



```

0000 1: ;-----
0000 2: ;      Copyright 2010 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: ;      Poke.com
0000 9: ;
0000 10: ; This is a program to poke a byte and display it to
0000 11: ;      init 3/08/2010   sjc new design
0000 12: ;      revisited 4/26/2012   sjc
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,"Poke.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 $16
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 8
0000 39: ;~~~~~
0000 40: ; the day in dec for the first time this revision
0000 41: ;was compiled
0000 42: c_day:      equ 26
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 our Header control Byte
0000      83: ;$ff=bypass intro
0000      84: ;$15 = Bypass Dos check
0000      85: r..crl:           equ    $7d
0000      86: r..crlf:          equ    $9b
0000      87: r..space:         equ    $20
0000      88: r..bypassDOS:      equ    $15
0000      89: r..bypass:        equ    $ff
0000      90:
0000      91: l_frstscreenbyte: equ    r..space
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: ;
42a7      114: pokeabyte:
42a7 ad01d3 115:          lda    portb
42aa 8deb44 116:          sta    under.the.os.data
42ad      117:
42ad      118:
42ad a03f 119:          ldy    #LBUF
42af b10a 120: .a:          lda    (comtab),y
42b1 c920 121:          cmp    #' '
42b3 f01f 122:          beq    .gotspacechar
42b5 c99b 123:          cmp    #$9b
42b7 f012 124:          beq    .noparmerror
42b9      125:
42b9 c8 126:          iny
42ba a906 127:          lda    #consol_start
42bc cd1fd0 128:          cmp    consol
42bf f00d 129:          beq    .tellfilehistory
42c1 a907 130:          lda    #consol_any
42c3 cd1fd0 131:          cmp    consol
42c6 d009 132:          bne    .abort
42c8 4caf42 133:          jmp    .a
42cb      134:
42cb 4c8845 135: .noparmerror:    jmp    noparmerror
42ce 4cd04c 136: .tellfilehistory:    jmp    my_file_notice
42d1 4ced44 137: .abort:          jmp    poke.a.byte.abort
42d4      138:
42d4      139:
42d4      140: .gotspacechar:
42d4 a200 141:          ldx    #$00
42d6 c8 142:          iny
42d7      143:
42d7 b10a 144: .af:          lda    (comtab),y
42d9 20184c 145:          jsr    upcase
42dc c948 146:          cmp    #'H'
42de f020 147:          beq    .showhelp
42e0 c924 148:          cmp    #'$'
42e2 f01f 149:          beq    .gothex
42e4 c93f 150:          cmp    #'?'
42e6 f0e6 151:          beq    .tellfilehistory
42e8 c920 152:          cmp    #' '
42ea f045 153:          beq    .oknum
42ec c99b 154:          cmp    #$9b
42ee f03e 155:          beq    .badparms
42f0 9ddb48 156:          sta    mynumberstring,x
42f3 c930 157:          cmp    #'0'
42f5 903a 158:          bcc    .oknum
42f7 c93a 159:          cmp    #'9'+1

```

```

42f9 b033 160:          bcs  .badparms
42fb e8 161:          inx
42fc c8 162:          iny
42fd 4cd742 163:         jmp  .af
4300 4cba43 164: .showhelp: jmp  go_show_help
4303      165:
4303 c8 166: .gothex:  iny
4304 a980 167:          lda  #$80
4306 8de544 168:          sta  inhexmode
4309 b10a 169: .h:      lda  (comtab),y
430b 20184c 170:          jsr  upcase
430e 9ddb48 171:          sta  mynumberstring,x
4311 c99b 172:          cmp  #$9b
4313 f019 173:          beq  .badparms
4315 c920 174:          cmp  #' '
4317 f018 175:          beq  .got.second.parm
4319      176:
4319 c930 177:          cmp  #'0'
431b 900d 178:          bcc  .badchar
431d c93a 179:          cmp  #'9'+1
431f 9008 180:          bcc  .charok
4321 c941 181:          cmp  #'A'
4323 9005 182:          bcc  .badchar
4325 c947 183:          cmp  #'F'+1
4327 b005 184:          bcs  .badparms
4329 e8 185: .charok:  inx
432a c8 186: .badchar:  iny
432b 4c0943 187:         jmp  .h
432e      188:
432e 4c8845 189: .badparms: jmp  noparmerror
4331      190:
4331      191: .got.second.parm:
4331      192: .oknum:
4331 e8 193:          inx
4332 a99b 194: .eol:      lda  #$9b
4334 9ddb48 195: .goodhexeol: sta  mynumberstring,x
4337      196:
4337 a200 197:          ldx  #$00
4339 c8 198:          iny
433a b10a 199: .h.h:      lda  (comtab),y
433c 20184c 200:          jsr  upcase
433f 9dda49 201:          sta  my.poke.byte,x
4342      202:
4342 c924 203:          cmp  #'$'
4344 f021 204:          beq  .byte.in.hex
4346 c927 205:          cmp  #""
4348 f026 206:          beq  .gotascii
434a c99b 207:          cmp  #$9b
434c f047 208:          beq  .real.eol
434e c920 209:          cmp  #$20
4350 f02c 210:          beq  .got.more.here
4352 c930 211:          cmp  #'0'
4354 900d 212:          bcc  .me.badchar

```

```

4356 c93a 213:      cmp  #'9'+1
4358 9008 214:      bcc  .me.char.ok
435a c941 215:      cmp  #'A'
435c 9005 216:      bcc  .me.badchar
435e c947 217:      cmp  #'F'+1
4360 b001 218:      bcs  .me.badchar
4362      219:
4362 e8    220: .me.char.ok:      inx
4363 c8    221: .me.badchar:      iny
4364 4c3a43 222:      jmp  .h.h
4367      223:
4367 a901 224: .byte.in.hex:      lda  #$01
4369 8de644 225:      sta  byteinhex
436c c8    226:      iny
436d 4c3a43 227:      jmp  .h.h
4370      228:
4370      229:
4370 c8    230: .gotascii:      iny
4371 b10a 231:      lda  (comtab),y
4373 8de744 232:      sta  hold_the_byte_forlater
4376 a902 233:      lda  #$02
4378 8de644 234:      sta  byteinhex
437b 4c3a43 235:      jmp  .h.h
437e      236:
437e c8    237: .got.more.here:      iny
437f b10a 238:      lda  (comtab),y
4381 c92f 239:      cmp  #'/'
4383 d010 240:      bne  .real.eol
4385 c99b 241:      cmp  #$9b
4387 f00c 242:      beq  .real.eol
4389 c8    243:      iny
438a b10a 244:      lda  (comtab),y
438c c955 245:      cmp  #'U'
438e d005 246:      bne  .real.eol
4390 a901 247:      lda  #$01
4392 8dea44 248:      sta  under.the.os
4395      249:
4395      250:
4395 e8    251: .real.eol:      inx
4396 a99b 252:      lda  #$9b
4398 9dda49 253:      sta  my.poke.byte,x
439b      254:
439b 208546 255:      jsr  convert_to_byte
439e 8de744 256:      sta  hold_the_byte_forlater
43a1      257:
43a1 ade544 258: .donewithcli:      lda  inhexmode
43a4 c900 259:      cmp  #$00
43a6 f015 260:      beq  indecmode
43a8 20554b 261:      jsr  dohexconversion
43ab ae914b 262:      ldx  expr+1
43ae ad904b 263:      lda  expr
43b1 8de844 264:      sta  source_location
43b4 8ee944 265:      stx  source_location+1

```

```

43b7 4c5344 266:          jmp  hex_screen_print
43ba      267:
43ba      268:
43ba      269:
43ba      270:
43ba 4ced44 271: go_show_help:      jmp  poke.a.byte.abort
43bd      272:
43bd      273:
43bd      274: indecmode:
43bd 209f46 275:          jsr  get_number_cli
43c0 8de844 276:          sta  source_location
43c3 8ee944 277:          stx  source_location+1
43c6      278:
43c6      279: dec_screen_print:
43c6 20524c 280:          jsr  printsi
43c9 9b204c 281:          dc.b  "$9b," Location ",-1
43d5 ade844 282:          lda  source_location
43d8 aee944 283:          ldx  source_location+1
43db 20db4a 284:          jsr  pr_card
43de      285:
43de 20524c 286:          jsr  printsi
43e1 203dff 287:          dc.b  "=", -1
43e4      288:
43e4 ade844 289:          lda  source_location
43e7 8d0a44 290:          sta  location_byte+1
43ea ade944 291:          lda  source_location+1
43ed 8d0b44 292:          sta  location_byte+2
43f0      293:
43f0 ade844 294:          lda  source_location
43f3 8d3544 295:          sta  location_byte2+1
43f6 ade944 296:          lda  source_location+1
43f9 8d3644 297:          sta  location_byte2+2
43fc      298:
43fc adea44 299:          lda  under.the.os
43ff c900  300:          cmp  #$00
4401 f003  301:          beq  .not_under
4403 20c44c 302:          jsr  os_off
4406      303: .not_under:
4406      304:
4406 ade744 305:          lda  hold_the_byte_forlater
4409 8dffff 306: location_byte:      sta  $ffff
440c      307:
440c      308:
440c 20114c 309:          jsr  wait_vbi
440f 20114c 310:          jsr  wait_vbi
4412      311:
4412      312:
4412 adea44 313:          lda  under.the.os
4415 c900  314:          cmp  #$00
4417 f003  315:          beq  .not_under
4419 20bd4c 316:          jsr  os_on
441c      317: .not_under:
441c ade744 318:          lda  hold_the_byte_forlater

```

```

441f 20d94a 319:      jsr  pr_byte
4422 20524c 320:      jsr  printsi
4425 202020 321:      dc.b  " ",-1
442a      322: ;
442a      323: ;
442a adea44 324:      lda  under.the.os
442d c900 325:      cmp  #$00
442f f003 326:      beq  ..not_under
4431 20c44c 327:      jsr  os_off
4434      328: ..not_under:
4434      329:
4434      330:
4434 adffff 331: location_byte2:      lda  $ffff
4437 8dec44 332:      sta  tammy.byte
443a adea44 333:      lda  under.the.os
443d c900 334:      cmp  #$00
443f f003 335:      beq  ..not_under
4441 20bd4c 336:      jsr  os_on
4444      337: ..not_under:
4444 adec44 338:      lda  tammy.byte
4447      339:
4447 207a4c 340:      jsr  echo
444a 20524c 341:      jsr  printsi
444d 27 342:      dc.b  ""
444e 9b9b9b 343:      dc.b  $9b,$9b,$9b,-1
4452 60 344:      rts
4453      345: ;~~~~~
4453      346: ; Now this routeen does all the Hex converion and prints
4453      347: ; to the screen
4453      348: ;
4453      349: ;~~~~~
4453      350:
4453      351: hex_screen_print:
4453      352:
4453 20524c 353:      jsr  printsi
4456 9b204c 354:      dc.b  $9b," Location $",-1
4463 ade844 355:      lda  source_location
4466 aee944 356:      ldx  source_location+1
4469 202d4b 357:      jsr  pr_hex_card
446c      358:
446c 20524c 359:      jsr  printsi
446f 203d24 360:      dc.b  " = $",-1
4473      361:
4473 ade844 362:      lda  source_location
4476 8d9944 363:      sta  .location_byte+1
4479 ade944 364:      lda  source_location+1
447c 8d9a44 365:      sta  .location_byte+2
447f      366:
447f ade844 367:      lda  source_location
4482 8dc744 368:      sta  .location_byte2+1
4485 ade944 369:      lda  source_location+1
4488 8dc844 370:      sta  .location_byte2+2
448b      371:

```



```

448b      372:
448b adea44 373:      lda    under.the.os
448e c900  374:      cmp    #$00
4490 f003  375:      beq    .not_under
4492 20c44c 376:      jsr    os_off
4495      377: .not_under:
4495 ade744 378:      lda    hold_the_byte_forlater
4498 8dffff 379: .location_byte:      sta    $ffff
449b 8dec44 380:      sta    tammy.byte
449e      381:
449e 20114c 382:      jsr    wait_vbi
44a1 20114c 383:      jsr    wait_vbi
44a4      384:
44a4 adea44 385:      lda    under.the.os
44a7 c900  386:      cmp    #$00
44a9 f003  387:      beq    .n_under
44ab 20bd4c 388:      jsr    os_on
44ae      389: .n_under:
44ae adec44 390:      lda    tammy.byte
44b1      391:
44b1      392:
44b1 203c4b 393:      jsr    pr_hex_byte
44b4 20524c 394:      jsr    printsi
44b7 202020 395:      dc.b  " ",-1
44bc      396:
44bc adea44 397:      lda    under.the.os
44bf c900  398:      cmp    #$00
44c1 f003  399:      beq    .location_byte2
44c3 20c44c 400:      jsr    os_off
44c6      401:
44c6 adffff 402: .location_byte2:      lda    $ffff
44c9 8dec44 403:      sta    tammy.byte
44cc      404:
44cc adea44 405:      lda    under.the.os
44cf c900  406:      cmp    #$00
44d1 f003  407:      beq    .tn_under
44d3 20bd4c 408:      jsr    os_on
44d6      409: .tn_under:
44d6 adec44 410:      lda    tammy.byte
44d9 207a4c 411:      jsr    echo
44dc 20524c 412:      jsr    printsi
44df 27     413:      dc.b  ""
44e0 9b9b9b 414:      dc.b  $9b,$9b,$9b,-1
44e4 60     415:      rts
44e5      416:
44e5      417:
44e5      418:
44e5      419:
44e5      420:
44e5      421: ;~~~~~
44e5      422: ;this tells me what mode to be in decimal or hex
44e5      423: ;
44e5 00     424: inhexmode:      dc.b  0

```

```

44e6      425:
44e6 00    426: byteinhex:      dc.b  0
44e7      427:
44e7 00    428: hold_the_byte_forlater:  dc.b  0
44e8      429: ;~~~~~
44e8      430: ;this the the location of the byte we are going to read.
44e8      431: ;
44e8 0000   432: source_location:  dc.w  0
44ea      433: ;~~~~~
44ea      434: ;OS reference bytes
44ea 00    435: under.the.os:    dc.b  0
44eb 00    436: under.the.os.data: dc.b  0
44ec 00    437: tammy.byte:     dc.b  0
44ed      438:
44ed      439: ;~~~~~
44ed      440: ; Aborted the program with a consol key
44ed      441: ;
44ed      442: ;~~~~~
44ed      443: poke.a.byte.abort:
44ed 20234c 444:          jsr  bell
44f0 20524c 445:          jsr  printsi
44f3 9b9b   446:          dc.b  $9b,$9b
44f5 20     447:          dc.b  " "
44f6 919292 448:          dc.b  <+128>,"
4518 9b     449:          dc.b  $9B
4519 20     450:          dc.b  " "
451a fca0a0 451:          dc.b  <+128>,"|  Poke a byte i
453c 9b     452:          dc.b  $9B
453d 20     453:          dc.b  " "
453e fca0d7 454:          dc.b  <+128>,"| Written by: Steph
4560 9b     455:          dc.b  $9B
4561 20     456:          dc.b  " "
4562 9a9292 457:          dc.b  <+128>,"

```