


```

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: ;      SDMessage.com
0000 9: ;
0000 10: ;This is a program that sends one text strings to the disp
0000 11: ; from what is in quotes
0000 12: ;      init 10/16/2012   sjc new design
0000 13: ;
0000 14: ;
0000 15: ;
0000 16: ;-----
0000 17: ; Notes: o This source code MAY NOT be placed for download
0000 18: ;      o "mea" is a macro that loads the address of the
0000 19: ;      into the pointer specified by the second field.
0000 20: ;-----
0000 21: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 22: ;      XASM  st and IBM VERSIONS
0000 23: ;-----
0000 24:
0000 25: com_file_name:      .macro
0000 26: ;      dc.b 13,"filename.ext",$9b
0000 27: ;      dc.b 12,"SDMessage.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 16
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: r..bypassdos:      equ    $15 ; bypass dos check
0000      88: r..bpasintro:      equ    $ff
0000      89:
0000      90:
0000      91: l_frstscreenbyte: equ    r..crl
0000      92: ;~~~~~
0000      93: ; The language the output file is in.
0000      94: ;
0000      95: ;
0000      96: ; 0 =   Undefined
0000      97: ; 1 =   English
0000      98: ; 2 =   German
0000      99: ;
0000     100: r..language:      equ    1
0000     101: ;~~~~~
0000     102:
0000     103:
0000     104:          .include   equates
0000     105:          .include   globals
0000     106:          .include   macros

```

```

0000      107:
4000      108:                .org $4000
4000      109:
4000      110: win_start:
4000      111: header_info:
4000      112:                .include    header
42a7      113: Command_line:
42a7      114:
42a7 a03f  115:                ldy    #LBUF
42a9 b10a  116: mux_1a:        lda    (comtab),y
42ab c920  117:                cmp    #' '
42ad f012  118:                beq    mux_1b
42af c99b  119:                cmp    #$9b
42b1 f00b  120:                beq    .b
42b3 c93f  121:                cmp    #'?'
42b5 f004  122:                beq    .a
42b7 c8    123:                iny
42b8 4ca942 124:                jmp    mux_1a
42bb      125:
42bb 4c6943 126: .a:            jmp    parm_errors
42be 4c6343 127: .b:            jmp    myhold
42c1      128:
42c1      129: mux_1b:
42c1 a200  130:                ldx    #$00
42c3 c8    131:                iny
42c4      132:
42c4 b10a  133: .a:            lda    (comtab),y
42c6 c99b  134:                cmp    #$9b
42c8 f009  135:                beq    .noparm
42ca c922  136:                cmp    #""
42cc f008  137:                beq    .got_d
42ce c8    138:                iny
42cf 4cc442 139:                jmp    .a
42d2      140:
42d2 00     141: .holdx:        dc.b    0
42d3      142:
42d3 4c6943 143: .noparm:       jmp    parm_errors
42d6      144:
42d6      145: .got_d:
42d6 a201  146:                ldx    #$01
42d8 c8    147:                iny
42d9 b10a  148:                lda    (comtab),y
42db 9d5d44 149:                sta    parm_1,x
42de c99b  150:                cmp    #$9b
42e0 f0f1  151:                beq    .noparm
42e2 c922  152:                cmp    #""
42e4 f0ed  153:                beq    .noparm
42e6      154: .loop
42e6 e8     155:                inx
42e7 8e5d44 156:                stx    parm_1
42ea c8    157:                iny
42eb b10a  158:                lda    (comtab),y
42ed 9d5d44 159:                sta    parm_1,x

```

```

42f0 e01c 160:      cpx  #28
42f2 f004 161:      beq  .post
42f4 c922 162:      cmp  #""
42f6 d0ee 163:      bne  .loop
42f8 a99b 164: .post:      lda  #$9b
42fa 9d5d44 165:      sta  parm_1,x
42fd      166:
42fd      167:
42fd      168: .done:
42fd      169: done_1:
42fd      170:      mea  parm_1,pr_ptr
42fd a95d ----:      lda  #low parm_1
42ff 8588 ----:      sta  pr_ptr
4301 a944 ----:      lda  #high parm_1
4303 8589 ----:      sta  pr_ptr + 1
4305 20b445 171:      jsr  echos.unify
4308      172:      mea  parm_1,s_ptr
4308 a95d ----:      lda  #low parm_1
430a 8582 ----:      sta  s_ptr
430c a944 ----:      lda  #high parm_1
430e 8583 ----:      sta  s_ptr + 1
4310      173:      mea  parm_2,d_ptr
4310 a997 ----:      lda  #low parm_2
4312 8584 ----:      sta  d_ptr
4314 a944 ----:      lda  #high parm_2
4316 8585 ----:      sta  d_ptr + 1
4318 207d45 174:      jsr  move_str
431b 205546 175:      jsr  printsi
431e 9b9b 176:      dc.b  $9b,$9b
4320 20596f 177:      dc.b  " Your Message to be sent t
4348 203d3d 178:      dc.b  " ==>"
434c 22 179:      dc.b  $22
434d ff 180:      dc.b  -1
434e      181:      mea  parm_2,pr_ptr
434e a997 ----:      lda  #low parm_2
4350 8588 ----:      sta  pr_ptr
4352 a944 ----:      lda  #high parm_2
4354 8589 ----:      sta  pr_ptr + 1
4356 201746 182:      jsr  echos
4359 205546 183:      jsr  printsi
435c 229b9b 184:      dc.b  $22,$9b,$9b,-1
4360      185:
4360 4cd344 186:      jmp  start
4363      187:
4363      188:
4363 4cfd42 189: myhold:      jmp  done_1
4366      190: ;
4366      191:
4366      192:
4366 4c0000 193: cli.crunch:      jmp  $0000
4369      194:
4369      195: parm_errors:
4369      196:      mea  parm_1,pr_ptr

```

```

4369 a95d ----:    lda    #low parm_1
436b 8588 ----:    sta    pr_ptr
436d a944 ----:    lda    #high parm_1
436f 8589 ----:    sta    pr_ptr + 1
4371 20b445 197:    jsr    echos.unify
4374      198:
4374      199:    mea    parm_1,s_ptr
4374 a95d ----:    lda    #low parm_1
4376 8582 ----:    sta    s_ptr
4378 a944 ----:    lda    #high parm_1
437a 8583 ----:    sta    s_ptr + 1
437c      200:    mea    parm_2,d_ptr
437c a997 ----:    lda    #low parm_2
437e 8584 ----:    sta    d_ptr
4380 a944 ----:    lda    #high parm_2
4382 8585 ----:    sta    d_ptr + 1
4384 207d45 201:    jsr    move_str
4387      202: ;
4387 205546 203:    jsr    printsi
438a 9b9b  204:    dc.b    $9B,$9B
438c 205379 205:    dc.b    " Syntax for use:","$9b
439d 205344 206:    dc.b    " SDMessage "
43a7 22    207:    dc.b    $22
43a8 596f75 208:    dc.b    "Your Message!"
43b5 229b  209:    dc.b    $22,$9b
43b7 205344 210:    dc.b    " SDMessage "
43c1 22    211:    dc.b    $22
43c2 20596f 212:    dc.b    " Your Message!"
43d0 229b  213:    dc.b    $22,$9b
43d2 205344 214:    dc.b    " SDMessage "
43dc 22    215:    dc.b    $22
43dd      216:
43dd      217:
43dd      218:
43dd ff    219:    dc.b    -1
43de      220:
43de      221:    mea    parm_2,pr_ptr
43de a997 ----:    lda    #low parm_2
43e0 8588 ----:    sta    pr_ptr
43e2 a944 ----:    lda    #high parm_2
43e4 8589 ----:    sta    pr_ptr + 1
43e6 201746 222:    jsr    echos
43e9      223:
43e9 205546 224:    jsr    printsi
43ec 22    225:    dc.b    $22
43ed 9b204d 226:    dc.b    $9b," Message Limit =20 Byt
4406 9b9b9b 227:    dc.b    $9b,$9b,$9b
4409      228:
4409 9bff   229:    dc.b    $9b,-1
440b 60    230:    RTS
440c      231:
440c      232: ;~~~~~
440c      233: ;

```

```
440c      234: ; My Xio work Vars and temp registers for all ciov
440c      235: ;
440c      236: ;
440c 00    237: need_pause:      dc.b  0
440d      238: FILENAME:      ds.b  80
445d      239:
445d 115265 240: parm_1:      dc.b  17,"RealDos Build 3
446f      241:      ds.b  40
4497      242:
4497      243: parm_2:      ds.b  60
44d3      244: ;
44d3      245:
44d3      246: ;
44d3      247: Start:
44d3      248:
44d3 a50a  249:      LDA  COMTAB ; calc address of S
44d5 38    250:      SEC
44d6 e90a  251:      SBC  # low lsio
44d8 8d5d45 252:      STA  XSIO+1
44db a50b  253:      LDA  COMTAB+1
44dd e900  254:      SBC  # high lsio
44df 8d5e45 255:      STA  XSIO+2
44e2 a96c  256:      lda  #$6c
44e4 8d5c45 257:      sta  xsio
44e7      258:
44e7      259:
44e7      260:
44e7      261:
44e7      262:
44e7      263: SIO2SD_message:
44e7 a970  264:      lda  #$70
44e9 8d0003 265:      sta  $300
44ec a903  266:      lda  #$03
44ee 8d0103 267:      sta  $301
44f1 a910  268:      lda  #$10
44f3 8d0203 269:      sta  $0302
44f6 a980  270:      lda  #$80
44f8 8d0303 271:      sta  $0303
44fb a95e  272:      lda  # low parm_1+1
44fd 8d0403 273:      sta  $0304
4500 a944  274:      lda  # high parm_1+1
4502 8d0503 275:      sta  $0305
4505 a928  276:      lda  #40
4507 8d0803 277:      sta  $0308
450a a900  278:      lda  #$00
450c 8d0903 279:      sta  $0309
450f a900  280:      lda  #0
4511 8d0a03 281:      sta  $030a
4514 a900  282:      lda  #$00
4516 8d0b03 283:      sta  $030b
4519      284:
4519 205c45 285: status_device:  jsr  xsio
451c ad0303 286:      lda  $303
```

```

451f c901 287:      cmp    #$01
4521 d01a 288:      bne    sio2sd_error
4523      289: good_message:
4523 205546 290:      jsr    printsi
4526 9b9b9b 291:      dc.b   $9b,$9b,$9b
4529 202053 292:      dc.b   " Sio2SD "
4532 d0f2e5 293:      dc.b   <+128>,"Present!"
453a 9b 294:      dc.b   $9b
453b ff 295:      dc.b   -1
453c 60 296:      rts
453d      297: ;
453d      298: sio2sd_error:
453d 205546 299:      jsr    printsi
4540 9b9b9b 300:      dc.b   $9b,$9b,$9b
4543 205369 301:      dc.b   " Sio2SD "
454b ceef4 302:      dc.b   <+128>,"Not Present"
4556 9b 303:      dc.b   $9b
4557 ff 304:      dc.b   -1
4558 205f45 305:      jsr    driverror
455b 60 306:      rts
455c      307:
455c      308: ;
455c 4cffff 309: XSIO:      jmp    $ffff
455f      310:
455f      311: driverror:
455f 48 312:      pha
4560 205546 313:      jsr    printsi
4563 9b2053 314:      dc.b   $9b," Siov Error # ",$FF
4573 68 315:      pla
4574 209545 316:      jsr    pr_byte
4577 205546 317:      jsr    printsi
457a 9bff 318:      dc.b   $9b,-1
457c 60 319:      rts
457d      320:
457d      321: ;
457d      322: ;      Move string pointed to by s_ptr to string pointed
457d      323: ;      ~~~~~
457d      324:
457d      325:
457d a000 326: move_str:      ldy    #0
457f b182 327:      lda    (s_ptr),y      ;ge
4581 a8 328:      tay      ;go
4582 b182 329: .loop:      lda    (s_ptr),y
4584 c99b 330:      cmp    #$9b
4586 d002 331:      bne    .k
4588 a993 332:      lda    #$93
458a 9184 333: .k:      sta    (d_ptr),y
458c c000 334:      cpy    #0
458e f004 335:      beq    .done
4590 88 336:      dey
4591 4c8245 337:      jmp    .loop
4594 60 338: .done:      rts
4595      339:

```



```

4595      340:
4595      341:
4595      342:
4595      343:
4595      344: ;
4595      345: ;
4595      346: ;
4595      347: pr_byte:
4595 a200  348:      ldx  #$00
4597      349: pr_card:
4597 85d4  350:      sta  $d4
4599 86d5  351:      stx  $d4+1
459b a900  352:      lda  #$00
459d 85f2  353:      sta  $f2
459f 20aad9 354:      jsr  $d9aa
45a2 20e6d8 355:      jsr  $d8e6
45a5 a000  356:      ldy  #$00
45a7      357: .a:
45a7 b1f3  358:      lda  ($f3),y
45a9 08    359:      php
45aa 297f  360:      and  #$7f
45ac c8    361:      iny
45ad 203c46 362:      jsr  echo
45b0 28    363:      plp
45b1 10f4  364:      bpl  .a
45b3 60    365:      rts
45b4      366: ;
45b4      367:
45b4      368: ; this unifies the length string
45b4      369: ; ~~~~~
45b4      370:
45b4      371: echos.unify:
45b4 a000  372:      ldy  #0
45b6 b188  373:      lda  (pr_ptr),y      ;ge
45b8 f03e  374:      beq  .xpri           ;ex
45ba 8dfa45 375:      sta  .maxlth
45bd c8    376:      iny
45be 8cf945 377:      sty  .tempy
45c1      378:
45c1 acf945 379:      ldy  .tempy
45c4 b188  380:      lda  (pr_ptr),y
45c6 20fb45 381:      jsr  upcase
45c9 9188  382:      sta  (pr_ptr),y
45cb 4cf045 383:      jmp  .prnext
45ce      384:
45ce acf945 385: .pril:      ldy  .tempy
45d1 c002  386:      cpy  #$02
45d3 901b  387:      bcc  .prnext
45d5 b188  388:      lda  (pr_ptr),y      ; c
45d7 200946 389:      jsr  locase
45da 9188  390:      sta  (pr_ptr),y
45dc      391:
45dc c920  392:      cmp  #' '           ; l

```

```

45de d010 393:      bne .prnext
45e0 eef945 394:      inc .tempy
45e3 cefa45 395:      dec .maxlth
45e6 acf945 396:      ldy .tempy
45e9 b188 397:      lda (pr_ptr),y      ; m
45eb 20fb45 398:      jsr upcase
45ee 9188 399:      sta (pr_ptr),y
45f0      400:
45f0      401:
45f0 eef945 402: .prnext:      inc .tempy
45f3 cefa45 403:      dec .maxlth
45f6 d0d6 404:      bne .pril
45f8 60 405: .xpri:      rts
45f9      406:
45f9 01 407: .tempy:      dc.b 1
45fa 01 408: .maxlth:      dc.b 1
45fb      409:
45fb      410:
45fb      411: ;
45fb      412: ;
45fb c961 413: upcase:      cmp #'a'
45fd 9009 414:      bcc .xupcase
45ff c97b 415:      cmp #'z'+1
4601 f005 416:      beq .xupcase
4603 b003 417:      bcs .xupcase
4605 38 418:      sec
4606 e920 419:      sbc #32
4608 60 420: .xupcase:      rts
4609      421:
4609      422:
4609 c941 423: locase:      cmp #'A'
460b 9009 424:      bcc .xlocase
460d c95b 425:      cmp #'Z'+1
460f f005 426:      beq .xlocase
4611 b003 427:      bcs .xlocase
4613 18 428:      clc
4614 6920 429:      adc #32
4616 60 430: .xlocase:      rts
4617      431:
4617      432:
4617      433:
4617      434:
4617      435: ;
4617      436: ; prints: prints string with address in PR_PTR, length by
4617      437: ; ~~~~~
4617      438:
4617      439: echos:
4617 a000 440:      ldy #0
4619 b188 441:      lda (pr_ptr),y      ;ge
461b f01b 442:      beq .xpri      ;ex
461d 8d3a46 443:      sta .maxlth
4620 c8 444:      iny
4621 8c3946 445:      sty .tempy

```

```

4624      446:
4624 ac3946 447: .pril:      ldy    .tempy
4627 b188  448:      lda    (pr_ptr),y
4629 c900  449:      cmp    #$00
462b f00b  450:      beq    .xpri
462d 203c46 451:      jsr    echo
4630 ee3946 452: .prnext:      inc    .tempy
4633 ce3a46 453:      dec    .maxlth
4636 d0ec  454:      bne    .pril
4638 60     455: .xpri:      rts
4639      456:
4639 01     457: .tempy:      dc.b    1
463a 01     458: .maxlth:     dc.b    1
463b      459:
463b 00     460: pr_mode:     dc.b    0
463c      461:
463c      462:
463c      463: ;
463c      464: ;
463c      465: ;=====
463c      466: ;   print/input routines
463c      467: ;=====
463c      468: ;
463c      469: ;   Print character
463c      470: ;   -----
463c      471: ; in:
463c      472: ;   A    = character to print
463c      473: ; out:
463c      474: ;   all registers preserved
463c      475: ;
463c 208746 476: ECHO:      JSR    SAVER
463f 204546 477:      JSR    ZOUT
4642 4cb746 478:      JMP    RESALL
4645      479: ;
4645 8d5146 480: ZOUT:      STA    ZTEMP+1
4648 ad4703 481:      LDA    $0347
464b 48     482:      PHA
464c ad4603 483:      LDA    $0346
464f 48     484:      PHA
4650 a900  485: ZTEMP:     LDA    #0
4652 a200  486:      LDX    #0          ;1
4654 60     487:      RTS
4655      488: ;
4655      489: ;
4655      490: ;   Print text
4655      491: ;   -----
4655      492: ; out:
4655      493: ;   all registers preserved
4655      494: ; notes:
4655      495: ;   This print routine will print all characters
4655      496: ;   following the JSR PRINT until a delimiter of
4655      497: ;   -1 ($FF) is reached. PRINT will return to the
4655      498: ;   point one byte beyond the delimiter.

```

```

4655      499: ;
4655 208746 500: printsi:      JSR  SAVER
4658 ba    501:      TSX
4659 bd0501 502:      LDA  $0105,X
465c 8d6846 503:      STA  ZOCH+1
465f bd0601 504:      LDA  $0106,X
4662 8d6946 505:      STA  ZOCH+2
4665      506: ;
4665 a001  507:      LDY  #1
4667 b9ffff 508: ZOCH:      LDA  $FFFF,Y
466a c9ff   509:      CMP  #$FF
466c f006   510:      BEQ  ZECHO
466e 203c46 511:      JSR  ECHO
4671 c8     512:      INY
4672 d0f3   513:      BNE  ZOCH
4674 98     514: ZECHO:      TYA
4675 18     515:      CLC
4676 7d0501 516:      ADC  $0105,X
4679 9d0501 517:      STA  $0105,X
467c bd0601 518:      LDA  $0106,X
467f 6900   519:      ADC  #0
4681 9d0601 520:      STA  $0106,X
4684 4cb746 521:      JMP  RESALL
4687      522: ;
4687      523: ;
4687      524: ;   SAVE & RESTORE registers
4687      525: ;   -----
4687      526: ;
4687 08     527: SAVER:      PHP
4688 48     528:      PHA
4689 48     529:      PHA
468a 48     530:      PHA
468b 08     531:      PHP
468c 48     532:      PHA
468d 8a     533:      TXA
468e 48     534:      PHA
468f ba     535:      TSX
4690 bd0901 536:      LDA  $0109,X
4693 9d0501 537:      STA  $0105,X
4696 bd0701 538:      LDA  $0107,X
4699 9d0901 539:      STA  $0109,X
469c bd0101 540:      LDA  $0101,X
469f 9d0701 541:      STA  $0107,X
46a2 bd0801 542:      LDA  $0108,X
46a5 9d0401 543:      STA  $0104,X
46a8 bd0601 544:      LDA  $0106,X
46ab 9d0801 545:      STA  $0108,X
46ae 98     546:      TYA
46af 9d0601 547:      STA  $0106,X
46b2 68     548:      PLA
46b3 aa     549:      TAX
46b4 68     550:      PLA
46b5 28     551:      PLP

```

```
46b6 60 552:          RTS
46b7      553: ;
46b7 68 554: RESALL:          PLA
46b8 a8 555:          TAY
46b9 68 556:          PLA
46ba aa 557:          TAX
46bb 68 558:          PLA
46bc 28 559:          PLP
46bd 60 560:          RTS
46be      561: ;
46be      562: ;
46be      563: ;
46be      564:
46be      565:
46be      566:
46be      567: win_end:      ds.b 0
46be      568: ;
46be      569: ;
```

End assembly: no errors