS
489b 661: ;
489b 662: BSAVE ; AND APPEND _XIO_41_
489b a929 663: LDA #$29
489d 20d948 664: JSR XIO_WORK
48a0 60 665: RTS
48a1 666: ;
48a1 667: CREDIR ; XIO_42
48a1 204247 668: JSR XIO_CLEAR
48a4 a92a 669: LDA #$2A
48a6 20d948 670: JSR XIO_WORK
48a9 60 671: RTS
48aa 672: ;
48aa 673: DELDIR ; XIO_43
48aa 204247 674: JSR XIO_CLEAR
48ad a92b 675: LDA #$2B
48af 20d948 676: JSR XIO_WORK
48b2 60 677: RTS
48b3 678: ;
48b3 679: CWD ; [C]hange [W]orking [D]ir
48b3 680: ; XIO_44
48b3 a92c 681: LDA #$2C
48b5 20d948 682: JSR XIO_WORK
48b8 60 683: RTS
48b9 684: ;
48b9 685: CHKDSK ; XIO 47 CHECK DISK
48b9 a588 686: LDA PR_PTR
48bb 9d4803 687: STA $0348,X
48be a589 688: LDA PR_PTR+1
48c0 9d4903 689: STA $0349,X
48c3 690: ;
48c3 a92f 691: LDA #$2F
48c5 20d948 692: JSR XIO_WORK
48c8 60 693: RTS
48c9 694: ;
48c9 695: GET_DIR_PATH ; ?DIR XIO 48
48c9 a588 696: LDA PR_PTR
48cb 9d4803 697: STA $0348,X
48ce a589 698: LDA PR_PTR+1
48d0 9d4903 699: STA $0349,X
48d3 700: ;
48d3 a930 701: LDA #$30
48d5 20d948 702: JSR XIO_WORK
48d8 60 703: RTS
```



```
48d9      704: ;
48d9      705: ;
48d9      706: ;
48d9      707: XIO_WORK
48d9 9d4203 708:   STA $0342,X
48dc a904  709:   LDA #$04
48de 9d4a03 710:   STA $034A,X
48e1 a97f  711:   LDA #$7F
48e3 9d4b03 712:   STA $034B,X
48e6 a586  713:   LDA FN_PTR
48e8 9d4403 714:   STA $0344,X
48eb a587  715:   LDA FN_PTR+1
48ed 9d4503 716:   STA $0345,X
48f0 204249 717:   JSR C_IOV
48f3 60    718:   RTS
48f4      719: ;
48f4      720: ;-----;
48f4      721: ; LDA PR_PTR ; BUFFER
48f4      722: ; STA #<BUFFER+1
48f4      723: ;
48f4      724: BGET
48f4 a907  725:   LDA #$07
48f6 9d4203 726:   STA $0342,X
48f9 a588  727:   LDA PR_PTR
48fb 9d4403 728:   STA $0344,X
48fe a589  729:   LDA PR_PTR+1
4900 9d4503 730:   STA $0345,X
4903 ad9348 731:   LDA LNPTR
4906 9d4803 732:   STA $0348,X
4909 ad9448 733:   LDA LNPTR+1
490c 9d4903 734:   STA $0349,X
490f a903  735:   LDA #$03
4911 9d4a03 736:   STA $034A,X
4914 a900  737:   LDA #$00
4916 9d4b03 738:   STA $034B,X
4919 204249 739:   JSR C_IOV
491c 60    740:   RTS
491d      741: ;-----
491d      742: ; PR_PTR=BUFFER FN_PTR=LEINGTH OF BUFFER
491d      743: ;-----
491d      744: BPUT
491d a90b  745:   LDA #$0B
491f 9d4203 746:   STA $0342,X
4922 a588  747:   LDA PR_PTR
4924 9d4403 748:   STA $0344,X
4927 a589  749:   LDA PR_PTR+1
4929 9d4503 750:   STA $0345,X
492c ad9348 751:   LDA LNPTR
492f 9d4803 752:   STA $0348,X
4932 ad9448 753:   LDA LNPTR+1
4935 9d4903 754:   STA $0349,X
4938 a903  755:   LDA #$03
493a 9d4a03 756:   STA $034A,X
```

```

493d a900 757:    LDA #$00
493f 9d4b03 758:    STA $034B,X
4942      759: C_IOV
4942 2056e4 760:    JSR $E456
4945 ad0303 761:    LDA $0303
4948 a8 762:    TAY
4949 c081 763:    CPY #$81
494b 9003 764:    BCC C_IOV_EXIT
494d 205249 765:    JSR DRIVE_ERROR
4950 60 766: C_IOV_EXIT RTS
4951 60 767:    RTS
4952      768: DRIVE_ERROR
4952 48 769:    PHA
4953 209349 770:    JSR printsi
4956 9b2043 771:    dc.b $9B," Ciov Error # ",$FF
4966 68 772:    PLA
4967 206e49 773:    JSR PR_BYTE
496a 208d49 774:    JSR CR_LF
496d 60 775:    RTS
496e      776: ;
496e      777: ;    print the byte currently in Accum.
496e      778: ;    ~~~~~
496e      779: ;
496e      780: pr_byte:
496e a200 781:    ldx #$00
4970      782: pr_card:
4970 85d4 783:    sta $d4
4972 86d5 784:    stx $d4+1
4974 a900 785:    lda #$00
4976 85f2 786:    sta $f2
4978 20aad9 787:    jsr $d9aa
497b 20e6d8 788:    jsr $d8e6
497e a000 789:    ldy #$00
4980      790: .aprbloop:
4980 b1f3 791:    lda ($f3),y
4982 08 792:    php
4983 297f 793:    and #$7f
4985 c8 794:    iny
4986 201c4b 795:    jsr putlocal
4989 28 796:    plp
498a 10f4 797:    bpl .aprbloop
498c 60 798:    rts
498d      799: ;
498d      800:
498d      801:
498d 209349 802: cr_lf:    jsr printsi
4990 9bff 803:    dc.b $9b,-1
4992 60 804:    rts
4993      805:
4993      806:
4993      807: ;Supporting stuff
4993      808: ;
4993      809: ;-----

```

```

4993      810: ;prints Routeen macro
4993      811: ;this is a slow print to the screen
4993      812: ;usage jsr  prints
4993      813: ;  .byte $9b," string to be printed",$ff
4993      814: ;-----
4993      815: ;
4993      816: prints:
4993 68    817:          pla
4994 8da449 818:          sta  .pstr+1
4997 68    819:          pla
4998 8da549 820:          sta  .pstr+2
499b eea449 821: .prsl:    inc  .pstr+1
499e d003   822:          bne  .pstr
49a0 eea549 823:          inc  .pstr+2
49a3 adffff 824: .pstr:    lda  $ffff
49a6 c9ff   825:          cmp  #$ff
49a8 f006   826:          beq  .estri
49aa 20e949 827:          jsr  fast_output
49ad 4c9b49 828:          jmp  .prsl
49b0 ada549 829: .estri:   lda  .pstr+2
49b3 48    830:          pha
49b4 ada449 831:          lda  .pstr+1
49b7 48    832:          pha
49b8 60    833:          rts
49b9      834: ;
49b9      835: ;
49b9      836: ;
49b9      837: ;-----
49b9      838: ;actual screen handler
49b9      839: ;screen offset definitions
49b9      840: ;-----
49b9      841: scr_offset:
49b9 0000   842:          dc.w  0
49bb 2800   843:          dc.w  40
49bd 5000   844:          dc.w  80
49bf 7800   845:          dc.w 120
49c1 a000   846:          dc.w 160
49c3 c800   847:          dc.w 200
49c5 f000   848:          dc.w 240
49c7 1801   849:          dc.w 280
49c9 4001   850:          dc.w 320
49cb 6801   851:          dc.w 360
49cd 9001   852:          dc.w 400
49cf b801   853:          dc.w 440
49d1 e001   854:          dc.w 480
49d3 0802   855:          dc.w 520
49d5 3002   856:          dc.w 560
49d7 5802   857:          dc.w 600
49d9 8002   858:          dc.w 640
49db a802   859:          dc.w 680
49dd d002   860:          dc.w 720
49df f802   861:          dc.w 760
49e1 2003   862:          dc.w 800

```

```

49e3 4803 863:          dc.w 840
49e5 7003 864:          dc.w 880
49e7 9803 865:          dc.w 920
49e9      866:
49e9      867: ;
49e9      868: fast_output:
49e9 855a 869:          sta $5a
49eb ad0407 870:         lda sc.redirect
49ee c902 871:          cmp #print_p
49f0 f01f 872:         beq .go_cio
49f2      873: .leo:
49f2 adff02 874:         lda $02ff      ; CONTROL 1
49f5 c900 875:          cmp #$00
49f7 d0f9 876:          bne .leo
49f9      877:
49f9 a55a 878:         lda $5a
49fb c920 879:          cmp #32
49fd 9004 880:          bcc .do_look
49ff c97d 881:          cmp #125
4a01 9013 882:          bcc .do_here
4a03 a000 883: .do_look:      ldy #0
4a05 b90c4b 884: .lookup:      lda .char_table,y
4a08 f00c 885:          beq .do_here
4a0a c55a 886:          cmp $5a
4a0c f003 887:          beq .go_cio
4a0e c8 888:          iny
4a0f d0f4 889:          bne .lookup
4a11      890:
4a11      891:
4a11 a55a 892: .go_cio:      lda $5a
4a13 4c1c4b 893:          jmp putlocal
4a16      894:
4a16      895:
4a16 a000 896: .do_here:      ldy #0
4a18 a55d 897:         lda $5d
4a1a 915e 898:         sta ($5e),y
4a1c a55a 899:         lda $5a
4a1e c99b 900:          cmp #$9b
4a20 f044 901:          beq .docr
4a22 20eb4a 902: .notcr:      jsr .get_adr
4a25 a55a 903:         lda $5a
4a27 297f 904:          and #$7f
4a29 c920 905:          cmp #32
4a2b 9007 906:          bcc .add64
4a2d c960 907:          cmp #96
4a2f 9009 908:          bcc .sub32
4a31 4c3d4a 909:          jmp .asis
4a34 18 910: .add64:      clc
4a35 6940 911:          adc #64
4a37 4c3d4a 912:          jmp .asis
4a3a 38 913: .sub32:      sec
4a3b e920 914:          sbc #32
4a3d 245a 915: .asis:       bit $5a

```

```

4a3f 1002 916:      bpl  .xxlate
4a41 0980 917:      ora  #$80
4a43 a000 918: .xxlate:      ldy  #0
4a45 915e 919:      sta  ($5e),y
4a47 e655 920:      inc  $55
4a49 e663 921:      inc  $63
4a4b a555 922:      lda  $55
4a4d c928 923:      cmp  #40
4a4f b019 924:      bcs  .next_row
4a51 c8    925:      iny
4a52 b15e 926:      lda  ($5e),y
4a54 855d 927:      sta  $5d
4a56 0980 928:      ora  #$80
4a58 aef002 929:      ldx  752
4a5b d002 930:      bne  .nocurs1
4a5d 915e 931:      sta  ($5e),y
4a5f e65e 932: .nocurs1:      inc  $5e
4a61 d002 933:      bne  .nooav
4a63 e65f 934:      inc  $5e+1
4a65 60    935: .nooav:      rts
4a66 a900 936: .docr:      lda  #0
4a68 8563 937:      sta  $63
4a6a a552 938: .next_row:    lda  $52
4a6c 8555 939:      sta  $55
4a6e e654 940:      inc  $54
4a70 a454 941:      ldy  $54
4a72 c018 942:      cpy  #24
4a74 b013 943:      bcs  .scroll
4a76 20eb4a 944:      jsr  .get_adr
4a79 a000 945:      ldy  #0
4a7b b15e 946:      lda  ($5e),y
4a7d 855d 947:      sta  $5d
4a7f aef002 948:      ldx  752
4a82 d004 949:      bne  .nocurs2
4a84 0980 950:      ora  #$80
4a86 915e 951:      sta  ($5e),y
4a88 60    952: .nocurs2:    rts
4a89      953:
4a89 c654 954: .scroll:      dec  $54
4a8b a558 955:      lda  $58
4a8d 8568 956:      sta  $68
4a8f 18    957:      clc
4a90 6928 958:      adc  #40
4a92 855e 959:      sta  $5e
4a94 a559 960:      lda  $58+1
4a96 8569 961:      sta  $68+1
4a98 6900 962:      adc  #0
4a9a 855f 963:      sta  $5e+1
4a9c a000 964:      ldy  #0
4a9e a203 965:      ldx  #3
4aa0 b15e 966: .scloop:      lda  ($5e),y
4aa2 9168 967:      sta  ($68),y
4aa4 c8    968:      iny

```

```

4aa5 d0f9 969:      bne  .scloop
4aa7 e65f 970:      inc  $5e+1
4aa9 e669 971:      inc  $68+1
4aab ca 972:      dex
4aac d0f2 973:      bne  .scloop
4aae b15e 974: .scloop2:      lda  ($5e),y
4ab0 9168 975:      sta  ($68),y
4ab2 c8 976:      iny
4ab3 c098 977:      cpy  #152
4ab5 90f7 978:      bcc  .scloop2
4ab7 a558 979:      lda  $58
4ab9 18 980:      clc
4aba 6998 981:      adc  #920&$ff
4abc 855e 982:      sta  $5e
4abe a559 983:      lda  $58+1
4ac0 6903 984:      adc  #920/256
4ac2 855f 985:      sta  $5e+1
4ac4 a000 986:      ldy  #0
4ac6 98 987:      tya
4ac7 915e 988: .clear:      sta  ($5e),y
4ac9 c8 989:      iny
4aca c028 990:      cpy  #40
4acc 90f9 991:      bcc  .clear
4ace aef002 992:      ldx  752
4ad1 d006 993:      bne  .nocurs3
4ad3 a980 994:      lda  #$80
4ad5 a455 995:      ldy  $55
4ad7 915e 996:      sta  ($5e),y
4ad9 a900 997: .nocurs3:      lda  #0
4adb 855d 998:      sta  $5d
4add a55e 999:      lda  $5e
4adf 18 1000:      clc
4ae0 6555 1001:      adc  $55
4ae2 855e 1002:      sta  $5e
4ae4 a55f 1003:      lda  $5e+1
4ae6 6900 1004:      adc  #0
4ae8 855f 1005:      sta  $5e+1
4aea 60 1006:      rts
4aeb a554 1007: .get_adr:      lda  $54
4aed 0a 1008:      asl
4aee a8 1009:      tay
4aef a558 1010:      lda  $58
4af1 18 1011:      clc
4af2 79b949 1012:      adc  scr_offset,y
4af5 855e 1013:      sta  $5e
4af7 a559 1014:      lda  $58+1
4af9 79ba49 1015:      adc  scr_offset+1,y
4afc 855f 1016:      sta  $5e+1
4afe a55e 1017:      lda  $5e
4b00 18 1018:      clc
4b01 6555 1019:      adc  $55
4b03 855e 1020:      sta  $5e
4b05 a55f 1021:      lda  $5e+1

```

```

4b07 6900 1022:      adc  #0
4b09 855f 1023:      sta  $5e+1
4b0b 60   1024:      rts
4b0c      1025: .char_table:
4b0c 1b1c1d 1026:      dc.b  27,28,29,30,31,125,126,127
4b14 9c9d9e 1027:      dc.b  156,157,158,159,253,254,255
4b1c      1028:
4b1c      1029:
4b1c a20b 1030: putlocal:      ldx  #11
4b1e 8e4203 1031:      stx  $0342
4b21 a200 1032:      ldx  #0
4b23 8e4803 1033:      stx  $0348
4b26 8e4903 1034:      stx  $0349
4b29 4c56e4 1035:      jmp  $e456
4b2c      1036: ;
4b2c      1037:
4b2c      1038:
4b2c      1039:
4b2c      1040:
4b2c a904 1041: open_in:      lda  #4
4b2e 4c4a4b 1042:      jmp  do_open
4b31      1043:
4b31 a908 1044: open_out:      lda  #8
4b33 4c4a4b 1045:      jmp  do_open
4b36      1046:
4b36 a909 1047: open_app:      lda  #9
4b38 4c4a4b 1048:      jmp  do_open
4b3b      1049:
4b3b a90c 1050: open_updt:      lda  #12
4b3d 4c4a4b 1051:      jmp  do_open
4b40      1052:
4b40 a906 1053: open_dir:      lda  #6
4b42 4c4a4b 1054:      jmp  do_open
4b45      1055:
4b45 a914 1056: open_raw:      lda  #20      ;open raw directory
4b47 4c4a4b 1057:      jmp  do_open
4b4a      1058:
4b4a      1059: do_open
4b4a 48   1060:      pha      ;open filespec poin
4b4b 8a   1061:      txa      ;
4b4c 48   1062:      pha
4b4d 20774b 1063:      jsr  close      ;make sure device i
4b50 68   1064:      pla
4b51 aa   1065:      tax
4b52 68   1066:      pla
4b53 9d4a03 1067:      sta  icaux1,x
4b56 a903 1068:      lda  #ccopen
4b58 9d4203 1069:      sta  iccom,x
4b5b a586 1070:      lda  fn_ptr
4b5d 9d4403 1071:      sta  icbal,x
4b60 a587 1072:      lda  fn_ptr+1
4b62 9d4503 1073:      sta  icbah,x
4b65 a950 1074:      lda  #80

```

```

4b67 9d4803 1075:      sta  icbll,x
4b6a a900  1076:      lda  #0
4b6c 9d4903 1077:      sta  icblh,x
4b6f a980  1078:      lda  #128      ;for long directory
4b71 9d4b03 1079:      sta  icaux2,x
4b74 4c56e4 1080:      jmp  ciov
4b77      1081:
4b77      1082:
4b77      1083:
4b77      1084:
4b77      1085: ;
4b77      1086: close:                ; g
4b77 a90c  1087:      lda  #$0c
4b79 9d4203 1088:      sta  $0342,x
4b7c 2056e4 1089:      jsr  $e456
4b7f 60    1090:      rts
4b80      1091: ;
4b80      1092: ;
4b80      1093: TEMP1  ds.b 250
4c7a      1094: ;
4c7a      1095: TEMP2  ds.b 150
4d10      1096: ;
4d10 00    1097: DIR_COUNT dc.b 0
4d11      1098: ;
4d11      1099: DIR_PATH  ds.b 90 ;      dc.b "d5:>*.",$9B,0
4d6b      1100: ;
4d6b      1101: FILENAME ds.b 21
4d80      1102: ;
4d80      1103: DIR_MASK ds.b 172
4e2c      1104: ;
4e2c      1105: IN      ds.b 100      ;dc.b "d5:>memo.txt",$9B,0
4e90      1106: ;
4e90      1107: OUT     ds.b 190      ;dc.b "d5:>memo.bak",$9B,0
4f4e      1108: ;
4f4e 020d0a 1109: LF      dc.b 2,13,10
4f51 012e  1110: DOT     dc.b 1,"."
4f53 042e4a 1111: EXT     dc.b 4,".JNK"
4f58 042e4b 1112: EXT1    dc.b 4,".KNJ"
4f5d      1113: ;
4f5d 0120  1114: SPACE   dc.b 1," "
4f5f      1115: ;
4f5f      1116: ;      Move string pointed to by s_ptr to string pointed
4f5f      1117: ;      ~~~~~
4f5f      1118:
4f5f      1119:
4f5f a000  1120: move_str: ldy  #0
4f61 b182  1121:      lda  (s_ptr),y      ;get lth b
4f63 a8    1122:      tay                ;go backwa
4f64 b182  1123: .loop:   lda  (s_ptr),y
4f66 9184  1124:      sta  (d_ptr),y
4f68 c000  1125:      cpy  #0
4f6a f004  1126:      beq  .done
4f6c 88    1127:      dey

```



```

4f6d 4c644f 1128:      jmp    .loop
4f70 60    1129: .done:      rts
4f71      1130:
4f71      1131:
4f71      1132:
4f71      1133: ;   Return the length of string at s_ptr in accumulo
4f71      1134: ;   ~~~~~
4f71      1135:
4f71 a000 1136: length:      ldy    #0
4f73 b182 1137:              lda    (s_ptr),y
4f75 60    1138:              rts
4f76      1139:
4f76      1140:
4f76      1141:
4f76      1142: ;   Concatenate the string at s_ptr onto the end of st
4f76      1143: ;   ~~~~~
4f76      1144:
4f76 a000 1145: concat:      ldy    #0
4f78 b182 1146:              lda    (s_ptr),y      ;get sourc
4f7a 8db64f 1147:              sta    .srclth
4f7d f02a 1148:              beq    .end            ;null stri
4f7f b184 1149:              lda    (d_ptr),y      ;get dest
4f81 8db54f 1150:              sta    .desty
4f84 eeb44f 1151:              inc     .srcy
4f87 eeb54f 1152:              inc     .desty
4f8a a001 1153:              ldy    #1              ;always st
4f8c 8cb44f 1154:              sty     .srcy          ; the sour
4f8f acb44f 1155: .loop:       ldy     .srcy
4f92 b182 1156:              lda    (s_ptr),y      ;get byte
4f94 acb54f 1157:              ldy     .desty
4f97 9184 1158:              sta    (d_ptr),y      ;store int
4f99 adb44f 1159:              lda     .srcy
4f9c cdb64f 1160:              cmp     .srclth        ;end of st
4f9f f008 1161:              beq     .end            ;yes
4fa1 eeb44f 1162:              inc     .srcy
4fa4 eeb54f 1163:              inc     .desty        ;up indexe
4fa7 d0e6 1164:              bne     .loop          ;and keep
4fa9 a000 1165: .end:        ldy     #0
4fab b184 1166:              lda    (d_ptr),y      ;up the le
4fad 18    1167:              clc                    ; of the d
4fae 6db64f 1168:              adc     .srclth
4fb1 9184 1169:              sta    (d_ptr),y
4fb3 60    1170:              rts
4fb4      1171:
4fb4 01    1172: .srcy:       dc.b    1
4fb5 01    1173: .desty:      dc.b    1
4fb6 01    1174: .srclth:     dc.b    1
4fb7      1175:
4fb7      1176:
4fb7      1177:
4fb7      1178: ;   input a string from device in X register
4fb7      1179: ;   into the string pointed to by in_ptr.
4fb7      1180: ;   ~~~~~

```

```

4fb7      1181:
4fb7 9d4803 1182: inputsd:  sta  icbll,x      ;length is in the
4fba a905  1183:          lda  #ccgetrec
4fbc 9d4203 1184:          sta  iccom,x
4fbf a580  1185:          lda  in_ptr
4fc1 18    1186:          clc
4fc2 6901  1187:          adc  #1          ;offset past lengt
4fc4 9d4403 1188:          sta  icbal,x
4fc7 a581  1189:          lda  in_ptr+1
4fc9 6900  1190:          adc  #0
4fcb 9d4503 1191:          sta  icbah,x
4fce a900  1192:          lda  #0
4fd0 9d4903 1193:          sta  icblh,x      ;zero hi byte of l
4fd3 2056e4 1194:          jsr  ciov        ;get the line
4fd6 300b  1195:          bmi  .error
4fd8 bd4803 1196: .noerror:  lda  icbll,x      ;get lth actually
4fdb 38    1197:          sec
4fdc e901  1198:          sbc  #1          ; minus 1
4fde a000  1199:          ldy  #0
4fe0 9180  1200:          sta  (in_ptr),y    ; and place it in
4fe2 60    1201:          rts
4fe3      1202:
4fe3 8cef4f 1203: .error:    sty  .holdy
4fe6 a000  1204:          ldy  #0
4fe8 98    1205:          tya
4fe9 9180  1206:          sta  (in_ptr),y
4feb acef4f 1207:          ldy  .holdy
4fee 60    1208:          rts
4fef      1209:
4fef 00    1210: .holdy:    dc.b  0
4ff0      1211:
4ff0      1212:
4ff0      1213: ;    print a string to device in X register
4ff0      1214: ;    from the string pointed to by pr_ptr.
4ff0      1215: ;    ~~~~~
4ff0      1216:
4ff0 a000  1217: printsd:   ldy  #0
4ff2 b188  1218:          lda  (pr_ptr),y    ;install EOL at en
4ff4 a8    1219:          tay
4ff5 c8    1220:          iny
4ff6 a99b  1221:          lda  #$9b
4ff8 9188  1222:          sta  (pr_ptr),y
4ffa a909  1223:          lda  #ccputrec
4ffc 9d4203 1224:          sta  iccom,x
4fff a588  1225:          lda  pr_ptr
5001 18    1226:          clc
5002 6901  1227:          adc  #1          ;offset past lengt
5004 9d4403 1228:          sta  icbal,x
5007 a589  1229:          lda  pr_ptr+1
5009 6900  1230:          adc  #0
500b 9d4503 1231:          sta  icbah,x
500e a950  1232:          lda  #80
5010 9d4803 1233:          sta  icbll,x

```

```

5013 a900 1234:      lda  #0
5015 9d4903 1235:      sta  icblh,x      ;zero hi byte of l
5018 4c56e4 1236:      jmp   ciov      ;get the line
501b      1237:
501b      1238: ;      Search the string pointed to by S_PTR for the subs
501b      1239: ;      pointed to by D_PTR. Returns a 0 if substring not
501b      1240: ;      ~~~~~
501b      1241:
501b a000 1242: substr:  ldy  #0
501d b182 1243:      lda  (s_ptr),y
501f 8d8f50 1244:      sta  .s_lth
5022 b184 1245:      lda  (d_ptr),y
5024 8d9050 1246:      sta  .d_lth
5027 a582 1247:      lda  s_ptr
5029 8d4b50 1248:      sta  .comp+1
502c a583 1249:      lda  s_ptr+1
502e 8d4c50 1250:      sta  .comp+2
5031 a584 1251:      lda  d_ptr
5033 8d5450 1252:      sta  .get+1
5036 a585 1253:      lda  d_ptr+1
5038 8d5550 1254:      sta  .get+2
503b a901 1255:      lda  #1
503d 8d8e50 1256:      sta  .loc
5040 a000 1257:      ldy  #0
5042 a200 1258:      ldx  #0
5044 207250 1259:      jsr  .up_str
5047 207f50 1260:      jsr  .up_sub
504a      1261:
504a b9ffff 1262: .comp:  lda  $ffff,y
504d 209150 1263:      jsr  upcase
5050 8d8d50 1264:      sta  .temp
5053 bdffff 1265: .get:   lda  $ffff,x
5056 209150 1266:      jsr  upcase
5059 cd8d50 1267:      cmp  .temp
505c f00b 1268:      beq  .match
505e a201 1269: .nomatch: ldx  #1
5060 ee8e50 1270:      inc  .loc
5063 207250 1271:      jsr  .up_str
5066 4c4a50 1272:      jmp  .comp
5069 207f50 1273: .match:  jsr  .up_sub
506c 207250 1274:      jsr  .up_str
506f 4c4a50 1275:      jmp  .comp
5072      1276:
5072 c8 1277: .up_str:  iny
5073 cc8f50 1278:      cpy  .s_lth
5076 9006 1279:      bcc  .xup
5078 f004 1280:      beq  .xup
507a 68 1281:      pla
507b 68 1282:      pla
507c a900 1283:      lda  #0
507e 60 1284: .xup:   rts
507f      1285:
507f e8 1286: .up_sub:  inx

```

```

5080 ec9050 1287:      cpx  .d_lth
5083 9007  1288:      bcc  .xsub
5085 f005  1289:      beq  .xsub
5087 68    1290:      pla
5088 68    1291:      pla
5089 ad8e50 1292:      lda  .loc
508c 60    1293: .xsub:  rts
508d      1294:
508d 00    1295: .temp:  dc.b  0
508e 00    1296: .loc:   dc.b  0
508f 00    1297: .s_lth: dc.b  0
5090 00    1298: .d_lth: dc.b  0
5091      1299:
5091 c961  1300: upcase:  cmp   #'a'
5093 9009  1301:      bcc  xupcase
5095 c97b  1302:      cmp  #'z'+1
5097 f005  1303:      beq  xupcase
5099 b003  1304:      bcs  xupcase
509b 38    1305:      sec
509c e920  1306:      sbc  #32
509e 60    1307: xupcase: rts
509f      1308:
509f      1309: ;   Make a length-byte string from an ASCII ($9b termina
509f      1310: ;   ~~~~~
509f      1311:
509f a000  1312: tostr:  ldy   #0
50a1 b182  1313: .tfe:   lda   (s_ptr),y      ;find end
50a3 f007  1314:      beq  .gotend
50a5 c99b  1315:      cmp  #$9b
50a7 f003  1316:      beq  .gotend
50a9 c8    1317:      iny
50aa d0f5  1318:      bne  .tfe
50ac 8cc750 1319: .gotend: sty   .lth      ;save leng
50af c000  1320:      cpy  #0
50b1 f00c  1321:      beq  .rtn
50b3 a000  1322:      ldy  #0
50b5 b182  1323: .loop:  lda   (s_ptr),y
50b7 c8    1324:      iny
50b8 9184  1325:      sta  (d_ptr),y
50ba ccc750 1326:      cpy  .lth
50bd 90f6  1327:      bcc  .loop
50bf a000  1328: .rtn:   ldy  #0
50c1 adc750 1329:      lda  .lth
50c4 9184  1330:      sta  (d_ptr),y
50c6 60    1331:      rts
50c7      1332:
50c7 01    1333: .lth:   dc.b  1
50c8      1334:
50c8      1335:
50c8      1336:
50c8      1337: WIN_END: ds.b  0
50c8      1338: ;
50c8      1339: ;

```

50c8 1340:

End assembly: no errors