


```

0000 1: ;-----
0000 2: ;
0000 3: ;      Copyright 2012 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: ;      PortTest.s
0000 10: ;
0000 11: ;Wrote this to test the Joy Stick Ports.
0000 12: ;First Design 10,19,2012
0000 13: ;
0000 14: ;      ~~~~~
0000 15: ;-----
0000 16: ; Notes: o This source code MAY NOT be placed for download
0000 17: ;      o "mea" is a macro that loads the address of the
0000 18: ;      into the pointer specified by the second field.
0000 19: ;-----
0000 20: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 21: ;      XASM st and IBM VERSIONS
0000 22: ;-----
0000 23: ;=====
0000 24:
0000 25: com_file_name:      .macro
0000 26: ;      dc.b 13,"filename.ext",$9b
0000 27: ;      dc.b 12,"PortTest.com",$9b
0000 28: ;      .endm
0000 29:
0000 30: ;
0000 31: ;~~~~~
0000 32: ; File revision history. Version Number in hex
0000 33: ;
0000 34: file_ver:      equ $10
0000 35: ;~~~~~
0000 36: ; the Month in dec for the first time this revision
0000 37: ;was compiled
0000 38: ;
0000 39: c_month:      equ 10
0000 40: ;~~~~~
0000 41: ; the day in dec for the first time this revision
0000 42: ;was compiled
0000 43: c_day:      equ 20
0000 44: ;~~~~~
0000 45: ; the year in dec for the first time this revision
0000 46: ;was compiled
0000 47: c_year:      equ 2012
0000 48: ;~~~~~
0000 49: ; Type of compiler used in program
0000 50: ;
0000 51: ; 0 = unknown Compiler
0000 52: ; 1 = xasm
0000 53: ; 2 = mac_65

```

```

0000      54: ; 3 = basic
0000      55: ; 4 = compiled basic xl
0000      56: ; 5 = C65
0000      57: ; 6 = Action
0000      58: ;
0000      59: ; We can add more as time goes on!
0000      60: ;
0000      61: ;
0000      62: ;
0000      63: xasm:          equ    1
0000      64: mac_65:          equ    2
0000      65: basic:      equ    3
0000      66: basicxl:      equ    4
0000      67: c65:         equ    5
0000      68: action:        equ    6
0000      69:
0000      70: file_compiler:    equ    xasm
0000      71: ;~~~~~
0000      72: ; is this relocatable code ?
0000      73: ; anyother value other than 1 or 2 would be unknown
0000      74: r..yes:           equ    1
0000      75: r..no:            equ    2
0000      76: ;
0000      77: ;~~~~~
0000      78: ; Gotta define if it can be relocated
0000      79: l_relocatable:    equ    r..no
0000      80: ;
0000      81: ;
0000      82: ;~~~~~
0000      83: ; Gotta define if it can be relocated
0000      84: r..crl:           equ    $7d
0000      85: r..crlf:          equ    $9b
0000      86: r..space:         equ    $20
0000      87:
0000      88: l_frstscreenbyte: equ    r..crl
0000      89: ;~~~~~
0000      90: ; The language the output file is in.
0000      91: ;
0000      92: ;
0000      93: ; 0 =   Undefined
0000      94: ; 1 =   English
0000      95: ; 2 =   German
0000      96: ;
0000      97: r..language:      equ    1
0000      98: ;~~~~~
0000      99:
0000     100:
0000     101:          .include    equates
0000     102:          .include    globals
0000     103:          .include    macros
0000     104:
0000     105:          .org    $4000
0000     106:

```

```

4000      107: win_start:
4000      108: header_info:
4000      109:          .include      header ; this read
42a7      110: ;          ; based on
42a7 a960  111: begin_port:      lda    #$60
42a9 8da742 112:          sta    begin_port
42ac 201447 113:          jsr    pin_reset
42af 20e546 114:          jsr    pin_off
42b2 a901  115:          lda    #$01
42b4 8df002 116:          sta    $2f0
42b7      117:
42b7      118:
42b7      119: top:
42b7 a200  120:          ldx    #0
42b9 a006  121:          ldy    #6
42bb 20bc46 122:          jsr    gotoxy
42be 20fa48 123:          jsr    printsi
42c1      124: ;          11111111112222222
42c1      125: ;          12345678901234567890123456
42c1 205468 126:          dc.b   " The purpose of this progr
42e9 207468 127:          dc.b   " the Joy Stick Port and Tr
4310 9b9b  128:          dc.b   $9b,$9b
4312 dbd2dd 129:          dc.b   <+128>,"[R]"
4315 204469 130:          dc.b   " Displays Joystick results
4331 9b  131:          dc.b   $9b
4332 dbd7dd 132:          dc.b   <+128>,"[W]"
4335 204a6f 133:          dc.b   " Joystick Ports in Write M
4354 9b  134:          dc.b   $9b
4355 dbc8dd 135:          dc.b   <+128>,"[H]"
4358 6f7720 136:          dc.b   "ow to Build Display Box!",
4372 dbd1dd 137:          dc.b   <+128>,"[Q]"
4375 756974 138:          dc.b   "uit to Dos",$9b,$9b,$9b
4382 d9eff5 139:          dc.b   <+128>,"Your Selection?"
4391 2020  140:          dc.b   " "
4393 ff  141:          dc.b   -1
4394 20934a 142: .again:      jsr    getkey
4397 204448 143:          jsr    upcase
439a c952  144:          cmp    #'R'
439c f012  145:          beq    .r
439e c957  146:          cmp    #'W'
43a0 f011  147:          beq    .w
43a2 c948  148:          cmp    #'H'
43a4 f010  149:          beq    .h
43a6 c951  150:          cmp    #'Q'
43a8 f012  151:          beq    .q
43aa c91b  152:          cmp    #$1b
43ac f00b  153:          beq    .bye
43ae d0e4  154:          bne    .again
43b0      155:
43b0      156: .r:
43b0 4c5645 157:          jmp    Joy_test
43b3      158: .w:
43b3 4cc843 159:          jmp    Joy_write

```

```

43b6      160: .H:
43b6 4ca84a 161:      jmp  help_menu
43b9 4c74e4 162: .bye:      jmp  $e474
43bc      163: .Q:
43bc a900  164:      lda  #$00      ; Zero is w
43be 8df002 165:      sta  $2f0      ; turn the
43c1      166:
43c1 20fa48 167:      jsr  printsi
43c4 9b    168:      dc.b  $9b
43c5 9b    169:      dc.b  $9b
43c6 ff    170:      dc.b  -1
43c7 60    171:      rts
43c8      172:
43c8      173:
43c8      174:
43c8      175:
43c8      176:
43c8      177:
43c8      178: Joy_write:
43c8 201447 179:      jsr  pin_reset
43cb 20f046 180:      jsr  pin_on
43ce a97d   181:      lda  #$7d
43d0 205049 182:      jsr  echo
43d3 20a442 183:      jsr  begin_port-3
43d6      184:
43d6 a200   185:      ldx  #0
43d8 a007   186:      ldy  #7
43da 20bc46 187:      jsr  gotoxy
43dd 20fa48 188:      jsr  printsi
43e0      189: ;      11111111112222222
43e0      190: ;      12345678901234567890123456
43e0 205468 191:      dc.b  " The purpose of this Menu
4406 202074 192:      dc.b  " to the Joy Stick Port an
442a 9b9bff 193:      dc.b  $9b,$9b,-1
442d 20f046 194:      jsr  pin_on
4430 201447 195:      jsr  pin_reset
4433 20f046 196:      jsr  pin_on
4436      197:
4436 a205   198: .loop:      ldx  #5
4438 a00e   199:      ldy  #14
443a 20bc46 200:      jsr  gotoxy
443d 20fa48 201:      jsr  printsi
4440 204c6f 202:      dc.b  " Low Trigger is ==>",-1
4454 206846 203:      jsr  low_trigger
4457      204:
4457 a205   205:      ldx  #5
4459 a00f   206:      ldy  #15
445b 20bc46 207:      jsr  gotoxy
445e 20fa48 208:      jsr  printsi
4461 486967 209:      dc.b  "High Trigger is ==>",-1
4475 204f46 210:      jsr  high_trigger
4478      211:
4478 202645 212:      jsr  Show_portA

```

```
-----
447b 20c146 213:      jsr  pin_1
447e 202645 214:      jsr  Show_portA
4481 208146 215:      jsr  delay
4484 20ca46 216:      jsr  pin_2
4487 202645 217:      jsr  Show_portA
448a 208146 218:      jsr  delay
448d 20d346 219:      jsr  pin_3
4490 202645 220:      jsr  Show_portA
4493 208146 221:      jsr  delay
4496 20dc46 222:      jsr  pin_4
4499 202645 223:      jsr  Show_portA
449c 208146 224:      jsr  delay
449f 201a47 225:      jsr  pin_5
44a2 202645 226:      jsr  Show_portA
44a5 208146 227:      jsr  delay
44a8 202347 228:      jsr  pin_6
44ab 202645 229:      jsr  Show_portA
44ae 208146 230:      jsr  delay
44b1 202c47 231:      jsr  pin_7
44b4 202645 232:      jsr  Show_portA
44b7 208146 233:      jsr  delay
44ba 203547 234:      jsr  pin_8
44bd 202645 235:      jsr  Show_portA
44c0 208146 236:      jsr  delay
44c3      237:
44c3 20c146 238:      jsr  pin_1
44c6 202645 239:      jsr  Show_portA
44c9 208146 240:      jsr  delay
44cc 20ca46 241:      jsr  pin_2
44cf 202645 242:      jsr  Show_portA
44d2 208146 243:      jsr  delay
44d5 20d346 244:      jsr  pin_3
44d8 202645 245:      jsr  Show_portA
44db 208146 246:      jsr  delay
44de 20dc46 247:      jsr  pin_4
44e1 202645 248:      jsr  Show_portA
44e4 208146 249:      jsr  delay
44e7 201a47 250:      jsr  pin_5
44ea 202645 251:      jsr  Show_portA
44ed 208146 252:      jsr  delay
44f0 202347 253:      jsr  pin_6
44f3 202645 254:      jsr  Show_portA
44f6 208146 255:      jsr  delay
44f9 202c47 256:      jsr  pin_7
44fc 202645 257:      jsr  Show_portA
44ff 208146 258:      jsr  delay
4502 203547 259:      jsr  pin_8
4505 202645 260:      jsr  Show_portA
4508 208146 261:      jsr  delay
450b      262:
450b      263:
450b      264:
450b adfc02 265:      lda  764
```

```

450e c9ff 266:      cmp  #$ff
4510 d003 267:      bne  .quit
4512 4c3644 268:     jmp  .loop
4515      269: .quit:
4515 a9ff 270:      lda  #$ff
4517 8dfc02 271:     sta  764
451a      272:
451a a97d 273:      lda  #$7d
451c 205049 274:     jsr  echo
451f 20a442 275:     jsr  begin_port-3
4522 4cb742 276:     jmp  top
4525      277:
4525 60 278:      rts
4526      279:
4526      280: Show_portA:
4526 a200 281:      ldx  #0
4528 a014 282:      ldy  #20
452a 20bc46 283:     jsr  gotoxy
452d 20fa48 284:     jsr  printsi
4530 4c6967 285:     dc.b  "Lighting up The LED's ",-1
4547      286:
4547 204247 287:     jsr  check_bit
454a      288:     mea  show_bits,pr_ptr
454a a9b7 ----:     lda  #low show_bits
454c 8588 ----:     sta  pr_ptr
454e a947 ----:     lda  #high show_bits
4550 8589 ----:     sta  pr_ptr + 1
4552 206048 289:     jsr  echos
4555 60 290:      rts
4556      291:
4556      292:
4556      293: Joy_test:
4556 201447 294:     jsr  pin_reset
4559 20e546 295:     jsr  pin_off
455c      296:
455c a900 297:     lda  #$00
455e 8552 298:     sta  $52
4560 a97d 299:     lda  #$7d
4562 205049 300:     jsr  echo
4565 20a442 301:     jsr  begin_port-3
4568      302:
4568 a200 303:     ldx  #0
456a a007 304:     ldy  #7
456c 20bc46 305:     jsr  gotoxy
456f 20fa48 306:     jsr  printsi
4572      307: ;      11111111112222222
4572      308: ;      12345678901234567890123456
4572 205468 309:     dc.b  " The purpose of this progr
459a 202020 310:     dc.b  " the Joy Stick Port an
45be 9b9bff 311:     dc.b  $9b,$9b,-1
45c1      312:
45c1 a205 313: .loop:     ldx  #5
45c3 a00a 314:     ldy  #10

```

```

45c5 20bc46 315:      jsr  gotoxy
45c8 20fa48 316:      jsr  printsi
45cb 204c6f 317:      dc.b  " Low Trigger is ==>",-1
45df 206846 318:      jsr  low_trigger
45e2      319:
45e2 a205 320:      ldx  #5
45e4 a00b 321:      ldy  #11
45e6 20bc46 322:      jsr  gotoxy
45e9 20fa48 323:      jsr  printsi
45ec 486967 324:      dc.b  "High Trigger is ==>",-1
4600 204f46 325:      jsr  high_trigger
4603      326:
4603 a205 327:      ldx  #5
4605 a014 328:      ldy  #20
4607 20bc46 329:      jsr  gotoxy
460a 20fa48 330:      jsr  printsi
460d 204a6f 331:      dc.b  " Joy Stick Port test ==>",
4626      332:
4626 204247 333:      jsr  check_bit
4629      334:      mea  show_bits,pr_ptr
4629 a9b7 ----:      lda  #low show_bits
462b 8588 ----:      sta  pr_ptr
462d a947 ----:      lda  #high show_bits
462f 8589 ----:      sta  pr_ptr + 1
4631 206048 335:      jsr  echos
4634      336:
4634 adfc02 337:      lda  764
4637 c9ff 338:      cmp  #$ff
4639 d003 339:      bne  .quit
463b 4cc145 340:      jmp  .loop
463e      341: .quit:
463e a9ff 342:      lda  #$ff
4640 8dfc02 343:      sta  764
4643      344:
4643 a97d 345:      lda  #$7d
4645 205049 346:      jsr  echo
4648 20a442 347:      jsr  begin_port-3
464b 4cb742 348:      jmp  top
464e      349:
464e 60 350:      rts
464f      351:
464f      352:
464f      353:
464f      354:
464f      355:
464f      356:
464f      357:
464f      358:
464f      359:
464f      360: high_trigger:
464f ad11d0 361:      lda  $D011
4652 c901 362:      cmp  #$01
4654 f009 363:      beq  .a

```



```

4656 20fa48 364:      jsr  printsi
4659 cfce 365:      dc.b  <+128>,"ON"
465b 209bff 366:      dc.b  " ",$9b,-1
465e 60 367:      rts
465f 368: .a:
465f 20fa48 369:      jsr  printsi
4662 cfc6c6 370:      dc.b  <+128>,"OFF"
4665 9bff 371:      dc.b  $9b,-1
4667 372:
4667 60 373:      rts
4668 374: low_trigger:
4668 ad10d0 375:      lda  $D010
466b c901 376:      cmp  #$01
466d f009 377:      beq  .a
466f 20fa48 378:      jsr  printsi
4672 cfce 379:      dc.b  <+128>,"ON"
4674 209bff 380:      dc.b  " ",$9b,-1
4677 60 381:      rts
4678 382: .a:
4678 20fa48 383:      jsr  printsi
467b cfc6c6 384:      dc.b  <+128>,"OFF"
467e 9bff 385:      dc.b  $9b,-1
4680 386:
4680 60 387:      rts
4681 388:
4681 389: delay:
4681 a900 390:      lda  #$00
4683 8514 391:      sta  $14
4685 392: one_second:
4685 adfc02 393:      lda  764
4688 c9ff 394:      cmp  #$ff
468a d006 395:      bne  d.out
468c a914 396: seconds:      lda  #20          ;60
468e c514 397:      cmp  $14
4690 d0f3 398:      bne  one_second
4692 399: d.out:
4692 60 400:      rts
4693 401: ;=====
4693 402: ;
4693 403: ;Draw out a big plus to look like what a joystick
4693 404: ;would do
4693 405: ;-----
4693 406: Left_arry:
4693 407: ;      lda
4693 408:
4693 409:
4693 60 410:      rts
4694 411:
4694 412:
4694 413: ;-----
4694 414: ; Print Hex Numbers to screen byte or card
4694 415: ;-----
4694 416: ;

```

```

4694      417: ;
4694      418: ; print a hex word value
4694      419: ; low byte in a reg x is high byte
4694      420: ;
4694 8da246 421: pr_hex_card:      sta  .h_cardd
4697 8a     422:                txa
4698 20a346 423:                jsr  pr_hex_byte
469b ada246 424:                lda  .h_cardd
469e 20a346 425:                jsr  pr_hex_byte
46a1 60     426:                rts
46a2      427: ;
46a2 00     428: .h_cardd:      dc.b  $00
46a3      429: ;
46a3 48     430: pr_hex_byte:   pha
46a4 4a     431:                lsr
46a5 4a     432:                lsr
46a6 4a     433:                lsr
46a7 4a     434:                lsr
46a8 20ae46 435:                jsr  .a
46ab 68     436:                pla
46ac 290f   437:                and  #$0f
46ae c90a   438: .a:           cmp  #$0a
46b0 b004   439:                bcs  .b
46b2 0930   440:                ora  #$30
46b4 d002   441:                bne  .c
46b6 6936   442: .b:           adc  #$36
46b8 205049 443: .c:           jsr  echo
46bb 60     444:                rts
46bc      445:
46bc      446:
46bc      447: ;-----
46bc      448:
46bc      449:
46bc 8655   450: gotoxy:       stx  85
46be 8454   451:                sty  84
46c0 60     452:                rts
46c1      453:
46c1      454:
46c1      455: ;
46c1      456: ;-----
46c1      457: ; turn pin 1 on/off
46c1      458: ;-----
46c1      459: pin_1:
46c1 ad00d3 460:                lda  $d300
46c4 4910   461:                eor  #$10
46c6 8d00d3 462:                sta  $d300
46c9 60     463:                rts
46ca      464: ;
46ca      465: ;
46ca      466: ;-----
46ca      467: ; TURN PIN 2 ON/off
46ca      468: ;-----
46ca      469: pin_2:

```

```

46ca ad00d3 470:          lda  $d300
46cd 4920 471:          eor  #$20
46cf 8d00d3 472:          sta  $d300
46d2 60 473:            rts
46d3      474: ;
46d3      475: ;-----
46d3      476: ;TURN PIN 3 ON/off
46d3      477: ;-----
46d3      478: pin_3:
46d3 ad00d3 479:          lda  $d300
46d6 4940 480:          eor  #$40
46d8 8d00d3 481:          sta  $d300
46db 60 482:            rts
46dc      483: ;
46dc      484: ;-----
46dc      485: ;TURN PIN 4 ON/off
46dc      486: ;-----
46dc      487: pin_4:
46dc ad00d3 488:          lda  $d300
46df 4980 489:          eor  #$80
46e1 8d00d3 490:          sta  $d300
46e4 60 491:            rts
46e5      492: ;
46e5      493: ;-----
46e5      494: ;PORT off
46e5      495: ;-----
46e5      496: pin_off:
46e5 a900 497:          lda  #$00
46e7 8d00d3 498:          sta  $d300
46ea a93c 499:          lda  #$3c
46ec 8d02d3 500:          sta  $d302
46ef 60 501:            rts
46f0      502: ;
46f0      503: ;-----
46f0      504: ;PORT ON
46f0      505: ;-----
46f0      506: pin_on:
46f0 ad02d3 507:          lda  $d302
46f3 38 508:            sec
46f4 e904 509:          sbc  #$04
46f6 8d02d3 510:          sta  $d302
46f9 a90a 511:          lda  #$0a
46fb 8d02d3 512:          sta  $d302
46fe a9f0 513:          lda  #$f0
4700 8d00d3 514:          sta  $d300
4703 a900 515:          lda  #$00
4705 8d00d3 516:          sta  $d300
4708 60 517:            rts
4709      518: ;-----
4709      519: ;FOR WHEN YOU NEED LESS
4709      520: ;THAN A JIFFY
4709      521: ;-----
4709 a900 522: spause:          lda  #$00

```

```

470b 6901 523: .splop:          adc  #$01
470d c960 524:                cmp  #$60
470f ea 525:                nop
4710 ea 526:                nop
4711 90f8 527:                bcc  .splop
4713 60 528:                rts
4714     529: ;
4714     530: ;-----
4714     531: ; PIN RESET
4714     532: ;-----
4714     533: pin_reset:
4714 a900 534:                lda  #$00
4716 8d00d3 535:                sta  $d300
4719 60 536:                rts
471a     537: ;-----
471a     538: ; PIN 5 ON/OFF
471a     539: ;-----
471a     540: pin_5:
471a ad00d3 541:                lda  $d300
471d 4901 542:                eor  #$01
471f 8d00d3 543:                sta  $d300
4722 60 544:                rts
4723     545: ;-----
4723     546: ; PIN 6 ON/OFF
4723     547: ;-----
4723     548: pin_6:
4723 ad00d3 549:                lda  $d300
4726 4902 550:                eor  #$02
4728 8d00d3 551:                sta  $d300
472b 60 552:                rts
472c     553: ;-----
472c     554: ; PIN 7 ON/OFF
472c     555: ;-----
472c     556: pin_7:
472c ad00d3 557:                lda  $d300
472f 4904 558:                eor  #$04
4731 8d00d3 559:                sta  $d300
4734 60 560:                rts
4735     561: ;-----
4735     562: ; PIN 8 ON/OFF
4735     563: ;-----
4735     564: pin_8:
4735 ad00d3 565:                lda  $d300
4738 4908 566:                eor  #$08
473a 8d00d3 567:                sta  $d300
473d 60 568:                rts
473e     569: ;
473e     570: ;-----
473e     571: ; LOAD PIN
473e     572: ;-----
473e     573: load_pin:
473e 8d00d3 574:                sta  $d300
4741 60 575:                rts

```

```

4742      576: ;-----
4742      577: ; BIT LOGIC LOAD ACC
4742      578: ; LOAD Y REG WITH BIT #
4742      579: ;-----
4742      580: check_bit:
4742 ad00d3 581:      lda    $d300          ;po
4745 a007 582:      ldy    #7
4747 8db547 583:      sta    check_hold
474a 8cb647 584:      sty    check_hold+1
474d a000 585:      ldy    #$00
474f a994 586:      lda    #$94          ;'1
4751 99b847 587: check1:      sta    check_line,y
4754 c8 588:      iny
4755 c008 589:      cpy    #$08
4757 90f8 590:      bcc    check1
4759 adb547 591:      lda    check_hold
475c 18 592:      clc
475d 0a 593:      asl
475e 9006 594:      bcc    c_bit_1
4760 a014 595:      ldy    #$14          ;'0
4762 8cbf47 596:      sty    check_line+7
4765 18 597:      clc
4766      598: c_bit_1:
4766 0a 599:      asl
4767 9006 600:      bcc    c_bit_2
4769 a014 601:      ldy    #$14          ;'0
476b 8cbe47 602:      sty    check_line+6
476e 18 603:      clc
476f      604: c_bit_2:
476f 0a 605:      asl
4770 9006 606:      bcc    c_bit_3
4772 a014 607:      ldy    #$14          ;'0
4774 8cbd47 608:      sty    check_line+5
4777 18 609:      clc
4778      610: c_bit_3:
4778 0a 611:      asl
4779 9006 612:      bcc    c_bit_4
477b a014 613:      ldy    #$14          ;'0
477d 8cbc47 614:      sty    check_line+4
4780 18 615:      clc
4781      616: c_bit_4:
4781 0a 617:      asl
4782 9006 618:      bcc    c_bit_5
4784 a014 619:      ldy    #$14          ;'0
4786 8cbb47 620:      sty    check_line+3
4789 18 621:      clc
478a      622: c_bit_5:
478a 0a 623:      asl
478b 9006 624:      bcc    c_bit_6
478d a014 625:      ldy    #$14          ;'0
478f 8cba47 626:      sty    check_line+2
4792 18 627:      clc
4793      628: c_bit_6:

```

```

4793 0a 629:          asl
4794 9006 630:          bcc  c_bit_7
4796 a014 631:          ldy  #$14          ;'0
4798 8cb947 632:          sty  check_line+1
479b 18 633:          clc
479c      634: c_bit_7:
479c c900 635:          cmp  #$00
479e f006 636:          beq  c_bit_8
47a0 a014 637:          ldy  #$14          ;'0
47a2 8cb847 638:          sty  check_line
47a5 18 639:          clc
47a6      640: c_bit_8:
47a6 18 641:          clc
47a7 acb647 642:          ldy  check_hold+1
47aa b9b847 643:          lda  check_line,y
47ad c901 644:          cmp  #$01
47af d002 645:          bne  c_bit_9
47b1 38 646:          sec
47b2 60 647:          rts
47b3      648:
47b3 18 649: c_bit_9:          clc
47b4 60 650:          rts
47b5      651:
47b5      652:
47b5 0000 653: check_hold:      dc.b  $00,$00
47b7 08 654: show_bits:      dc.b  8
47b8 000000 655: check_line:      dc.b  0,0,0,0
47bc 000000 656:                  dc.b  0,0,0,0
47c0      657:
47c0      658: driverror:
47c0 48 659:          pha
47c1 20fa48 660:          jsr  printsi
47c4 9b2053 661:          dc.b  $9b," SioV Error # ", $FF
47d4 68 662:          pla
47d5 20de47 663:          jsr  pr_byte
47d8 20fa48 664:          jsr  printsi
47db 9bff 665:          dc.b  $9b,-1
47dd 60 666:          rts
47de      667:
47de      668:
47de      669:
47de      670: ;
47de      671: ;
47de      672: ;
47de      673: pr_byte:
47de a200 674:          ldx  #$00
47e0      675: pr_card:
47e0 85d4 676:          sta  $d4
47e2 86d5 677:          stx  $d4+1
47e4 a900 678:          lda  #$00
47e6 85f2 679:          sta  $f2
47e8 20aad9 680:          jsr  $d9aa
47eb 20e6d8 681:          jsr  $d8e6

```

```

47ee a000 682:          ldy  #$00
47f0      683: .a:
47f0 b1f3 684:          lda  ($f3),y
47f2 08   685:          php
47f3 297f 686:          and  #$7f
47f5 c8   687:          iny
47f6 205049 688:          jsr  echo
47f9 28   689:          plp
47fa 10f4 690:          bpl  .a
47fc 60   691:          rts
47fd      692: ;
47fd      693:
47fd      694: ; this unifies the length string
47fd      695: ; ~~~~~
47fd      696:
47fd      697: echos.unify:
47fd a000 698:          ldy  #0
47ff b188 699:          lda  (pr_ptr),y      ;ge
4801 f03e 700:          beq  .xpri          ;ex
4803 8d4348 701:          sta  .maxlth
4806 c8   702:          iny
4807 8c4248 703:          sty  .tempy
480a      704:
480a ac4248 705:          ldy  .tempy
480d b188 706:          lda  (pr_ptr),y
480f 204448 707:          jsr  upcase
4812 9188 708:          sta  (pr_ptr),y
4814 4c3948 709:          jmp  .prnext
4817      710:
4817 ac4248 711: .pril:          ldy  .tempy
481a c002 712:          cpy  #$02
481c 901b 713:          bcc  .prnext
481e b188 714:          lda  (pr_ptr),y      ; c
4820 205248 715:          jsr  locase
4823 9188 716:          sta  (pr_ptr),y
4825      717:
4825 c920 718:          cmp  #' '          ; l
4827 d010 719:          bne  .prnext
4829 ee4248 720:          inc  .tempy
482c ce4348 721:          dec  .maxlth
482f ac4248 722:          ldy  .tempy
4832 b188 723:          lda  (pr_ptr),y      ; m
4834 204448 724:          jsr  upcase
4837 9188 725:          sta  (pr_ptr),y
4839      726:
4839      727:
4839 ee4248 728: .prnext:          inc  .tempy
483c ce4348 729:          dec  .maxlth
483f d0d6 730:          bne  .pril
4841 60   731: .xpri:          rts
4842      732:
4842 01   733: .tempy:          dc.b  1
4843 01   734: .maxlth:         dc.b  1

```

```

4844      735:
4844      736:
4844      737: ;
4844      738: ;
4844 c961  739: upcase:          cmp    #'a'
4846 9009  740:          bcc    .xupcase
4848 c97b  741:          cmp    #'z'+1
484a f005  742:          beq    .xupcase
484c b003  743:          bcs    .xupcase
484e 38    744:          sec
484f e920  745:          sbc    #32
4851 60    746: .xupcase:      rts
4852      747:
4852      748:
4852 c941  749: locase:        cmp    #'A'
4854 9009  750:          bcc    .xlocase
4856 c95b  751:          cmp    #'Z'+1
4858 f005  752:          beq    .xlocase
485a b003  753:          bcs    .xlocase
485c 18    754:          clc
485d 6920  755:          adc    #32
485f 60    756: .xlocase:      rts
4860      757:
4860      758:
4860      759:
4860      760:
4860      761: ;
4860      762: ; prints: prints string with address in PR_PTR, length by
4860      763: ; ~~~~~
4860      764:
4860      765: echos:
4860 a000  766:          ldy    #0
4862 b188  767:          lda    (pr_ptr),y      ;ge
4864 f01b  768:          beq    .xpri           ;ex
4866 8d8348 769:          sta    .maxlth
4869 c8    770:          iny
486a 8c8248 771:          sty    .tempy
486d      772:
486d ac8248 773: .pril:        ldy    .tempy
4870 b188  774:          lda    (pr_ptr),y
4872 c900  775:          cmp    #$00
4874 f00b  776:          beq    .xpri
4876 205049 777:          jsr    echo
4879 ee8248 778: .prnext:      inc    .tempy
487c ce8348 779:          dec    .maxlth
487f d0ec  780:          bne    .pril
4881 60    781: .xpri:        rts
4882      782:
4882 01     783: .tempy:        dc.b  1
4883 01     784: .maxlth:      dc.b  1
4884      785:
4884 00     786: pr_mode:      dc.b  0
4885      787:

```



```

4885      788:
4885      789: ;
4885      790: ;
4885      791: ;-----
4885      792: ;this one make's a nice tone
4885      793: ;-----
4885      794: bell
4885 8db148 795:          sta  .data
4888 8cb248 796:          sty  .data+1
488b 8eb348 797:          stx  .data+2
488e a0af 798:          ldy  #$af
4890 a914 799:          lda  #$14
4892 8d00d2 800:          sta  $d200
4895 8c01d2 801: .a:          sty  $d201
4898 a900 802:          lda  #$00
489a 8514 803:          sta  $14
489c a514 804: .b:          lda  $14
489e c902 805:          cmp  #$02
48a0 90fa 806:          bcc  .b
48a2 88 807:          dey
48a3 c0a0 808:          cpy  #$a0
48a5 b0ee 809:          bcs  .a
48a7 adb148 810:          lda  .data
48aa acb248 811:          ldy  .data+1
48ad aeb348 812:          ldx  .data+2
48b0 60 813:          rts
48b1 000000 814: .data:      dc.b  $00,$00,$00
48b4      815: ;
48b4      816: ;
48b4      817: ;-----
48b4      818: ;this make's inveres charter flash
48b4      819: ;      FLASH MACRO
48b4      820: ;-----
48b4      821: ;
48b4      822: ;
48b4 48 823: flash:      pha
48b5 48 824:          pha
48b6      825: flash_1:
48b6      826: ; lda  #$00
48b6      827: ; sta  $41
48b6 eef848 828: flash_loop4: inc  flash_data
48b9 a90a 829:          lda  #$0a
48bb cdf848 830:          cmp  flash_data
48be f026 831:          beq  flash_loop
48c0      832: ;
48c0 a900 833:          lda  #$00
48c2 8514 834:          sta  $14
48c4      835: flash_loop3
48c4 ea 836:          nop
48c5 a901 837:          lda  #$01
48c7 8df302 838:          sta  $02f3
48ca a514 839:          lda  $14
48cc cdf848 840:          cmp  flash_data

```

```

48cf f002 841:      beq  flash_lppp
48d1 d0f1 842:      bne  flash_loop3
48d3 a900 843: flash_lppp:    lda  #$00
48d5 8514 844:      sta  $14
48d7 ea    845: flash_loop5:    nop
48d8 a902 846:      lda  #$02
48da 8df302 847:      sta  $02f3
48dd a514 848:      lda  $14
48df cdf948 849:      cmp  flash_data+1
48e2 f0d2 850:      beq  flash_loop4
48e4 d0f1 851:      bne  flash_loop5
48e6      852: flash_loop
48e6      853: ; lda #$00
48e6      854: ; sta  $41
48e6 a900 855:      lda  #$00
48e8 8df002 856:      sta  $02f0
48eb 68    857:      pla
48ec 68    858:      pla
48ed a905 859:      lda  #$05
48ef 8df848 860:      sta  flash_data
48f2 a90a 861:      lda  #$0a
48f4 8df948 862:      sta  flash_data+1
48f7 60    863:      rts
48f8 050a 864: flash_data:      dc.b  $05,$0a
48fa      865: ;
48fa      866: ;-----
48fa      867: ;prints Routine macro
48fa      868: ;this is a slow print to the screen
48fa      869: ;usage jsr  prints
48fa      870: ; .byte $9b," string to be printed",$ff
48fa      871: ;-----
48fa      872: ;
48fa 68    873: prints:      pla
48fb 8d0b49 874:      sta  pstr+1
48fe 68    875:      pla
48ff 8d0c49 876:      sta  pstr+2
4902 ee0b49 877: prsl:      inc  pstr+1
4905 d003 878:      bne  pstr
4907 ee0c49 879:      inc  pstr+2
490a adffff 880: pstr:      lda  $ffff
490d c9ff 881:      cmp  #$ff
490f f006 882:      beq  estri
4911 205049 883:      jsr  fast_output
4914 4c0249 884:      jmp  prsl
4917 ad0c49 885: estri:     lda  pstr+2
491a 48    886:      pha
491b ad0b49 887:      lda  pstr+1
491e 48    888:      pha
491f 60    889:      rts
4920      890: ;
4920      891: ;
4920      892: ;
4920      893: ;

```

```

4920      894: ;-----
4920      895: ;actual screen handler
4920      896: ;screen offset definitions
4920      897: ;-----
4920 0000  898: scr_offset:      dc.w  0
4922 2800  899:                dc.w  40
4924 5000  900:                dc.w  80
4926 7800  901:                dc.w 120
4928 a000  902:                dc.w 160
492a c800  903:                dc.w 200
492c f000  904:                dc.w 240
492e 1801  905:                dc.w 280
4930 4001  906:                dc.w 320
4932 6801  907:                dc.w 360
4934 9001  908:                dc.w 400
4936 b801  909:                dc.w 440
4938 e001  910:                dc.w 480
493a 0802  911:                dc.w 520
493c 3002  912:                dc.w 560
493e 5802  913:                dc.w 600
4940 8002  914:                dc.w 640
4942 a802  915:                dc.w 680
4944 d002  916:                dc.w 720
4946 f802  917:                dc.w 760
4948 2003  918:                dc.w 800
494a 4803  919:                dc.w 840
494c 7003  920:                dc.w 880
494e 9803  921:                dc.w 920
4950      922: ;
4950      923: echo:
4950      924: fast_output:
4950 855a  925:                sta  $5a
4952 ad0407 926:                lda  sc.redirect
4955 c902  927:                cmp  #print_p
4957 f01f  928:                beq  .go_cio
4959      929: .leo:
4959 adff02 930:                lda  $02ff      ; CONTROL 1
495c c900  931:                cmp  #$00
495e d0f9  932:                bne  .leo
4960      933:
4960 a55a  934:                lda  $5a
4962 c920  935:                cmp  #32
4964 9004  936:                bcc  .do_look
4966 c97d  937:                cmp  #125
4968 9013  938:                bcc  .do_here
496a a000  939: .do_look:      ldy  #0
496c b9734a 940: .lookup:      lda  .char_table,y
496f f00c  941:                beq  .do_here
4971 c55a  942:                cmp  $5a
4973 f003  943:                beq  .go_cio
4975 c8  944:                iny
4976 d0f4  945:                bne  .lookup
4978      946:

```

```

4978      947:
4978 a55a 948: .go_cio:      lda  $5a
497a 4c834a 949:      jmp  putlocal
497d      950:
497d      951:
497d a000 952: .do_here:      ldy  #0
497f a55d 953:      lda  $5d
4981 915e 954:      sta  ($5e),y
4983 a55a 955:      lda  $5a
4985 c99b 956:      cmp  #$9b
4987 f044 957:      beq  .docr
4989 20524a 958: .notcr:      jsr  .get_adr
498c a55a 959:      lda  $5a
498e 297f 960:      and  #$7f
4990 c920 961:      cmp  #32
4992 9007 962:      bcc  .add64
4994 c960 963:      cmp  #96
4996 9009 964:      bcc  .sub32
4998 4ca449 965:      jmp  .asis
499b 18 966: .add64:      clc
499c 6940 967:      adc  #64
499e 4ca449 968:      jmp  .asis
49a1 38 969: .sub32:      sec
49a2 e920 970:      sbc  #32
49a4 245a 971: .asis:      bit  $5a
49a6 1002 972:      bpl  .xxlate
49a8 0980 973:      ora  #$80
49aa a000 974: .xxlate:      ldy  #0
49ac 915e 975:      sta  ($5e),y
49ae e655 976:      inc  $55
49b0 e663 977:      inc  $63
49b2 a555 978:      lda  $55
49b4 c928 979:      cmp  #40
49b6 b019 980:      bcs  .next_row
49b8 c8 981:      iny
49b9 b15e 982:      lda  ($5e),y
49bb 855d 983:      sta  $5d
49bd 0980 984:      ora  #$80
49bf aef002 985:      ldx  752
49c2 d002 986:      bne  .nocurs1
49c4 915e 987:      sta  ($5e),y
49c6 e65e 988: .nocurs1:      inc  $5e
49c8 d002 989:      bne  .nooav
49ca e65f 990:      inc  $5e+1
49cc 60 991: .nooav:      rts
49cd a900 992: .docr:      lda  #0
49cf 8563 993:      sta  $63
49d1 a552 994: .next_row:      lda  $52
49d3 8555 995:      sta  $55
49d5 e654 996:      inc  $54
49d7 a454 997:      ldy  $54
49d9 c018 998:      cpy  #24
49db b013 999:      bcs  .scroll

```

```

49dd 20524a 1000:      jsr  .get_adr
49e0 a000 1001:      ldy  #0
49e2 b15e 1002:      lda  ($5e),y
49e4 855d 1003:      sta  $5d
49e6 aef002 1004:     ldx  752
49e9 d004 1005:     bne  .nocurs2
49eb 0980 1006:     ora  #$80
49ed 915e 1007:     sta  ($5e),y
49ef 60 1008: .nocurs2: rts
49f0      1009:
49f0 c654 1010: .scroll:   dec  $54
49f2 a558 1011:      lda  $58
49f4 8568 1012:      sta  $68
49f6 18 1013:      clc
49f7 6928 1014:      adc  #40
49f9 855e 1015:      sta  $5e
49fb a559 1016:      lda  $58+1
49fd 8569 1017:      sta  $68+1
49ff 6900 1018:      adc  #0
4a01 855f 1019:      sta  $5e+1
4a03 a000 1020:      ldy  #0
4a05 a203 1021:      ldx  #3
4a07 b15e 1022: .sloop:   lda  ($5e),y
4a09 9168 1023:      sta  ($68),y
4a0b c8 1024:      iny
4a0c d0f9 1025:      bne  .sloop
4a0e e65f 1026:      inc  $5e+1
4a10 e669 1027:      inc  $68+1
4a12 ca 1028:      dex
4a13 d0f2 1029:      bne  .sloop
4a15 b15e 1030: .sloop2:  lda  ($5e),y
4a17 9168 1031:      sta  ($68),y
4a19 c8 1032:      iny
4a1a c098 1033:      cpy  #152
4a1c 90f7 1034:      bcc  .sloop2
4a1e a558 1035:      lda  $58
4a20 18 1036:      clc
4a21 6998 1037:      adc  #920&$ff
4a23 855e 1038:      sta  $5e
4a25 a559 1039:      lda  $58+1
4a27 6903 1040:      adc  #920/256
4a29 855f 1041:      sta  $5e+1
4a2b a000 1042:      ldy  #0
4a2d 98 1043:      tya
4a2e 915e 1044: .clear:   sta  ($5e),y
4a30 c8 1045:      iny
4a31 c028 1046:      cpy  #40
4a33 90f9 1047:      bcc  .clear
4a35 aef002 1048:     ldx  752
4a38 d006 1049:     bne  .nocurs3
4a3a a980 1050:     lda  #$80
4a3c a455 1051:     ldy  $55
4a3e 915e 1052:     sta  ($5e),y

```

```

4a40 a900 1053: .nocurs3:      lda  #0
4a42 855d 1054:                sta  $5d
4a44 a55e 1055:                lda  $5e
4a46 18   1056:      clc
4a47 6555 1057:                adc  $55
4a49 855e 1058:                sta  $5e
4a4b a55f 1059:                lda  $5e+1
4a4d 6900 1060:                adc  #0
4a4f 855f 1061:                sta  $5e+1
4a51 60   1062:      rts
4a52 a554 1063: .get_adr:      lda  $54
4a54 0a   1064:      asl
4a55 a8   1065:      tay
4a56 a558 1066:                lda  $58
4a58 18   1067:      clc
4a59 792049 1068:                adc  scr_offset,y
4a5c 855e 1069:                sta  $5e
4a5e a559 1070:                lda  $58+1
4a60 792149 1071:                adc  scr_offset+1,y
4a63 855f 1072:                sta  $5e+1
4a65 a55e 1073:                lda  $5e
4a67 18   1074:      clc
4a68 6555 1075:                adc  $55
4a6a 855e 1076:                sta  $5e
4a6c a55f 1077:                lda  $5e+1
4a6e 6900 1078:                adc  #0
4a70 855f 1079:                sta  $5e+1
4a72 60   1080:      rts
4a73      1081: .char_table:
4a73 1b1c1d 1082:                dc.b  27,28,29,30,31,125,126,127
4a7b 9c9d9e 1083:                dc.b  156,157,158,159,253,254,255
4a83      1084:
4a83      1085:
4a83 a20b 1086: putlocal:      ldx  #11
4a85 8e4203 1087:                stx  $0342
4a88 a200 1088:                ldx  #0
4a8a 8e4803 1089:                stx  $0348
4a8d 8e4903 1090:                stx  $0349
4a90 4c56e4 1091:      jmp  $e456
4a93      1092: ;
4a93      1093: ;-----
4a93      1094: ; GET KEY ROUTEEN
4a93      1095: ;this routeen get's one key and
4a93      1096: ;return it to the accumulator
4a93      1097: ;-----
4a93      1098: ;
4a93 86b6 1099: getkey:                stx  $b6
4a95 84b7 1100:                sty  $b7
4a97 209f4a 1101:      jsr  .a
4a9a a6b6 1102:                ldx  $b6
4a9c a4b7 1103:                ldy  $b7
4a9e 60   1104:      rts
4a9f ad25e4 1105: .a:                lda  $e420+5

```

```

4aa2 48 1106:          pha
4aa3 ad24e4 1107:        lda  $e420+4
4aa6 48 1108:          pha
4aa7 60 1109:          rts
4aa8      1110: ;
4aa8      1111:
4aa8      1112: ;-----
4aa8      1113: ; help menu section
4aa8      1114: ;-----
4aa8      1115: help_menu
4aa8      1116:
4aa8      1117:
4aa8      1118:
4aa8 a97d 1119:          lda  #$7d
4aaa 205049 1120:         jsr  echo
4aad 20a442 1121:         jsr  begin_port-3
4ab0      1122:
4ab0 20fa48 1123:         jsr  printsi
4ab3      1124: ;                11111111112222222
4ab3      1125: ;                12345678901234567890123456
4ab3 446973 1126:         dc.b  "Display Joystick Port resu
4adb 686172 1127:         dc.b  "hardware box. Parts requir
4b01 203120 1128:         dc.b  " 1 220 ohm Resister!",$9b
4b17 203120 1129:         dc.b  " 1 Project Box large enou
4b3e 203820 1130:         dc.b  " 8 LED's Color of you cho
4b5d 203130 1131:         dc.b  " 10 Momentary Pushbuttons
4b80 203120 1132:         dc.b  " 1 9 pin Joystick Extensi
4ba4 9b9bff 1133:         dc.b  $9b,$9b,-1
4ba7 206c54 1134:         jsr  pause
4baa      1135:
4baa 20fa48 1136:         jsr  printsi
4bad 7d 1137:          dc.b  $7d
4bae 9b 1138:          dc.b  $9b
4baf      1139: ;                11111111112222222
4baf      1140: ;                12345678901234567890123456
4baf      1141:
4baf 50696e 1142:         dc.b  "Pin 7 = +5---////-----
4bd6      1143: ;          dc.b  "
4bd6 2b3320 1144:         dc.b  "+3 N/C --+--+--+--+--+--+
4bfd 202020 1145:         dc.b  "      ||| | || |
4c21 4c2050 1146:         dc.b  "L Pin 8 --)-)-)-)-)-)-)-
4c45 202020 1147:         dc.b  "      ||| | || |
4c69 4c2050 1148:         dc.b  "L Pin 1 --)-)-)-)-)-)-)-
4c8d 202020 1149:         dc.b  "      ||| | || |
4cb1 4c2050 1150:         dc.b  "L Pin 2 --)-)-)-)-)-)-O-
4cd5 202020 1151:         dc.b  "      ||| | || |
4cf9 4c2050 1152:         dc.b  "L Pin 3 --)-)-)-)-)-O----
4d1d 202020 1153:         dc.b  "      ||| | || |
4d41 4c2050 1154:         dc.b  "L Pin 4 --)-)-)-)-)-O-----
4d65 202020 1155:         dc.b  "      ||| | || |
4d89 4c2050 1156:         dc.b  "L Pin 6 --)-)-)-)-)-O-----
4dad 202020 1157:         dc.b  "      ||| | || |
4dd1 522050 1158:         dc.b  "R Pin 1 --)-)-)-)-)-O-----

```

```

4df5 202020 1159:      dc.b  "    |||
4e19 522050 1160:      dc.b  "R Pin 2 --)-O-----
4e3d 202020 1161:      dc.b  "    ||
4e61 522050 1162:      dc.b  "R Pin 3 --)-O-----
4e85 202020 1163:      dc.b  "    |
4ea9 522050 1164:      dc.b  "R Pin 4 --O-----
4ecd 202020 1165:      dc.b  "
4ef1 522050 1166:      dc.b  "R Pin 6 -----
4f15      1167:
4f15 208f54 1168:      jsr   pause1
4f18 20fa48 1169:      jsr   printsi
4f1b 7d     1170:      dc.b  $7d
4f1c 9b     1171:      dc.b  $9b
4f1d      1172:
4f1d 417461 1173:      dc.b  "Atari Joystick Port Pins a
4f44 202020 1174:      dc.b  "  Left Port:    Rig
4f67 3d3d3d 1175:      dc.b  "=====
4f8e 5f5f5f 1176:      dc.b  "
4fb5 5c2020 1177:      dc.b  " 1 2 3 4 5 / 1
4fdc 205c20 1178:      dc.b  "    /
5003 20205c 1179:      dc.b  " 6 7 8 9 / 6
502a 202020 1180:      dc.b  "    /
5051 202020 1181:      dc.b  "  -----  ---
5078 3d3d3d 1182:      dc.b  "=====
509f 50696e 1183:      dc.b  "Pin #1 = (joystick) forwar
50c1 32203d 1184:      dc.b  "2 = (joystick) back input"
50db 33203d 1185:      dc.b  "3 = (joystick) left input"
50f5 34203d 1186:      dc.b  "4 = (joystick) right input
5110 35203d 1187:      dc.b  "5 = "
5114 22     1188:      dc.b  $22
5115 42     1189:      dc.b  "B"
5116 22     1190:      dc.b  $22
5117 20706f 1191:      dc.b  " potentiometer input",$9b
512c 36203d 1192:      dc.b  "6 = trigger input",$9b
513e 37203d 1193:      dc.b  "7 = +5 volts",$9b
514b 38203d 1194:      dc.b  "8 = ground",$9b
5156 39203d 1195:      dc.b  "9 = "
515a 22     1196:      dc.b  $22
515b 41     1197:      dc.b  "A"
515c 22     1198:      dc.b  $22
515d 20706f 1199:      dc.b  " potentiometer input",$9b
5172 9b     1200:      dc.b  $9b
5173 ff     1201:      dc.b  -1
5174      1202:
5174      1203:
5174 206c54 1204:      jsr   pause
5177 20fa48 1205:      jsr   printsi
517a      1206:
517a 205363 1207:      dc.b  " Schematics Ledge",$9b
518c 202020 1208:      dc.b  "
51b3 4d6f6d 1209:      dc.b  "Momentary Pushbutton Switc
51da 202020 1210:      dc.b  "
5201 4c6564 1211:      dc.b  "Led  -->|--

```



```

5228 202020 1212:      dc.b  "
524f 526573 1213:      dc.b  "Resister ---////---- 2
5276 9b9bff 1214:      dc.b  $9b,$9b,-1
5279      1215:
5279 206c54 1216:      jsr  pause
527c 20fa48 1217:      jsr  printsi
527f      1218:
527f 9b2053 1219:      dc.b  $9b," Should a company wish
52a3 204920 1220:      dc.b  " I will provide a site lic
52c8 20636f 1221:      dc.b  " cost $100.00 US Dollars."
52e2 204174 1222:      dc.b  " Atari user or usergroups"
52fc 206d61 1223:      dc.b  " may use it FREE of charge
5321 204242 1224:      dc.b  " BBS Systems",$9b,-1
532f 206c54 1225:      jsr  pause
5332 20fa48 1226:      jsr  printsi
5335 9b9b 1227:      dc.b  $9b,$9b
5337 206966 1228:      dc.b  " if you wish to have the s
5357 20636f 1229:      dc.b  " code then contact:",$9B,$
536c 202020 1230:      dc.b  "      Stephen J. Carden",
5385 202020 1231:      dc.b  "      3321 Jonathan Cir."
539f 202020 1232:      dc.b  "      Augusta, Ga. 30906"
53b9 202020 1233:      dc.b  "      sjcarden@bellsouth.
53d8      1234: ;
53d8 206f72 1235:      dc.b  " or send a donation of $20
53fb 746865 1236:      dc.b  "the above address.",$9B,$F
540f 20fa48 1237:      jsr  printsi
5412 9b9b 1238:      dc.b  $9b,$9b
5414 205072 1239:      dc.b  " Press "
541b dbd2e5 1240:      dc.b  <+128>,"[Return]"
5423 20666f 1241:      dc.b  " for Dos, any other key ",
543c 20746f 1242:      dc.b  " to continue with this fil
5459 20934a 1243:      jsr  getkey
545c c99b 1244:      cmp  #$9b
545e d001 1245:      bne  help2
5460 60 1246:      rts
5461      1247:
5461      1248: help2:
5461 a97d 1249:      lda  #$7d
5463 205049 1250:      jsr  echo
5466 20a442 1251:      jsr  begin_port-3
5469 4cb742 1252:      jmp  top
546c      1253: ;
546c      1254: pause:      ;          a more command
546c 20fa48 1255:      jsr  printsi
546f dba0ad 1256:      dc.b  <+128>,"[ -More- ]"
5479 ff 1257:      dc.b  -1
547a 20934a 1258:      jsr  getkey
547d 20fa48 1259:      jsr  printsi
5480 7e7e7e 1260:      dc.b  "~~~~~",$FF
548e 60 1261:      rts
548f      1262: pause1:      ;          a more command
548f 20fa48 1263:      jsr  printsi
5492 dbccdd 1264:      dc.b  <+128>,"[M]"

```

```
5495 ff 1265:          dc.b  -1
5496 20934a 1266:      jsr  getkey
5499 20fa48 1267:      jsr  printsi
549c 7e7e7e 1268:      dc.b  "~~~",$FF
54a0 60 1269:          rts
54a1 1270:
54a1 1271:
54a1 1272:
54a1 1273: win_end:      ds.b  0
54a1 1274: ;
54a1 1275: ;
54a1 1276: ;
54a1 1277:
54a1 1278:
54a1 1279:
54a1 1280:
54a1 1281:
```

End assembly: no errors