


```

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: ;      SIO2SD.com
0000 9: ;
0000 10: ;This is a program that sends two text strings to the disp
0000 11: ;      init  9/09/2012      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,"Sio2sd.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  10
0000 39: ;~~~~~
0000 40: ; the day in dec for the first time this revision
0000 41: ;was compiled
0000 42: c_day:      equ  12
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:
0000     102:
0000     103:          .include    equates
0000     104:          .include    globals
0000     105:          .include    macros
0000     106:

```

```

4000      107:                .org $4000
4000      108:
4000      109: win_start:
4000      110: header_info:
4000      111:                .include      header
42a7      112:
42a7      113: ;
42a7      114: Start:
42a7      115:
42a7      116: get_sector_count:
42a7 a50a  117:                LDA    COMTAB ; calc address of S
42a9 38    118:                SEC
42aa e90a  119:                SBC    # low lsio
42ac 8d6e43 120:                STA    XSIO+1
42af a50b  121:                LDA    COMTAB+1
42b1 e900  122:                SBC    # high lsio
42b3 8d6f43 123:                STA    XSIO+2
42b6 a96c  124:                lda    #$6c
42b8 8d6d43 125:                sta    xsio
42bb      126:
42bb a970  127:                lda    #$70
42bd 8d0003 128:                sta    $300
42c0 a903  129:                lda    #$03
42c2 8d0103 130:                sta    $301
42c5 a910  131:                lda    #$10
42c7 8d0203 132:                sta    $0302
42ca a980  133:                lda    #$80
42cc 8d0303 134:                sta    $0303
42cf a92f  135:                lda    # low real_data
42d1 8d0403 136:                sta    $0304
42d4 a944  137:                lda    # high real_data
42d6 8d0503 138:                sta    $0305
42d9 a928  139:                lda    #40
42db 8d0803 140:                sta    $0308
42de a900  141:                lda    #$00
42e0 8d0903 142:                sta    $0309
42e3 a900  143:                lda    #0
42e5 8d0a03 144:                sta    $030a
42e8 a900  145:                lda    #$00
42ea 8d0b03 146:                sta    $030b
42ed 206d43 147:                jsr    xsio
42f0 ad0303 148:                lda    $303
42f3 c901  149:                cmp    #$01
42f5 d039  150:                bne    .error
42f7      151:
42f7 a970  152:                lda    #$70
42f9 8d0003 153:                sta    $300
42fc a903  154:                lda    #$03
42fe 8d0103 155:                sta    $301
4301 a910  156:                lda    #$10
4303 8d0203 157:                sta    $0302
4306 a980  158:                lda    #$80
4308 8d0303 159:                sta    $0303

```

```

430b a967 160:      lda  # low huh_data
430d 8d0403 161:      sta  $0304
4310 a944 162:      lda  # high huh_data
4312 8d0503 163:      sta  $0305
4315 a928 164:      lda  #40
4317 8d0803 165:      sta  $0308
431a a900 166:      lda  #$00
431c 8d0903 167:      sta  $0309
431f a901 168:      lda  #1
4321 8d0a03 169:      sta  $030a
4324 a900 170:      lda  #$00
4326 8d0b03 171:      sta  $030b
4329 206d43 172:      jsr  xsio
432c 4c5043 173:      jmp  good_message
432f 60 174:      rts
4330 175: .error:
4330 20c643 176:      jsr  printsi
4333 9b9b9b 177:      dc.b  $9b,$9b,$9b
4336 202053 178:      dc.b  " Sio2SD "
433f ceeff4 179:      dc.b  <+128>,"Not Present"
434a 9b 180:      dc.b  $9b
434b ff 181:      dc.b  -1
434c 207043 182:      jsr  driverror
434f 60 183:      rts
4350 184: good_message:
4350 20c643 185:      jsr  printsi
4353 9b9b9b 186:      dc.b  $9b,$9b,$9b
4356 202053 187:      dc.b  " Sio2SD "
435f d0f2e5 188:      dc.b  <+128>,"Present!"
4367 9b 189:      dc.b  $9b
4368 ff 190:      dc.b  -1
4369 207043 191:      jsr  driverror
436c 60 192:      rts
436d 193: ;
436d 194: ;
436d 4cffff 195: XSIO:      jMP  $ffff
4370 196:
4370 197: driverror:
4370 48 198:      pha
4371 20c643 199:      jsr  printsi
4374 9b2053 200:      dc.b  $9b," Siov Error # ",$FF
4384 68 201:      pla
4385 208e43 202:      jsr  pr_byte
4388 20c643 203:      jsr  printsi
438b 9bff 204:      dc.b  $9b,-1
438d 60 205:      rts
438e 206:
438e 207: ;
438e 208: pr_byte:
438e a200 209:      ldx  #$00
4390 210: pr_card:
4390 85d4 211:      sta  $d4
4392 86d5 212:      stx  $d4+1

```

```

4394 a900 213:      lda  #$00
4396 85f2 214:      sta  $f2
4398 20aad9 215:     jsr  $d9aa
439b 20e6d8 216:     jsr  $d8e6
439e a000 217:     ldy  #$00
43a0      218: .a:
43a0 b1f3 219:      lda  ($f3),y
43a2 08 220:      php
43a3 297f 221:      and  #$7f
43a5 c8 222:      iny
43a6 20ad43 223:     jsr  echo
43a9 28 224:      plp
43aa 10f4 225:      bpl  .a
43ac 60 226:      rts
43ad      227: ;
43ad      228: ;=====
43ad      229: ;   print/input routines
43ad      230: ;=====
43ad      231: ;
43ad      232: ;   Print character
43ad      233: ;   -----
43ad      234: ; in:
43ad      235: ;   A   = character to print
43ad      236: ; out:
43ad      237: ;   all registers preserved
43ad      238: ;
43ad 20f843 239: ECHO:  JSR SAVER
43b0 20b643 240:      JSR ZOUT
43b3 4c2844 241:      JMP RESALL
43b6      242: ;
43b6 8dc243 243: ZOUT:  STA ZTEMP+1
43b9 ad4703 244:      LDA $0347
43bc 48 245:      PHA
43bd ad4603 246:      LDA $0346
43c0 48 247:      PHA
43c1 a900 248: ZTEMP: LDA #0
43c3 a200 249:      LDX #0          ;1   ; force NO
43c5 60 250:      RTS
43c6      251: ;
43c6      252: ;
43c6      253: ;   Print text
43c6      254: ;   -----
43c6      255: ; out:
43c6      256: ;   all registers preserved
43c6      257: ; notes:
43c6      258: ;   This print routine will print all characters
43c6      259: ;   following the JSR PRINT until a delimiter of
43c6      260: ;   -1 ($FF) is reached. PRINT will return to the
43c6      261: ;   point one byte beyond the delimiter.
43c6      262: ;
43c6 20f843 263: printsi: JSR SAVER
43c9 ba 264:      TSX
43ca bd0501 265:      LDA $0105,X

```

```
43cd 8dd943 266: STA ZOCH+1
43d0 bd0601 267: LDA $0106,X
43d3 8dda43 268: STA ZOCH+2
43d6      269: ;
43d6 a001 270: LDY #1
43d8 b9ffff 271: ZOCH: LDA $FFFF,Y
43db c9ff 272: CMP #$FF
43dd f006 273: BEQ ZECHO
43df 20ad43 274: JSR ECHO
43e2 c8 275: INY
43e3 d0f3 276: BNE ZOCH
43e5 98 277: ZECHO: TYA
43e6 18 278: CLC
43e7 7d0501 279: ADC $0105,X
43ea 9d0501 280: STA $0105,X
43ed bd0601 281: LDA $0106,X
43f0 6900 282: ADC #0
43f2 9d0601 283: STA $0106,X
43f5 4c2844 284: JMP RESALL
43f8      285: ;
43f8      286: ;
43f8      287: ; SAVE & RESTORE registers
43f8      288: ; -----
43f8      289: ;
43f8 08 290: SAVER: PHP
43f9 48 291: PHA
43fa 48 292: PHA
43fb 48 293: PHA
43fc 08 294: PHP
43fd 48 295: PHA
43fe 8a 296: TXA
43ff 48 297: PHA
4400 ba 298: TSX
4401 bd0901 299: LDA $0109,X
4404 9d0501 300: STA $0105,X
4407 bd0701 301: LDA $0107,X
440a 9d0901 302: STA $0109,X
440d bd0101 303: LDA $0101,X
4410 9d0701 304: STA $0107,X
4413 bd0801 305: LDA $0108,X
4416 9d0401 306: STA $0104,X
4419 bd0601 307: LDA $0106,X
441c 9d0801 308: STA $0108,X
441f 98 309: TYA
4420 9d0601 310: STA $0106,X
4423 68 311: PLA
4424 aa 312: TAX
4425 68 313: PLA
4426 28 314: PLP
4427 60 315: RTS
4428      316: ;
4428 68 317: RESALL: PLA
4429 a8 318: TAY
```

```
442a 68 319: PLA
442b aa 320: TAX
442c 68 321: PLA
442d 28 322: PLP
442e 60 323: RTS
442f 324: ;
442f 325: ;
442f 326: ;
442f 327:
442f 328:
442f 329:
442f 330:
442f 526561 331: real_data: dc.b "RealDos Build 30"
443f 332: ds.b 40
4467 53494f 333: huh_data: dc.b "SIO2SD Detected!"
4477 334: ds.b 256
4577 335:
4577 336: win_end: ds.b 0
4577 337: ;
4577 338: ;
```

End assembly: no errors