



```

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: ;      Reset Mux utility for real.dos v1.0)
0000 10: ;-----
0000 11: ;      Real.dos savemux.com
0000 12: ;      This program is to replace
0000 13: ;      savemux.com
0000 14: ;
0000 15: ;      Code History.  Init design,  October 7, 2012
0000 16: ;
0000 17: ;      Updated the screen Print Utility so it will work wi
0000 18: ; This change will also help "Print P:"      8/1
0000 19: ;
0000 20: ;
0000 21: ;Also added code for Control+1 pause of screen to the high
0000 22: ;handling.      8/21/2012      sjc
0000 23: ;;
0000 24: ;
0000 25: ;      ~~~~~
0000 26: ;-----
0000 27: ; Notes: o This source code MAY NOT be placed for download
0000 28: ;      o "mea" is a macro that loads the address of the
0000 29: ;      into the pointer specified by the second field.
0000 30: ;-----
0000 31: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 32: ;      XASM  st and IBM VERSIONS
0000 33: ;-----
0000 34: ;=====
0000 35: com_file_name:      .macro
0000 36: ;      dc.b  13,"filename.ext",$9b
0000 37: ;      dc.b  12,"Savemux.com ", $9b
0000 38: ;      .endm
0000 39: ;
0000 40: ;
0000 41: ;~~~~~
0000 42: ; File revision history. Version Number in hex
0000 43: ;
0000 44: file_ver:      equ  $10
0000 45: ;~~~~~
0000 46: ; the Month in dec for the first time this revision
0000 47: ;was compiled
0000 48: ;
0000 49: c_month:      equ  10
0000 50: ;~~~~~
0000 51: ; the day in dec for the first time this revision
0000 52: ;was compiled
0000 53: c_day:      equ  7

```

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

```

0000 107: ;~~~~~
0000 108:
0000 109:
0000 110:         .include equates
0000 111:         .include globals
0000 112:         .include macros
0000 113:
0000 114:
0000 115:
0000 116:         .org $5000
0000 117: win_start:
0000 118: header_info:
0000 119:         .include header
52a7 120:
52a7 4c5155 121:         JMP MUXIO
52aa 122:
52aa 123: ;
52aa 124: ;
52aa 125: ;-----
52aa 126: ;prints Routine macro
52aa 127: ;this is a slow print to the screen
52aa 128: ;usage jsr prints
52aa 129: ; .byte $9b," string to be printed",$ff
52aa 130: ;-----
52aa 131: ;
52aa 132: prints:
52aa 68 133:         pla
52ab 8dbb52 134:         sta .pstr+1
52ae 68 135:         pla
52af 8dbc52 136:         sta .pstr+2
52b2 eebb52 137: .prsl:   inc .pstr+1
52b5 d003 138:         bne .pstr
52b7 eebe52 139:         inc .pstr+2
52ba adffff 140: .pstr:   lda $ffff
52bd c9ff 141:         cmp #$ff
52bf f006 142:         beq .estri
52c1 200053 143:         jsr fast_output
52c4 4cb252 144:         jmp .prsl
52c7 adbc52 145: .estri:  lda .pstr+2
52ca 48 146:         pha
52cb adbb52 147:         lda .pstr+1
52ce 48 148:         pha
52cf 60 149:         rts
52d0 150: ;
52d0 151: ;
52d0 152: ;
52d0 153: ;-----
52d0 154: ;actual screen handler
52d0 155: ;screen offset definitions
52d0 156: ;-----
52d0 157: scr_offset:
52d0 0000 158:         dc.w 0
52d2 2800 159:         dc.w 40

```

```

52d4 5000 160:          dc.w 80
52d6 7800 161:          dc.w 120
52d8 a000 162:          dc.w 160
52da c800 163:          dc.w 200
52dc f000 164:          dc.w 240
52de 1801 165:          dc.w 280
52e0 4001 166:          dc.w 320
52e2 6801 167:          dc.w 360
52e4 9001 168:          dc.w 400
52e6 b801 169:          dc.w 440
52e8 e001 170:          dc.w 480
52ea 0802 171:          dc.w 520
52ec 3002 172:          dc.w 560
52ee 5802 173:          dc.w 600
52f0 8002 174:          dc.w 640
52f2 a802 175:          dc.w 680
52f4 d002 176:          dc.w 720
52f6 f802 177:          dc.w 760
52f8 2003 178:          dc.w 800
52fa 4803 179:          dc.w 840
52fc 7003 180:          dc.w 880
52fe 9803 181:          dc.w 920
5300      182: ;
5300      183: fast_output:
5300 855a 184:          sta $5a
5302 ad0407 185:         lda sc.redirect
5305 c902 186:          cmp #print_p
5307 f01f 187:          beq .go_cio
5309      188: .leo:
5309 adff02 189:         lda $02ff      ; CONTROL 1
530c c900 190:          cmp #$00
530e d0f9 191:          bne .leo
5310      192:
5310 a55a 193:          lda $5a
5312 c920 194:          cmp #32
5314 9004 195:          bcc .do_look
5316 c97d 196:          cmp #125
5318 9013 197:          bcc .do_here
531a a000 198: .do_look:      ldy #0
531c b92354 199: .lookup:      lda .char_table,y
531f f00c 200:          beq .do_here
5321 c55a 201:          cmp $5a
5323 f003 202:          beq .go_cio
5325 c8 203:          iny
5326 d0f4 204:          bne .lookup
5328      205:
5328      206:
5328 a55a 207: .go_cio:      lda $5a
532a 4c3354 208:          jmp putlocal
532d      209:
532d      210:
532d a000 211: .do_here:      ldy #0
532f a55d 212:          lda $5d

```

```

5331 915e 213:      sta  ($5e),y
5333 a55a 214:      lda  $5a
5335 c99b 215:      cmp  #$9b
5337 f044 216:      beq  .docr
5339 200254 217: .notcr:      jsr  .get_adr
533c a55a 218:      lda  $5a
533e 297f 219:      and  #$7f
5340 c920 220:      cmp  #32
5342 9007 221:      bcc  .add64
5344 c960 222:      cmp  #96
5346 9009 223:      bcc  .sub32
5348 4c5453 224:      jmp  .asis
534b 18 225: .add64:      clc
534c 6940 226:      adc  #64
534e 4c5453 227:      jmp  .asis
5351 38 228: .sub32:      sec
5352 e920 229:      sbc  #32
5354 245a 230: .asis:      bit  $5a
5356 1002 231:      bpl  .xxlate
5358 0980 232:      ora  #$80
535a a000 233: .xxlate:      ldy  #0
535c 915e 234:      sta  ($5e),y
535e e655 235:      inc  $55
5360 e663 236:      inc  $63
5362 a555 237:      lda  $55
5364 c928 238:      cmp  #40
5366 b019 239:      bcs  .next_row
5368 c8 240:      iny
5369 b15e 241:      lda  ($5e),y
536b 855d 242:      sta  $5d
536d 0980 243:      ora  #$80
536f aef002 244:      ldx  752
5372 d002 245:      bne  .nocurs1
5374 915e 246:      sta  ($5e),y
5376 e65e 247: .nocurs1:      inc  $5e
5378 d002 248:      bne  .nooav
537a e65f 249:      inc  $5e+1
537c 60 250: .nooav:      rts
537d a900 251: .docr:      lda  #0
537f 8563 252:      sta  $63
5381 a552 253: .next_row:      lda  $52
5383 8555 254:      sta  $55
5385 e654 255:      inc  $54
5387 a454 256:      ldy  $54
5389 c018 257:      cpy  #24
538b b013 258:      bcs  .scroll
538d 200254 259:      jsr  .get_adr
5390 a000 260:      ldy  #0
5392 b15e 261:      lda  ($5e),y
5394 855d 262:      sta  $5d
5396 aef002 263:      ldx  752
5399 d004 264:      bne  .nocurs2
539b 0980 265:      ora  #$80

```

```

539d 915e 266:      sta  ($5e),y
539f 60 267: .nocurs2:      rts
53a0 268:
53a0 c654 269: .scroll:      dec  $54
53a2 a558 270:      lda  $58
53a4 8568 271:      sta  $68
53a6 18 272:      clc
53a7 6928 273:      adc  #40
53a9 855e 274:      sta  $5e
53ab a559 275:      lda  $58+1
53ad 8569 276:      sta  $68+1
53af 6900 277:      adc  #0
53b1 855f 278:      sta  $5e+1
53b3 a000 279:      ldy  #0
53b5 a203 280:      ldx  #3
53b7 b15e 281: .sloop:      lda  ($5e),y
53b9 9168 282:      sta  ($68),y
53bb c8 283:      iny
53bc d0f9 284:      bne  .sloop
53be e65f 285:      inc  $5e+1
53c0 e669 286:      inc  $68+1
53c2 ca 287:      dex
53c3 d0f2 288:      bne  .sloop
53c5 b15e 289: .sloop2:      lda  ($5e),y
53c7 9168 290:      sta  ($68),y
53c9 c8 291:      iny
53ca c098 292:      cpy  #152
53cc 90f7 293:      bcc  .sloop2
53ce a558 294:      lda  $58
53d0 18 295:      clc
53d1 6998 296:      adc  #920&$ff
53d3 855e 297:      sta  $5e
53d5 a559 298:      lda  $58+1
53d7 6903 299:      adc  #920/256
53d9 855f 300:      sta  $5e+1
53db a000 301:      ldy  #0
53dd 98 302:      tya
53de 915e 303: .clear:      sta  ($5e),y
53e0 c8 304:      iny
53e1 c028 305:      cpy  #40
53e3 90f9 306:      bcc  .clear
53e5 aef002 307:      ldx  752
53e8 d006 308:      bne  .nocurs3
53ea a980 309:      lda  #$80
53ec a455 310:      ldy  $55
53ee 915e 311:      sta  ($5e),y
53f0 a900 312: .nocurs3:      lda  #0
53f2 855d 313:      sta  $5d
53f4 a55e 314:      lda  $5e
53f6 18 315:      clc
53f7 6555 316:      adc  $55
53f9 855e 317:      sta  $5e
53fb a55f 318:      lda  $5e+1

```

```

53fd 6900 319:      adc  #0
53ff 855f 320:      sta  $5e+1
5401 60 321:      rts
5402 a554 322: .get_adr:      lda  $54
5404 0a 323:      asl
5405 a8 324:      tay
5406 a558 325:      lda  $58
5408 18 326:      clc
5409 79d052 327:      adc  scr_offset,y
540c 855e 328:      sta  $5e
540e a559 329:      lda  $58+1
5410 79d152 330:      adc  scr_offset+1,y
5413 855f 331:      sta  $5e+1
5415 a55e 332:      lda  $5e
5417 18 333:      clc
5418 6555 334:      adc  $55
541a 855e 335:      sta  $5e
541c a55f 336:      lda  $5e+1
541e 6900 337:      adc  #0
5420 855f 338:      sta  $5e+1
5422 60 339:      rts
5423      340: .char_table:
5423 1b1c1d 341:      dc.b 27,28,29,30,31,125,126,127
542b 9c9d9e 342:      dc.b 156,157,158,159,253,254,255
5433      343:
5433      344:
5433 a20b 345: putlocal:      ldx  #11
5435 8e4203 346:      stx  $0342
5438 a200 347:      ldx  #0
543a 8e4803 348:      stx  $0348
543d 8e4903 349:      stx  $0349
5440 4c56e4 350:      jmp  $e456
5443      351: ;
5443      352:
5443      353: ;
5443      354: c_iov:
5443 2056e4 355:      jsr  $e456
5446 ad0303 356:      lda  $0303
5449 a8 357:      tay
544a c081 358:      cpy  #$81
544c 9003 359:      bcc  c_iov_exit
544e 205354 360:      jsr  driverror
5451 60 361: c_iov_exit:      rts
5452 60 362:      rts
5453      363: driverror:
5453 48 364:      pha
5454 20aa52 365:      jsr  printsi
5457 9b2043 366:      dc.b $9b," Ciov Error # ", $FF
5467 68 367:      pla
5468 209254 368:      jsr  pr_byte
546b 20aa52 369:      jsr  printsi
546e 9bff 370:      dc.b $9b,-1
5470 60 371:      rts

```



```

5471      372: ;;
5471      373: ;
5471 00    374: STATUS1:      dc.b  $00
5472      375: STATUS:
5472 8d7154 376:      STA  STATUS1
5475 208354 377:      JSR  XIO_CLEAR
5478 ad7154 378:      LDA  STATUS1
547b 9d4a03 379:      STA  ICAX1,X
547e a90d   380:      LDA  #$0D
5480 4c0555 381:      JMP  XIO_WORK
5483      382: ;
5483      383: ;
5483      384: xio_clear:
5483 a900   385:      lda  #$00
5485 8d2455 386:      sta  prptr
5488 8d2555 387:      sta  prptr+1
548b 8d2255 388:      sta  lnptr
548e 8d2355 389:      sta  lnptr+1
5491 60     390:      rts
5492      391:
5492      392: ;
5492      393: pr_byte:
5492 a200   394:      ldx  #$00
5494      395: pr_card:
5494 85d4   396:      sta  $d4
5496 86d5   397:      stx  $d4+1
5498 a900   398:      lda  #$00
549a 85f2   399:      sta  $f2
549c 20aad9 400:      jsr  $d9aa
549f 20e6d8 401:      jsr  $d8e6
54a2 a000   402:      ldy  #$00
54a4      403: .aprbloop:
54a4 b1f3   404:      lda  ($f3),y
54a6 08     405:      php
54a7 297f   406:      and  #$7f
54a9 c8     407:      iny
54aa 20b154 408:      jsr  echo
54ad 28     409:      plp
54ae 10f4   410:      bpl  .aprbloop
54b0 60     411:      rts
54b1      412: ;
54b1      413: ;-----
54b1      414: ; PRINT BYTE
54b1      415: ;-----
54b1      416: echo:
54b1 85b8   417: sprint:      sta  $b8
54b3 a904   418:      lda  #$04
54b5 8d4a03 419:      sta  $034a
54b8 86b6   420:      stx  $b6
54ba 84b7   421:      sty  $b7
54bc a206   422:      ldx  #$06
54be bd01e4 423:      lda  $e400+1,x
54c1 85b3   424:      sta  $b3

```

```

54c3 bd00e4 425:      lda  $e400,x
54c6 85b2 426:      sta  $b2
54c8 a5b8 427:      lda  $b8
54ca 20d954 428:      jsr  .a
54cd a6b6 429:      ldx  $b6
54cf a4b7 430:      ldY  $b7
54d1 a90c 431:      lda  #$0c
54d3 8d4a03 432:      sta  $034a
54d6 a5b8 433:      lda  $b8
54d8 60 434:      rts
54d9 435: ;
54d9 e6b2 436: .a:      inc  $b2
54db 6cb200 437:      jmp  ($b2)
54de 60 438:      rts
54df 439:
54df 440: ;
54df 441: ;
54df 442: ;
54df 443: ;
54df 444: get:
54df a905 445:      lda  #$05
54e1 9d4203 446:      sta  $0342,x
54e4 ad2455 447:      lda  prptr
54e7 9d4403 448:      sta  $0344,x
54ea ad2555 449:      lda  prptr+1
54ed 9d4503 450:      sta  $0345,x
54f0 ad2255 451:      lda  lnptr
54f3 9d4803 452:      sta  $0348,x
54f6 ad2355 453:      lda  lnptr+1
54f9 9d4903 454:      sta  $0349,x
54fc a903 455:      lda  #$03
54fe 9d4a03 456:      sta  $034a,x
5501 457: ;      LDA  #0
5501 458: ;      STA  $034B,X
5501 204354 459:      jsr  c_iov
5504 60 460:      rts
5505 461: ;
5505 462: ;
5505 463: xio_work:
5505 9d4203 464:      sta  $0342,x
5508 a904 465:      lda  #$04
550a 9d4a03 466:      sta  $034a,x
550d a97f 467:      lda  #$7f
550f 9d4b03 468:      sta  $034b,x
5512 ad2755 469:      lda  fnptr
5515 9d4403 470:      sta  $0344,x
5518 ad2855 471:      lda  fnptr+1
551b 9d4503 472:      sta  $0345,x
551e 204354 473:      jsr  c_iov
5521 60 474:      rts
5522 475: ;
5522 0000 476: lnptr:      dc.w  0
5524 000000 477: prptr:      dc.b  0,0,0

```

```
5527 0000 478: fnptr:      dc.w  0
5529 4d3a9b 479: M_DEVICE      dc.b  "M:",$9B,$00
552d      480:
552d 000000 481: HOLD_M:      dc.b  0,0,0,0,0,0,0,0,0
5551      482:
5551      483: MUXIO:
5551 a220 484:      LDX  #$20
5553 208b55 485:      JSR  CLOSE
5556 a929 486:      LDA  #low M_DEVICE
5558 8d2755 487:      STA  FNPTR
555b a955 488:      LDA  #high M_DEVICE
555d 8d2855 489:      STA  FNPTR+1
5560 a220 490:      LDX  #$20
5562      491:
5562 a996 492:      LDA  #150      ;save confi
5564 200555 493:      JSR  XIO_WORK
5567 a220 494:      LDX  #$20
5569 207254 495:      JSR  STATUS
556c a901 496:      LDA  #1
556e 8d2255 497:      STA  LNPTR
5571 a900 498:      LDA  #$00
5573 8d2355 499:      STA  LNPTR+1
5576 a92d 500:      LDA  #low HOLD_M
5578 8d2455 501:      STA  PRPTR
557b a955 502:      LDA  #high HOLD_M
557d 8d2555 503:      STA  PRPTR+1
5580 a220 504:      LDX  #$20
5582 20df54 505:      JSR  GET
5585 a220 506:      LDX  #$20
5587 208b55 507:      JSR  CLOSE
558a 60 508:      RTS
558b      509:
558b      510: close:
558b a90c 511:      LDA  #$0C
558d a220 512:      LDX  #$20
558f 9d4203 513:      STA  ICCOM,X
5592 204354 514:      JSR  c_iov
5595 60 515:      RTS
5596      516:
5596      517: win_end:      ds.b  0
5596      518:
```

End assembly: no errors