


```

0000 1: ;-----
0000 2: ;      Copyright 2012 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: ;      SDrive.com
0000 9: ;
0000 10: ;This is a program is for testing a detection routeen
0000 11: ;      init 9/10/2012   sjc new design
0000 12: ;
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: com_file_name:      .macro
0000 25: ;      dc.b 13,"filename.ext",$9b
0000 26: ;      dc.b 12,"SDrive.com ",$9b
0000 27: ;      .endm
0000 28:
0000 29: ;
0000 30: ;~~~~~
0000 31: ; File revision history. Version Number in hex
0000 32: ;
0000 33: file_ver:      equ $10
0000 34: ;~~~~~
0000 35: ; the Month in dec for the first time this revision
0000 36: ;was compiled
0000 37: ;
0000 38: c_month:      equ 9
0000 39: ;~~~~~
0000 40: ; the day in dec for the first time this revision
0000 41: ;was compiled
0000 42: c_day:      equ 19
0000 43: ;~~~~~
0000 44: ; the year in dec for the first time this revision
0000 45: ;was compiled
0000 46: c_year:      equ 2012
0000 47: ;~~~~~
0000 48: ; Type of compiler used in program
0000 49: ;
0000 50: ; 0 = unknown Compiler
0000 51: ; 1 = xasm
0000 52: ; 2 = mac_65
0000 53: ; 3 = basic

```

```

0000      54: ; 4 = compiled basic xl
0000      55: ; 5 = C65
0000      56: ; 6 = Action
0000      57: ;
0000      58: ; We can add more as time goes on!
0000      59: ;
0000      60: ;
0000      61: ;
0000      62: xasm:          equ    1
0000      63: mac_65:         equ    2
0000      64: basic:       equ    3
0000      65: basicxl:     equ    4
0000      66: c65:         equ    5
0000      67: action:      equ    6
0000      68: ;
0000      69: file_compiler:    equ    xasm
0000      70: ;~~~~~
0000      71: ; is this relocatable code ?
0000      72: ; anyother value other than 1 or 2 would be unknown
0000      73: r..yes:           equ    1
0000      74: r..no:           equ    2
0000      75: ;
0000      76: ;~~~~~
0000      77: ; Gotta define if it can be relocated
0000      78: l_relocatable:    equ    r..no
0000      79: ;
0000      80: ;
0000      81: ;~~~~~
0000      82: ; Gotta define if it can be relocated
0000      83: r..crl:          equ    $7d
0000      84: r..crlf:         equ    $9b
0000      85: r..space:        equ    $20
0000      86: r..bypassdos:    equ    $15 ; bypass dos check
0000      87: r..bpasintro:    equ    $ff
0000      88:
0000      89:
0000      90: l_frstscreenbyte: equ    r..crl
0000      91: ;~~~~~
0000      92: ; The language the output file is in.
0000      93: ;
0000      94: ;
0000      95: ; 0 =    Undefined
0000      96: ; 1 =    English
0000      97: ; 2 =    German
0000      98: ;
0000      99: r..language:      equ    1
0000     100: ;~~~~~
0000     101: siov_read:        equ    $40 ; Read
0000     102: siov_write:       equ    $80 ; write
0000     103: ;-----
0000     104: ;
0000     105: sdr_disk1:        equ    1 ; Emulated
0000     106: sdr_disk2:        equ    2 ; Emulated

```

```

0000    107: sdr_disk3:      equ    3          ; Emulated
0000    108: sdr_disk0:      equ    0          ; Emulated
0000    109: ;
0000    110: sdr_timeout:          equ    7          ; sio timeo
0000    111: xfersize:              equ    35          ; Buffer si
0000    112: ;
0000    113: sdr_inuse:             equ    sdr_disk0    ; USB devic
0000    114: siov_do:               equ    siov_read
0000    115: sdr_disk:              equ    sdr_disk0    ;
0000    116:
0000    117:
0000    118:                      .include    equates
0000    119:                      .include    globals
0000    120:                      .include    macros
0000    121:
4000    122:                      .org    $4000
4000    123:
4000    124: win_start:
4000    125: header_info:
4000    126:                      .include    header
42a7    127:
42a7    128: ;
42a7    129: Start:
42a7    130:
42a7 a50a 131:          LDA    COMTAB ; calc address of S
42a9 38 132:          SEC
42aa e90a 133:          SBC    # low lsio
42ac 8d1944 134:          STA    XSIO+1
42af a50b 135:          LDA    COMTAB+1
42b1 e900 136:          SBC    # high lsio
42b3 8d1a44 137:          STA    XSIO+2
42b6 a96c 138:          lda    #$6c
42b8 8d1844 139:          sta    xsio
42bb 140: SDrive:
42bb 141:
42bb a971 142:          lda    #$71
42bd 8d0003 143:          sta    $300          ; D
42c0 a901 144:          lda    #$01
42c2 8d0103 145:          sta    $301          ; D
42c5 146:
42c5 a9e0 147:          lda    #$e0
42c7 8d0203 148:          sta    $0302          ; s
42ca a940 149:          lda    #siov_read
42cc 8d0303 150:          sta    $0303
42cf a918 151:          lda    # low sdrive_data
42d1 8d0403 152:          sta    $0304          ; D
42d4 a943 153:          lda    # high sdrive_data
42d6 8d0503 154:          sta    $0305          ; D
42d9 155:
42d9 a907 156:          lda    #SDr_timeout
42db 8d0603 157:          sta    $0306          ; T
42de a900 158:          lda    #$00
42e0 8d0703 159:          sta    $0307          ; D

```

```

42e3      160:
42e3 a926  161:      lda    #low 38
42e5 8d0803 162:      sta    $0308      ; N
42e8 a900  163:      lda    #high 38
42ea 8d0903 164:      sta    $0309      ; N
42ed a926  165:      lda    #low 38
42ef 8d0a03 166:      sta    $030a      ; D
42f2 a900  167:      lda    #high 38
42f4 8d0b03 168:      sta    $030b      ; D
42f7      169:
42f7 201844 170:      jsr    xsio
42fa ad0303 171:      lda    $303      ; D
42fd c901  172:      cmp    #$01
42ff f001  173:      beq    .a
4301 60    174:      rts
4302      175:
4302 201b44 176: .a:      jsr    printsi
4305 534472 177:      dc.b   "SDrive Detected!",$9b
4316 ff    178:      dc.b   -1
4317 60    179:      rts
4318      180:
4318      181:
4318      182:
4318      183:
4318      184: sdrive_data:
4318      185:      ds.b   256
4418      186:
4418      187:
4418      188:
4418 4cffff 189: XSIO:      jmp    $ffff
441b      190:
441b      191:
441b      192: ;-----
441b      193: ;prints Routine macro
441b      194: ;this is a slow print to the screen
441b      195: ;usage jsr    printsi
441b      196: ; .byte $9b," string to be printed",$ff
441b      197: ;-----
441b      198: ;
441b      199: printsi:
441b 68    200:      pla
441c 8d2c44 201:      sta    .pstr+1
441f 68    202:      pla
4420 8d2d44 203:      sta    .pstr+2
4423 ee2c44 204: .prsl:      inc    .pstr+1
4426 d003  205:      bne    .pstr
4428 ee2d44 206:      inc    .pstr+2
442b adffff 207: .pstr:      lda    $ffff
442e c9ff  208:      cmp    #$ff
4430 f006  209:      beq    .estri
4432 207144 210:      jsr    fast_output
4435 4c2344 211:      jmp    .prsl
4438 ad2d44 212: .estri:      lda    .pstr+2

```

```

443b 48 213:          pha
443c ad2c44 214:        lda  .pstr+1
443f 48 215:          pha
4440 60 216:          rts
4441      217: ;
4441      218: ;
4441      219:
4441      220: ;
4441      221: ;-----
4441      222: ;actual screen handler
4441      223: ;screen offset definitions
4441      224: ;-----
4441      225: scr_offset:
4441 0000 226:          dc.w 0
4443 2800 227:          dc.w 40
4445 5000 228:          dc.w 80
4447 7800 229:          dc.w 120
4449 a000 230:          dc.w 160
444b c800 231:          dc.w 200
444d f000 232:          dc.w 240
444f 1801 233:          dc.w 280
4451 4001 234:          dc.w 320
4453 6801 235:          dc.w 360
4455 9001 236:          dc.w 400
4457 b801 237:          dc.w 440
4459 e001 238:          dc.w 480
445b 0802 239:          dc.w 520
445d 3002 240:          dc.w 560
445f 5802 241:          dc.w 600
4461 8002 242:          dc.w 640
4463 a802 243:          dc.w 680
4465 d002 244:          dc.w 720
4467 f802 245:          dc.w 760
4469 2003 246:          dc.w 800
446b 4803 247:          dc.w 840
446d 7003 248:          dc.w 880
446f 9803 249:          dc.w 920
4471      250: ;
4471      251: fast_output:
4471 855a 252:          sta  $5a
4473 ad0407 253:        lda  sc.redirect
4476 c902 254:          cmp  #print_p
4478 f01f 255:          beq  .go_cio
447a      256: .leo:
447a adff02 257:        lda  $02ff      ; CONTROL 1
447d c900 258:          cmp  #$00
447f d0f9 259:          bne  .leo
4481      260:
4481 a55a 261:          lda  $5a
4483 c920 262:          cmp  #32
4485 9004 263:          bcc  .do_look
4487 c97d 264:          cmp  #125
4489 9013 265:          bcc  .do_here

```

```

448b a000 266: .do_look:      ldy  #0
448d b99445 267: .lookup:      lda  .char_table,y
4490 f00c 268:              beq  .do_here
4492 c55a 269:              cmp  $5a
4494 f003 270:              beq  .go_cio
4496 c8 271:              iny
4497 d0f4 272:              bne  .lookup
4499      273:
4499      274:
4499 a55a 275: .go_cio:      lda  $5a
449b 4ca445 276:              jmp  putlocal
449e      277:
449e      278:
449e a000 279: .do_here:      ldy  #0
44a0 a55d 280:              lda  $5d
44a2 915e 281:              sta  ($5e),y
44a4 a55a 282:              lda  $5a
44a6 c99b 283:              cmp  #$9b
44a8 f044 284:              beq  .docr
44aa 207345 285: .notcr:      jsr  .get_adr
44ad a55a 286:              lda  $5a
44af 297f 287:              and  #$7f
44b1 c920 288:              cmp  #32
44b3 9007 289:              bcc  .add64
44b5 c960 290:              cmp  #96
44b7 9009 291:              bcc  .sub32
44b9 4cc544 292:              jmp  .asis
44bc 18 293: .add64:      clc
44bd 6940 294:              adc  #64
44bf 4cc544 295:              jmp  .asis
44c2 38 296: .sub32:      sec
44c3 e920 297:              sbc  #32
44c5 245a 298: .asis:      bit  $5a
44c7 1002 299:              bpl  .xxlate
44c9 0980 300:              ora  #$80
44cb a000 301: .xxlate:      ldy  #0
44cd 915e 302:              sta  ($5e),y
44cf e655 303:              inc  $55
44d1 e663 304:              inc  $63
44d3 a555 305:              lda  $55
44d5 c928 306:              cmp  #40
44d7 b019 307:              bcs  .next_row
44d9 c8 308:              iny
44da b15e 309:              lda  ($5e),y
44dc 855d 310:              sta  $5d
44de 0980 311:              ora  #$80
44e0 aef002 312:              ldx  752
44e3 d002 313:              bne  .nocursl
44e5 915e 314:              sta  ($5e),y
44e7 e65e 315: .nocursl:      inc  $5e
44e9 d002 316:              bne  .nooav
44eb e65f 317:              inc  $5e+1
44ed 60 318: .nooav:      rts

```

```

44ee a900 319: .docr:      lda  #0
44f0 8563 320:      sta  $63
44f2 a552 321: .next_row:    lda  $52
44f4 8555 322:      sta  $55
44f6 e654 323:      inc  $54
44f8 a454 324:      ldy  $54
44fa c018 325:      cpy  #24
44fc b013 326:      bcs  .scroll
44fe 207345 327:      jsr  .get_adr
4501 a000 328:      ldy  #0
4503 b15e 329:      lda  ($5e),y
4505 855d 330:      sta  $5d
4507 aef002 331:      ldx  752
450a d004 332:      bne  .nocurs2
450c 0980 333:      ora  #$80
450e 915e 334:      sta  ($5e),y
4510 60 335: .nocurs2:    rts
4511 336:
4511 c654 337: .scroll:      dec  $54
4513 a558 338:      lda  $58
4515 8568 339:      sta  $68
4517 18 340:      clc
4518 6928 341:      adc  #40
451a 855e 342:      sta  $5e
451c a559 343:      lda  $58+1
451e 8569 344:      sta  $68+1
4520 6900 345:      adc  #0
4522 855f 346:      sta  $5e+1
4524 a000 347:      ldy  #0
4526 a203 348:      ldx  #3
4528 b15e 349: .scloop:      lda  ($5e),y
452a 9168 350:      sta  ($68),y
452c c8 351:      iny
452d d0f9 352:      bne  .scloop
452f e65f 353:      inc  $5e+1
4531 e669 354:      inc  $68+1
4533 ca 355:      dex
4534 d0f2 356:      bne  .scloop
4536 b15e 357: .scloop2:     lda  ($5e),y
4538 9168 358:      sta  ($68),y
453a c8 359:      iny
453b c098 360:      cpy  #152
453d 90f7 361:      bcc  .scloop2
453f a558 362:      lda  $58
4541 18 363:      clc
4542 6998 364:      adc  #920&$ff
4544 855e 365:      sta  $5e
4546 a559 366:      lda  $58+1
4548 6903 367:      adc  #920/256
454a 855f 368:      sta  $5e+1
454c a000 369:      ldy  #0
454e 98 370:      tya
454f 915e 371: .clear:      sta  ($5e),y

```



```

4551 c8 372:      iny
4552 c028 373:    cpy #40
4554 90f9 374:    bcc .clear
4556 aef002 375:   ldx 752
4559 d006 376:    bne .nocurs3
455b a980 377:    lda #$80
455d a455 378:    ldy $55
455f 915e 379:    sta ($5e),y
4561 a900 380: .nocurs3:   lda #0
4563 855d 381:    sta $5d
4565 a55e 382:    lda $5e
4567 18 383:     clc
4568 6555 384:    adc $55
456a 855e 385:    sta $5e
456c a55f 386:    lda $5e+1
456e 6900 387:    adc #0
4570 855f 388:    sta $5e+1
4572 60 389:     rts
4573 a554 390: .get_adr:   lda $54
4575 0a 391:     asl
4576 a8 392:     tay
4577 a558 393:    lda $58
4579 18 394:     clc
457a 794144 395:   adc scr_offset,y
457d 855e 396:    sta $5e
457f a559 397:    lda $58+1
4581 794244 398:   adc scr_offset+1,y
4584 855f 399:    sta $5e+1
4586 a55e 400:    lda $5e
4588 18 401:     clc
4589 6555 402:    adc $55
458b 855e 403:    sta $5e
458d a55f 404:    lda $5e+1
458f 6900 405:    adc #0
4591 855f 406:    sta $5e+1
4593 60 407:     rts
4594 408: .char_table:
4594 1b1c1d 409:   dc.b 27,28,29,30,31,125,126,127
459c 9c9d9e 410:   dc.b 156,157,158,159,253,254,255
45a4 411:
45a4 412:
45a4 413: putlocal:
45a4 a20b 414:    ldx #11
45a6 8e4203 415:   stx $0342
45a9 a200 416:    ldx #0
45ab 8e4803 417:   stx $0348
45ae 8e4903 418:   stx $0349
45b1 4c56e4 419:   jmp $e456
45b4 420: ;
45b4 421:
45b4 422:
45b4 423:
45b4 424:

```

```
45b4      425:
45b4      426:
45b4      427:
45b4      428:
45b4      429: win_end:      ds.b  0
45b4      430: ;
45b4      431: ;
```

End assembly: no errors