


```

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: ; title 'Drive programming Stuff'
0000 10: ;
0000 11: ;-----
0000 12: ; Wipedisk.s
0000 13: ;
0000 14: ; Now an easy way to wipe an Atr just like a lowlevel
0000 15: ;
0000 16: ; ~~~~~
0000 17: ;-----
0000 18: ; Notes: o This source code MAY NOT be placed for download
0000 19: ; o "mea" is a macro that loads the address of the
0000 20: ; into the pointer specified by the second field.
0000 21: ;-----
0000 22: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 23: ; XASM st and IBM VERSIONS
0000 24: ;-----
0000 25: ;=====
0000 26:
0000 27: com_file_name: .macro
0000 28: ; dc.b 13,"filename.ext",$9b
0000 29: ; dc.b 12,"Wipedisk.com",$9b
0000 30: ; .endm
0000 31:
0000 32: ;
0000 33: ;~~~~~
0000 34: ; File revision history. Version Number in hex
0000 35: ;
0000 36: file_ver: equ $12
0000 37: ;~~~~~
0000 38: ; the Month in dec for the first time this revision
0000 39: ;was compiled
0000 40: ;
0000 41: c_month: equ 8
0000 42: ;~~~~~
0000 43: ; the day in dec for the first time this revision
0000 44: ;was compiled
0000 45: c_day: equ 26
0000 46: ;~~~~~
0000 47: ; the year in dec for the first time this revision
0000 48: ;was compiled
0000 49: c_year: equ 2012
0000 50: ;~~~~~
0000 51: ; Type of compiler used in program
0000 52: ;
0000 53: ; 0 = unknown Compiler

```

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

```
0000 107:          .include  globals
0000 108:          .include  macros
0000 109:
0000 110:
4000 111:          .org  $4000
4000 112:
4000 113: win_start:
4000 114: header_info:
4000 115:          .include  header
42a7 4cc045 116:          jmp  start
42aa 117:
42aa 118:
42aa 119:
42aa 00 120: par_drv_letter:      dc.b  0
42ab 121: ;
42ab 122:
42ab 123: cmdtab2:          ds.b  256
43ab 124:
43ab 00 125: par_Name:          dc.b  0
43ac 126:
43ac 127: buffer:          ds.b  256
44ac 128:
44ac 129: buffer_1:        ds.b  256
45ac 130:
45ac 8000 131: sector_size:      dc.w  128
45ae 08 132: srcdrv:          dc.b  8
45af 08 133: dstdrv:          dc.b  8
45b0 134:
45b0 0000 135: heads:          dc.w  0
45b2 0000 136: sector_per_trac:  dc.w  0
45b4 0000 137: percom_density:   dc.w  0
45b6 0000 138: real_sector_count: dc.w  0
45b8 0000 139: end_sector:      dc.w  0
45ba 140:
45ba 141:
45ba ff 142: sector_density:   dc.b  -1
45bb 143:
45bb 0000 144: start_sector:    dc.w  0
45bd 145:
45bd 146:
45bd 147:
45bd 4cffff 148: xiov:          jmp  $ffff
45c0 149:
45c0 150:
45c0 151: start:
45c0 152:
45c0 a03f 153:          LDY  #LBUF
45c2 154:
45c2 b10a 155: .comp:        lda  (comtab),y
45c4 c99b 156:          cmp  #$9b
45c6 f008 157:          beq  .exit
45c8 c920 158:          cmp  #' '
45ca f007 159:          beq  .cont
```

```

45cc c8 160:          iny
45cd 4cc245 161:      jmp  .comp
45d0 162:
45d0 163:
45d0 4c1a46 164: .exit:      jmp  ask_what_drive
45d3 165:
45d3 166:
45d3 167:
45d3 168:
45d3 169: .cont:
45d3 c8 170:          iny
45d4 c8 171:          iny
45d5 b10a 172:          LDA  (comtab),Y
45d7 8daa42 173:          sta  par_drv_letter
45da c931 174:          CMP  #'1'          ;$2
45dc 9007 175:          bcc  .ea
45de c93a 176:          cmp  #'9'+1
45e0 b003 177:          bcs  .ea
45e2 4ce845 178:          jmp  .c
45e5 179:
45e5 4c1a46 180: .ea:      jmp  ask_what_drive
45e8 181:
45e8 182:
45e8 a200 183: .c:      ldx  #$00
45ea c8 184:          iny          ; g
45eb c8 185:          iny          ; g
45ec 186:
45ec e8 187: .d:      inx
45ed c8 188: .e:      iny
45ee b10a 189:          lda  (comtab),y
45f0 c92f 190:          cmp  #'/'
45f2 f0f9 191:          beq  .e
45f4 c920 192:          cmp  #$20
45f6 f0f5 193:          beq  .e
45f8 c93a 194:          cmp  #':'
45fa f0f1 195:          beq  .e
45fc c951 196:          cmp  #'Q'
45fe f006 197:          beq  .tam
4600 c99b 198:          cmp  #$9b
4602 f005 199:          beq  .done_with_cli
4604 d0e7 200:          bne  .e
4606 201: .tam
4606 8dab43 202:          sta  par_Name
4609 203: ;      jmp  .d
4609 204:
4609 205: .done_with_cli:
4609 adaa42 206:          lda  par_drv_letter
460c e930 207:          sbc  #48
460e 8dae45 208:          sta  srcdrv
4611 209:
4611 210:
4611 4c3047 211:          jmp  start_it_now
4614 212:

```

```
4614      213: exit_program:
4614 a99b 214:          LDA  #$9B
4616 20ff4c 215:          JSR  echo
4619      216: ;          LDX  stack_save
4619      217: ;          TXS
4619 60 218:          RTS
461a      219: ask_what_drive:
461a 20804d 220:          jsr  printsi
461d 9b2057 221:          dc.b  $9b," What drive Number to
463d 202d4d 222:          jsr  get_key
4640 8daa42 223:          sta  par_drv_letter
4643 20424d 224:          jsr  upcase
4646 c941 225:          cmp  #'A'          ; a
4648 f030 226:          beq  .abort
464a c958 227:          cmp  #'X'
464c f02c 228:          beq  .abort
464e c91c 229:          cmp  #$1c          ;es
4650 f028 230:          beq  .abort
4652 c945 231:          cmp  #'E'          ; E
4654 f024 232:          beq  .abort
4656 c931 233:          CMP  #'1'          ;$2
4658 904f 234:          bcc  error
465a c93a 235:          cmp  #'9'+1
465c b04b 236:          bcs  error
465e adaa42 237:          lda  par_drv_letter
4661 20ff4c 238:          jsr  echo
4664 20804d 239:          jsr  printsi
4667 3aff 240:          dc.b  ":",-1
4669 adaa42 241:          lda  par_drv_letter
466c e930 242:          sbc  #48
466e 8dae45 243:          sta  srcdrv
4671 20804d 244:          jsr  printsi
4674 1c9cff 245:          dc.b  $1c,$9c,-1
4677 4c3047 246:          jmp  start_it_now
467a      247: .abort:
467a 20804d 248:          jsr  printsi
467d 9b 249:          dc.b  $9b
467e 204f70 250:          dc.b  " Operation Aborted!",$9b,-
4693 a900 251:          lda  #$00
4695 8df002 252:          sta  $2f0          ; t
4698 20804d 253:          jsr  printsi
469b 9b9b41 254:          dc.b  $9b,$9b," All Done!",$9b,-1
46a8 60 255:          rts
46a9      256:
46a9      257:
46a9      258: error:
46a9 20804d 259:          JSR  printsi
46ac 9b5769 260:          dc.b  $9b,"Wipedisk Dn:"
46b9 9b 261:          dc.b  $9b
46ba 446e3a 262:          dc.b  "Dn: n= drive id 1 to 9"
46d5 53796e 263:          dc.b  "Syntax =", $9b
46de 576970 264:          dc.b  "Wipedisk D8: "
46eb 5b5061 265:          dc.b  "[Parm]",$9b
```

```

46f2 5b5061 266:          dc.b "[Parm] = /Q will not give
4714 202761 267:          dc.b " 'are you Sure' message!"
472c 9b9bff 268:          dc.b $9b,$9b,-1
472f      269:
472f      270:
472f 60     271:          RTS
4730      272: ;
4730      273:
4730      274:
4730      275: start_it_now:
4730 adab43 276:          lda  par_Name
4733 20424d 277:          jsr  upcase
4736 c951   278:          cmp  #'Q'
4738 d003   279:          bne  .domsg
473a 4c2f48 280:          jmp  .time_to_zero.drv
473d      281: .domsg:
473d 20804d 282:          jsr  printsi
4740 9b9b9b 283:          dc.b $9b,$9b,$9b
4743 d4e8e9 284:          dc.b <+128>,"This will completel
4765 ff     285:          dc.b -1
4766      286:
4766 adaa42 287:          lda  par_drv_letter
4769 4980   288:          eor  #$80
476b 20ff4c 289:          jsr  echo
476e 20804d 290:          jsr  printsi
4771 babc   291:          dc.b <+128>,":<"
4773 9b     292:          dc.b $9b
4774 f7e9f4 293:          dc.b <+128>,"with $00! Are you s
4799 9b20   294:          dc.b $9b," "
479b dbf9dd 295:          dc.b <+128>,"[y]"
479e 657320 296:          dc.b "es "
47a1 dbcedd 297:          dc.b <+128>,"[N]"
47a4 6f203f 298:          dc.b "o ?",-1
47a8 202d4d 299:          jsr  get_key
47ab 20424d 300:          jsr  upcase
47ae c959   301:          cmp  #'Y'
47b0 d060   302:          bne  .bye
47b2      303:
47b2 20804d 304:          jsr  printsi
47b5 9b9b   305:          dc.b $9b,$9b
47b7 417265 306:          dc.b "Are you sure? WipeDisk D"
47cf ff     307:          dc.b -1
47d0 adaa42 308:          lda  par_drv_letter
47d3 20ff4c 309:          jsr  echo
47d6      310:
47d6      311:
47d6 20804d 312:          jsr  printsi
47d9 3a9b   313:          dc.b ":",$9b
47db dbf9dd 314:          dc.b <+128>,"[y]"
47de 657320 315:          dc.b "es "
47e1 dbcedd 316:          dc.b <+128>,"[N]"
47e4 6f20ff 317:          dc.b "o ",-1
47e7 202d4d 318:          jsr  get_key

```

```

47ea 20424d 319:      jsr  upcase
47ed c959  320:      cmp  #'Y'
47ef d021  321:      bne  .bye
47f1 20804d 322:      jsr  printsi
47f4 9b9b9b 323:      dc.b  $9b,$9b,$9b,$9b,$9b,$9b
47fa      324:
47fa 1c1c1c 325:      dc.b  $1c,$1c,$1c,$1c,$1c,$1c,$1c
4804 9c9c9c 326:      dc.b  $9c,$9c,$9c,$9c,$9c,$9c,$9c
480e      327:
480e ff     328:      dc.b  -1
480f 4c2f48 329:      jmp  .time_to_zero.drv
4812      330: .bye:
4812 20804d 331:      jsr  printsi
4815 9b9b9b 332:      dc.b  $9b,$9b,$9b
4818 cff0e5 333:      dc.b  <+128>,"Operation Aborted!"
482a 9bff   334:      dc.b  $9b,-1
482c 4c8648 335:      jmp  .all_done
482f      336:
482f      337: ;-----
482f      338: .time_to_zero.drv:
482f a900   339:      lda  #low 0
4831 8dbb45 340:      sta  start_sector
4834 a900   341:      lda  #high 0
4836 8dbb45 342:      sta  start_sector
4839 a000   343:      ldy  #$00
483b a900   344:      lda  #$00
483d 99ac44 345: .zero:      sta  buffer_1,y
4840 99ab42 346:      sta  cmdtab2,y
4843 c8     347:      iny
4844 c000   348:      cpy  #$00
4846 d0f5   349:      bne  .zero
4848      350:
4848      351:
4848      352:
4848 a000   353:      ldy  #$00
484a a900   354:      lda  #$00
484c 99ac43 355: .b6c:      sta  buffer,y
484f c8     356:      iny
4850 c000   357:      cpy  #$00
4852 d0f8   358:      bne  .b6c
4854 a202   359:      ldx  #2
4856 a00a   360:      ldy  #10
4858 20fa4c 361:      jsr  gotoxy
485b 20804d 362:      jsr  printsi
485e 436c65 363:      dc.b  "Cleaning up Disk "
486f ff     364:      dc.b  -1
4870      365:
4870      366:
4870 a900   367:      lda  #$00
4872 8dbb45 368:      sta  start_sector
4875 8dbc45 369:      sta  start_sector+1
4878      370:
4878 20434c 371:      jsr  calc_sector_count

```



```

487b a901 372:      lda  #$01
487d 8df002 373:      sta  $2f0          ; t
4880      374:
4880      375:
4880 209c48 376:      jsr  .show_name
4883      377:
4883 20f348 378:      jsr  run_loop
4886      379:
4886      380: .all_done:
4886 a900 381:      lda  #$00
4888 8df002 382:      sta  $2f0          ; t
488b 20804d 383:      jsr  printsi
488e 9b9b41 384:      dc.b  $9b,$9b,"All Done!",$9b,-1
489b 60 385:      rts
489c      386:
489c      387: .show_name:
489c      388:
489c      389:
489c a202 390:      ldx  #2
489e a00d 391:      ldy  #13
48a0 20fa4c 392:      jsr  gotoxy
48a3 20804d 393:      jsr  printsi
48a6 596f75 394:      dc.b  "Your Current Partition Has
48c3 adb845 395:      LDA  end_sector
48c6 38 396:      SEC
48c7 e901 397:      SBC  # low 1
48c9 8db845 398:      STA  end_sector
48cc adb945 399:      LDA  end_sector+1
48cf e900 400:      SBC  # high 1
48d1 8db945 401:      STA  end_sector+1
48d4      402:
48d4 adb645 403:      lda  real_sector_count
48d7 aeb745 404:      ldx  real_sector_count+1
48da 201b4f 405:      jsr  pr_card
48dd      406:
48dd 20804d 407:      jsr  printsi
48e0 279b 408:      dc.b  "",$9b
48e2 536563 409:      dc.b  "Sectors to fix!",-1
48f2      410:
48f2      411:
48f2      412:
48f2      413:
48f2 60 414:      rts
48f3      415:
48f3      416:
48f3      417: run_loop:
48f3 adbb45 418:      LDA  start_sector
48f6 18 419:      CLC
48f7 6901 420:      ADC  # low 1
48f9 8dbb45 421:      STA  start_sector
48fc adbc45 422:      LDA  start_sector+1
48ff 6900 423:      ADC  # high 1
4901 8dbc45 424:      STA  start_sector+1

```

```
4904      425:
4904      426:
4904 a202  427:      ldx  #2
4906 a014  428:      ldy  #20
4908 20fa4c 429:      jsr  gotoxy
490b 20804d 430:      jsr  printsi
490e 577269 431:      dc.b  "Writing Sector #",-1
491f      432:
491f adbb45 433:      lda  start_sector
4922 aebc45 434:      ldx  start_sector+1
4925      435:
4925 201b4f 436:      jsr  pr_card
4928      437:
4928 adbc45 438:      lda  start_sector+1
492b c900  439:      cmp  #$00
492d d014  440:      bne  .do_256
492f adbb45 441:      lda  start_sector
4932 c903  442:      cmp  #$03
4934 b00d  443:      bcs  .do_256
4936      444:
4936 a980  445:      lda  #low 128
4938 8dac45 446:      sta  sector_size
493b a900  447:      lda  #high 128
493d 8dad45 448:      sta  sector_size+1
4940 4c4d49 449:      jmp  .do_rw
4943      450:
4943      451:
4943 a900  452: .do_256:      lda  #low 256
4945 8dac45 453:      sta  sector_size
4948 a901  454:      lda  #high 256
494a 8dad45 455:      sta  sector_size+1
494d      456:
494d      457: .do_rw:
494d 206149 458:      jsr  read_write
4950      459:
4950 adbc45 460:      lda  start_sector+1
4953 cdb745 461:      cmp  real_sector_count+1 ;en
4956 d09b  462:      bne  run_loop
4958 adbb45 463:      lda  start_sector
495b cdb645 464:      cmp  real_sector_count ;en
495e d093  465:      bne  run_loop
4960      466:
4960 60     467:      rts          ; a
4961      468:
4961      469:
4961      470:
4961      471:
4961      472:
4961      473:
4961      474: read_write:
4961      475: ;read
4961 a931  476:      lda  #$31
4963 8d0003 477:      sta  $300
```

```
-----
4966 adae45 478:      lda  srcdrv
4969 8d0103 479:      sta  $301
496c a952 480:      lda  #'R'
496e 8d0203 481:      sta  $302
4971 a940 482:      lda  #$40
4973 8d0303 483:      sta  $303
4976 a9ac 484:      lda  #low buffer
4978 8d0403 485:      sta  dbuflo
497b a943 486:      lda  #high buffer
497d 8d0503 487:      sta  dbufhi
4980 adac45 488:      lda  sector_size
4983 8d0803 489:      sta  $308
4986      490:
4986 adad45 491:      lda  sector_size+1
4989 8d0903 492:      sta  $309
498c adbb45 493:      lda  start_sector
498f 8d0a03 494:      sta  $30a
4992 adbc45 495:      lda  start_sector+1
4995 8d0b03 496:      sta  $30b
4998 20da49 497:      jsr  do_r_siov
499b 4c9f49 498:      jmp  .write_it
499e      499:
499e 60 500: .nowrt:      rts
499f      501:
499f      502:
499f      503:
499f      504: .write_it
499f a931 505:      lda  #$31
49a1 8d0003 506:      sta  $300
49a4 adae45 507:      lda  srcdrv
49a7 8d0103 508:      sta  $301
49aa a957 509:      lda  #'W'
49ac 8d0203 510:      sta  $302
49af a980 511:      lda  #$80
49b1 8d0303 512:      sta  $303
49b4 a9ac 513:      lda  #low buffer_1
49b6 8d0403 514:      sta  dbuflo
49b9 a944 515:      lda  #high buffer_1
49bb 8d0503 516:      sta  dbufhi
49be adac45 517:      lda  sector_size
49c1 8d0803 518:      sta  $308
49c4      519:
49c4 adad45 520:      lda  sector_size+1
49c7 8d0903 521:      sta  $309
49ca adbb45 522:      lda  start_sector
49cd 8d0a03 523:      sta  $30a
49d0 adbc45 524:      lda  start_sector+1
49d3 8d0b03 525:      sta  $30b
49d6 20eb49 526:      jsr  do_w_siov
49d9      527:
49d9 60 528:      rts
49da      529:
49da      530:
```

```

49da      531: do_r_siov:
49da a952  532:          lda  #'R'
49dc 8d0203 533:          sta  $302
49df a940  534:          lda  #$40
49e1 8d0303 535:          sta  $303
49e4 2059e4 536:          jsr  $e459
49e7 ad0303 537:          lda  $303
49ea 60    538:          rts
49eb      539:
49eb      540: do_w_siov:
49eb a957  541:          lda  #'W'
49ed 8d0203 542:          sta  $302
49f0 a980  543:          lda  #$80
49f2 8d0303 544:          sta  $303
49f5 2059e4 545:          jsr  $e459
49f8 ad0303 546:          lda  $303
49fb 60    547:          rts
49fc      548:
49fc      549:
49fc      550: get_sector_count:
49fc a931  551:          lda  #$31
49fe 8d0003 552:          sta  $300
4a01 adae45 553:          lda  srcdrv
4a04 8d0103 554:          sta  $301
4a07 a94e  555:          lda  #'N'
4a09 8d0203 556:          sta  $0302
4a0c a940  557:          lda  #$40
4a0e 8d0303 558:          sta  $0303
4a11 a91d  559:          lda  # low huh_data
4a13 8d0403 560:          sta  $0304
4a16 a94c  561:          lda  # high huh_data
4a18 8d0503 562:          sta  $0305
4a1b a90c  563:          lda  #$0c
4a1d 8d0803 564:          sta  $0308
4a20 a900  565:          lda  #$00
4a22 8d0903 566:          sta  $0309
4a25 8d0a03 567:          sta  $030a
4a28 8d0b03 568:          sta  $030b
4a2b 2059e4 569:          jsr  $e459
4a2e ad0303 570:          lda  $0303
4a31 c901  571:          cmp  #$01
4a33 f001  572:          beq  .huh_1
4a35 60    573:          rts
4a36      574: .huh_1:
4a36 ad1fd0 575:          lda  consol
4a39 c905  576:          cmp  #consol_select
4a3b f001  577:          beq  .tam
4a3d 60    578:          rts
4a3e      579: .tam:
4a3e 20804d 580:          jsr  printsi
4a41 7d9b9b 581:          dc.b $7d,$9b,$9b,$9b
4a45 202020 582:          dc.b "      Drive configuration
4a70 ad1d4c 583:          lda  huh_data

```

```

4a73 20064c 584:      jsr  sub_error
4a76 20804d 585:      jsr  printsi
4a79 6e756d 586:      dc.b  "number of tracks",$9B,$FF
4a8b ad1e4c 587:      lda  huh_data+1
4a8e 20064c 588:      jsr  sub_error
4a91 20804d 589:      jsr  printsi
4a94 537465 590:      dc.b  "Step rate! normaly 1",$9B,
4aaa ad1f4c 591:      lda  huh_data+2
4aad 20064c 592:      jsr  sub_error
4ab0 20804d 593:      jsr  printsi
4ab3 536563 594:      dc.b  "Sector/Track high byte",$9
4acb ad204c 595:      lda  huh_data+3
4ace 20064c 596:      jsr  sub_error
4ad1 20804d 597:      jsr  printsi
4ad4 536563 598:      dc.b  "Sector/Track LOW byte",$9B
4aeb ad214c 599:      lda  huh_data+4
4aee 20064c 600:      jsr  sub_error
4af1 20804d 601:      jsr  printsi
4af4 4d6178 602:      dc.b  "Max head number",$9B,$FF
4b05 ad224c 603:      lda  huh_data+5
4b08 20064c 604:      jsr  sub_error
4b0b 20804d 605:      jsr  printsi
4b0e 44656e 606:      dc.b  "Density -0=s, 4 double, 8
4b2e ad234c 607:      lda  huh_data+6
4b31 20064c 608:      jsr  sub_error
4b34 20804d 609:      jsr  printsi
4b37 427974 610:      dc.b  "Byte/sector h byte 1=256 0
4b57 ad244c 611:      lda  huh_data+7
4b5a 20064c 612:      jsr  sub_error
4b5d 20804d 613:      jsr  printsi
4b60 427974 614:      dc.b  "Byte/sector l byte 0=256 1
4b80 ad254c 615:      lda  huh_data+8
4b83 20064c 616:      jsr  sub_error
4b86 20804d 617:      jsr  printsi
4b89 447269 618:      dc.b  "Drive present flag -return
4baa ad264c 619:      lda  huh_data+9
4bad 20064c 620:      jsr  sub_error
4bb0 20804d 621:      jsr  printsi
4bb3 6e6f74 622:      dc.b  "not used",$9B,$FF
4bbd ad274c 623:      lda  huh_data+10
4bc0 20064c 624:      jsr  sub_error
4bc3 20804d 625:      jsr  printsi
4bc6 6e6f74 626:      dc.b  "not used",$9B,$FF
4bd0 ad284c 627:      lda  huh_data+11
4bd3 20064c 628:      jsr  sub_error
4bd6 20804d 629:      jsr  printsi
4bd9 6e6f74 630:      dc.b  "not used",$9B
4be2 9b5072 631:      dc.b  $9b,"Press any key to conti
4bfd 202d4d 632:      jsr  get_key
4c00 a97d   633:      lda  #$7d
4c02 20ff4c 634:      jsr  echo
4c05 60     635:      rts
4c06      636: sub_error:

```

```

4c06 8d1c4c 637:      sta  .dan_k
4c09 20804d 638:      jsr  printsi
4c0c 2024ff 639:      dc.b  "$", $FF
4c0f ad1c4c 640:      lda  .dan_k
4c12 20674d 641:      jsr  drive_error
4c15 20804d 642:      jsr  printsi
4c18 2020ff 643:      dc.b  " ", $FF
4c1b 60     644:      rts
4c1c      645: ;
4c1c 00     646: .dan_k:      dc.b  0
4c1d 000000 647: huh_data:    dc.b  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
4c30 000000 648: huh_data_d:  dc.b  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
4c43      649:
4c43      650:
4c43      651:
4c43      652:
4c43      653:
4c43      654:
4c43      655: calc_sector_count:
4c43 20fc49 656:      jsr  get_sector_count
4c46 a9ff   657:      lda  #$ff
4c48 8dba45 658:      sta  sector_density
4c4b ad224c 659:      lda  huh_data+5
4c4e c908   660:      cmp  #$08
4c50 d008   661:      bne  .a
4c52 a902   662:      lda  #$02
4c54 8db445 663:      sta  percom_density
4c57 4c6b4c 664:      jmp  .b
4c5a a901   665: .a:      lda  #$01
4c5c 8db445 666:      sta  percom_density
4c5f ad224c 667:      lda  huh_data+5
4c62 c900   668:      cmp  #$00
4c64 d005   669:      bne  .b
4c66 a980   670:      lda  #$80
4c68 8dba45 671:      sta  sector_density
4c6b      672:
4c6b      673: .b:
4c6b ad1f4c 674:      lda  huh_data+2
4c6e c900   675:      cmp  #$00
4c70 d00f   676:      bne  .c
4c72 ad204c 677:      lda  huh_data+3
4c75 c900   678:      cmp  #$00
4c77 d008   679:      bne  .c
4c79 a9ff   680:      lda  #$ff
4c7b 8d1f4c 681:      sta  huh_data+2
4c7e 8d204c 682:      sta  huh_data+3
4c81      683:
4c81 ad1f4c 684: .c:      lda  huh_data+2
4c84 8db345 685:      sta  sector_per_trac+1
4c87 ad204c 686:      lda  huh_data+3
4c8a 8db245 687:      sta  sector_per_trac
4c8d      688:
4c8d ad1d4c 689:      lda  huh_data

```

```

4c90 c903 690:      cmp    #$03
4c92 b005 691:      bcs    .d
4c94 a901 692:      lda    # low 1
4c96 8d1d4c 693:      sta    huh_data
4c99 ad1d4c 694: .d:      lda    huh_data
4c9c 8db045 695:      sta    heads
4c9f a900 696:      lda    #$00
4ca1 8db145 697:      sta    heads+1
4ca4      698:
4ca4      699:
4ca4 2044da 700:      jsr    zfr0
4ca7 adb045 701:      lda    heads
4caa 85d4 702:      sta    fr0
4cac adb145 703:      lda    heads+1
4caf 85d5 704:      sta    fr0+1
4cb1 20aad9 705:      jsr    ifp
4cb4 a900 706:      lda    #$00
4cb6 85f2 707:      sta    cix
4cb8 20b6dd 708:      jsr    fmove
4cbb adb245 709:      lda    sector_per_trac
4cbe 85d4 710:      sta    fr0
4cc0 adb345 711:      lda    sector_per_trac+1
4cc3 85d5 712:      sta    fr0+1
4cc5 a900 713:      lda    #$00
4cc7 85f2 714:      sta    cix
4cc9 20aad9 715:      jsr    ifp
4ccc 20dbda 716:      jsr    fmult
4ccf 20b6dd 717:      jsr    fmove
4cd2 adb445 718:      lda    percom_density
4cd5 85d4 719:      sta    fr0
4cd7 adb545 720:      lda    percom_density+1
4cda 85d5 721:      sta    fr0+1
4cdc a900 722:      lda    #$00
4cde 85f2 723:      sta    cix
4ce0 20aad9 724:      jsr    ifp
4ce3 20dbda 725:      jsr    fmult
4ce6 20d2d9 726:      jsr    fpi
4ce9 a5d4 727:      lda    fr0
4ceb a6d5 728:      ldx    fr0+1
4ced 8db645 729:      sta    real_sector_count
4cf0 8eb745 730:      stx    real_sector_count+1
4cf3 8db845 731:      sta    end_sector
4cf6 8eb945 732:      stx    end_sector+1
4cf9 60 733:      rts
4cfa      734:
4cfa      735:
4cfa      736:
4cfa      737:
4cfa      738:
4cfa 8655 739: gotoxy:      stx    85
4cfc 8454 740:      sty    84
4cfe 60 741:      rts
4cff      742:

```

```

4cff      743:
4cff      744: ;
4cff      745: ;-----
4cff      746: ; PRINT BYTE
4cff      747: ;-----
4cff      748: echo:
4cff 85b8  749: sprint:          sta  $b8
4d01 a904  750:          lda  #$04
4d03 8d4a03 751:          sta  $034a
4d06 86b6  752:          stx  $b6
4d08 84b7  753:          sty  $b7
4d0a a206  754:          ldx  #$06
4d0c bd01e4 755:          lda  $e400+1,x
4d0f 85b3  756:          sta  $b3
4d11 bd00e4 757:          lda  $e400,x
4d14 85b2  758:          sta  $b2
4d16 a5b8  759:          lda  $b8
4d18 20274d 760:          jsr  188da
4d1b a6b6  761:          ldx  $b6
4d1d a4b7  762:          ldy  $b7
4d1f a90c  763:          lda  #$0c
4d21 8d4a03 764:          sta  $034a
4d24 a5b8  765:          lda  $b8
4d26 60     766:          rts
4d27      767: ;
4d27 e6b2  768: 188da:      inc  $b2
4d29 6cb200 769:          jmp  ($b2)
4d2c 60     770:          rts
4d2d      771: ;-----
4d2d      772: ; GET KEY ROUTEEN
4d2d      773: ;this routeen get's one key and
4d2d      774: ;return it to the accumulator
4d2d      775: ;-----
4d2d      776: ;
4d2d 86b6  777: get_key:    stx  $b6
4d2f 84b7  778:          sty  $b7
4d31 20394d 779:          jsr  .a
4d34 a6b6  780:          ldx  $b6
4d36 a4b7  781:          ldy  $b7
4d38 60     782:          rts
4d39 ad25e4 783: .a:        lda  $e420+5
4d3c 48     784:          pha
4d3d ad24e4 785:          lda  $e420+4
4d40 48     786:          pha
4d41 60     787:          rts
4d42      788: ;
4d42      789: ;-----
4d42      790: ;CHANGES THE CHARTER TO UPCASE
4d42      791: ;-----
4d42      792: upcase:
4d42 c961  793:          cmp  #$61
4d44 9006  794:          bcc  .a
4d46 c97a  795:          cmp  #$7a

```



```

4d48 b002 796:          bcs  .a
4d4a e91f 797:          sbc  #$1f
4d4c      798: .a:
4d4c 60   799:          rts
4d4d      800: ;
4d4d      801: ;-----
4d4d      802: ;CHANGES THE CHARTER TO LOCASE
4d4d      803: ;-----
4d4d      804: locase:
4d4d c941 805:          cmp  #$41
4d4f 9006 806:          bcc  .a
4d51 c95a 807:          cmp  #$5a
4d53 b002 808:          bcs  .a
4d55 691f 809:          adc  #$1f
4d57      810: .a:
4d57 60   811:          rts
4d58      812: ;
4d58      813: ;-----
4d58      814: ;  ERROR NUMBER
4d58      815: ;this take's the value of the a reg
4d58      816: ;and put a hex number to the screen
4d58      817: ;-----
4d58      818: ;
4d58 00   819: h_cardd:      dc.b  $00
4d59      820: ;
4d59      821: ; print a hex word value
4d59      822: ; low byte in a reg x is high byte
4d59      823: ;
4d59 8d584d 824: h_card:          sta  h_cardd
4d5c 8a   825:          txa
4d5d 20674d 826:          jsr  drive_error
4d60 ad584d 827:          lda  h_cardd
4d63 20674d 828:          jsr  drive_error
4d66 60   829:          rts
4d67      830: ;
4d67      831: ;
4d67 48   832: drive_error:    pha
4d68 4a   833:          lsr
4d69 4a   834:          lsr
4d6a 4a   835:          lsr
4d6b 4a   836:          lsr
4d6c 20724d 837:          jsr  .a
4d6f 68   838:          pla
4d70 290f 839:          and  #$0f
4d72 c90a 840: .a:          cmp  #$0a
4d74 b004 841:          bcs  .b
4d76 0930 842:          ora  #$30
4d78 d002 843:          bne  .c
4d7a 6936 844: .b:          adc  #$36
4d7c 20ff4c 845: .c:          jsr  sprint
4d7f 60   846:          rts
4d80      847: ;
4d80      848: ;

```

```

4d80      849: ;-----
4d80      850: ; jsr printsi
4d80      851: ;this prints an inline string
4d80      852: ;   terminated by $ff
4d80      853: ;-----
4d80      854: ;   jsr prints
4d80      855: ;   dc.b "print this", $9B, $FF
4d80      856: echosi:
4d80 68    857: printsi:      pla
4d81 8d914d 858:              sta  pstr+1
4d84 68    859:              pla
4d85 8d924d 860:              sta  pstr+2
4d88 ee914d 861: prsl:        inc  pstr+1
4d8b d003   862:              bne  pstr
4d8d ee924d 863:              inc  pstr+2
4d90 adffff 864: pstr:        lda  $ffff
4d93 c9ff   865:              cmp  #$ff
4d95 f006   866:              beq  estri
4d97 20d64d 867:              jsr  fast_output
4d9a 4c884d 868:              jmp  prsl
4d9d ad924d 869: estri:       lda  pstr+2
4da0 48     870:              pha
4da1 ad914d 871:              lda  pstr+1
4da4 48     872:              pha
4da5 60     873:              rts
4da6      874: ;
4da6      875: ;-----
4da6      876: ;actual screen handler
4da6      877: ;screen offset definitions
4da6      878: ;-----
4da6 0000   879: scr_offset:   dc.w  0
4da8 2800   880:              dc.w  40
4daa 5000   881:              dc.w  80
4dac 7800   882:              dc.w 120
4dae a000   883:              dc.w 160
4db0 c800   884:              dc.w 200
4db2 f000   885:              dc.w 240
4db4 1801   886:              dc.w 280
4db6 4001   887:              dc.w 320
4db8 6801   888:              dc.w 360
4dba 9001   889:              dc.w 400
4dbc b801   890:              dc.w 440
4dbe e001   891:              dc.w 480
4dc0 0802   892:              dc.w 520
4dc2 3002   893:              dc.w 560
4dc4 5802   894:              dc.w 600
4dc6 8002   895:              dc.w 640
4dc8 a802   896:              dc.w 680
4dca d002   897:              dc.w 720
4dcc f802   898:              dc.w 760
4dce 2003   899:              dc.w 800
4dd0 4803   900:              dc.w 840
4dd2 7003   901:              dc.w 880

```

```

4dd4 9803 902:          dc.w 920
4dd6      903: ;
4dd6      904: fast_output:
4dd6 855a 905:          sta $5a
4dd8 ad0407 906:         lda sc.redirect
4ddb c902 907:          cmp #print_p
4ddd f01f 908:          beq .go_cio
4ddf      909: .leo:
4ddf adff02 910:         lda $02ff      ; CONTROL 1
4de2 c900 911:          cmp #$00
4de4 d0f9 912:          bne .leo
4de6      913:
4de6 a55a 914:         lda $5a
4de8 c920 915:          cmp #32
4dea 9004 916:          bcc .do_look
4dec c97d 917:          cmp #125
4dee 9013 918:          bcc .do_here
4df0 a000 919: .do_look:      ldy #0
4df2 b9f94e 920: .lookup:      lda .char_table,y
4df5 f00c 921:          beq .do_here
4df7 c55a 922:          cmp $5a
4df9 f003 923:          beq .go_cio
4dfb c8 924:          iny
4dfc d0f4 925:          bne .lookup
4dfe      926:
4dfe      927:
4dfe a55a 928: .go_cio:      lda $5a
4e00 4c094f 929:          jmp putlocal
4e03      930:
4e03      931:
4e03 a000 932: .do_here:      ldy #0
4e05 a55d 933:          lda $5d
4e07 915e 934:          sta ($5e),y
4e09 a55a 935:          lda $5a
4e0b c99b 936:          cmp #$9b
4e0d f044 937:          beq .docr
4e0f 20d84e 938: .notcr:      jsr .get_adr
4e12 a55a 939:          lda $5a
4e14 297f 940:          and #$7f
4e16 c920 941:          cmp #32
4e18 9007 942:          bcc .add64
4e1a c960 943:          cmp #96
4e1c 9009 944:          bcc .sub32
4e1e 4c2a4e 945:          jmp .asis
4e21 18 946: .add64:      clc
4e22 6940 947:          adc #64
4e24 4c2a4e 948:          jmp .asis
4e27 38 949: .sub32:      sec
4e28 e920 950:          sbc #32
4e2a 245a 951: .asis:      bit $5a
4e2c 1002 952:          bpl .xxlate
4e2e 0980 953:          ora #$80
4e30 a000 954: .xxlate:      ldy #0

```

```
4e32 915e 955:      sta  ($5e),y
4e34 e655 956:      inc  $55
4e36 e663 957:      inc  $63
4e38 a555 958:      lda  $55
4e3a c928 959:      cmp  #40
4e3c b019 960:      bcs  .next_row
4e3e c8    961:      iny
4e3f b15e 962:      lda  ($5e),y
4e41 855d 963:      sta  $5d
4e43 0980 964:      ora  #$80
4e45 aef002 965:      ldx  752
4e48 d002 966:      bne  .nocurs1
4e4a 915e 967:      sta  ($5e),y
4e4c e65e 968: .nocurs1:      inc  $5e
4e4e d002 969:      bne  .nooav
4e50 e65f 970:      inc  $5e+1
4e52 60    971: .nooav:      rts
4e53 a900 972: .docr:      lda  #0
4e55 8563 973:      sta  $63
4e57 a552 974: .next_row:   lda  $52
4e59 8555 975:      sta  $55
4e5b e654 976:      inc  $54
4e5d a454 977:      ldy  $54
4e5f c018 978:      cpy  #24
4e61 b013 979:      bcs  .scroll
4e63 20d84e 980:      jsr  .get_adr
4e66 a000 981:      ldy  #0
4e68 b15e 982:      lda  ($5e),y
4e6a 855d 983:      sta  $5d
4e6c aef002 984:      ldx  752
4e6f d004 985:      bne  .nocurs2
4e71 0980 986:      ora  #$80
4e73 915e 987:      sta  ($5e),y
4e75 60    988: .nocurs2:   rts
4e76      989:
4e76 c654 990: .scroll:    dec  $54
4e78 a558 991:      lda  $58
4e7a 8568 992:      sta  $68
4e7c 18    993:      clc
4e7d 6928 994:      adc  #40
4e7f 855e 995:      sta  $5e
4e81 a559 996:      lda  $58+1
4e83 8569 997:      sta  $68+1
4e85 6900 998:      adc  #0
4e87 855f 999:      sta  $5e+1
4e89 a000 1000:      ldy  #0
4e8b a203 1001:      ldx  #3
4e8d b15e 1002: .sclloop:   lda  ($5e),y
4e8f 9168 1003:      sta  ($68),y
4e91 c8    1004:      iny
4e92 d0f9 1005:      bne  .sclloop
4e94 e65f 1006:      inc  $5e+1
4e96 e669 1007:      inc  $68+1
```

```

4e98 ca 1008:      dex
4e99 d0f2 1009:      bne  .sloop
4e9b b15e 1010: .sloop2:      lda  ($5e),y
4e9d 9168 1011:      sta  ($68),y
4e9f c8 1012:      iny
4ea0 c098 1013:      cpy  #152
4ea2 90f7 1014:      bcc  .sloop2
4ea4 a558 1015:      lda  $58
4ea6 18 1016:      clc
4ea7 6998 1017:      adc  #920&$ff
4ea9 855e 1018:      sta  $5e
4eab a559 1019:      lda  $58+1
4ead 6903 1020:      adc  #920/256
4eaf 855f 1021:      sta  $5e+1
4eb1 a000 1022:      ldy  #0
4eb3 98 1023:      tya
4eb4 915e 1024: .clear:      sta  ($5e),y
4eb6 c8 1025:      iny
4eb7 c028 1026:      cpy  #40
4eb9 90f9 1027:      bcc  .clear
4ebb aef002 1028:      ldx  752
4ebe d006 1029:      bne  .nocurs3
4ec0 a980 1030:      lda  #$80
4ec2 a455 1031:      ldy  $55
4ec4 915e 1032:      sta  ($5e),y
4ec6 a900 1033: .nocurs3:      lda  #0
4ec8 855d 1034:      sta  $5d
4eca a55e 1035:      lda  $5e
4ecc 18 1036:      clc
4ecd 6555 1037:      adc  $55
4ecf 855e 1038:      sta  $5e
4ed1 a55f 1039:      lda  $5e+1
4ed3 6900 1040:      adc  #0
4ed5 855f 1041:      sta  $5e+1
4ed7 60 1042:      rts
4ed8 a554 1043: .get_adr:      lda  $54
4eda 0a 1044:      asl
4edb a8 1045:      tay
4edc a558 1046:      lda  $58
4ede 18 1047:      clc
4edf 79a64d 1048:      adc  scr_offset,y
4ee2 855e 1049:      sta  $5e
4ee4 a559 1050:      lda  $58+1
4ee6 79a74d 1051:      adc  scr_offset+1,y
4ee9 855f 1052:      sta  $5e+1
4eeb a55e 1053:      lda  $5e
4eed 18 1054:      clc
4eee 6555 1055:      adc  $55
4ef0 855e 1056:      sta  $5e
4ef2 a55f 1057:      lda  $5e+1
4ef4 6900 1058:      adc  #0
4ef6 855f 1059:      sta  $5e+1
4ef8 60 1060:      rts

```

```
4ef9      1061: .char_table:
4ef9 1b1c1d 1062:          dc.b  27,28,29,30,31,125,126,127
4f01 9c9d9e 1063:          dc.b  156,157,158,159,253,254,255
4f09      1064:
4f09      1065:
4f09 a20b 1066: putlocal:      ldx   #11
4f0b 8e4203 1067:          stx   $0342
4f0e a200 1068:          ldx   #0
4f10 8e4803 1069:          stx   $0348
4f13 8e4903 1070:          stx   $0349
4f16 4c56e4 1071:          jmp   $e456
4f19      1072: ;
4f19      1073:
4f19      1074:
4f19      1075:
4f19      1076: pr_byte:
4f19 a200 1077:          ldx   #$00
4f1b      1078: pr_card:
4f1b 85d4 1079:          sta   $d4
4f1d 86d5 1080:          stx   $d4+1
4f1f a900 1081:          lda   #$00
4f21 85f2 1082:          sta   $f2
4f23 20aad9 1083:          jsr   $d9aa
4f26 20e6d8 1084:          jsr   $d8e6
4f29 a000 1085:          ldy   #$00
4f2b      1086: .aprbloop:
4f2b b1f3 1087:          lda   ($f3),y
4f2d 08 1088:          php
4f2e 297f 1089:          and   #$7f
4f30 c8 1090:          iny
4f31 20ff4c 1091:          jsr   echo
4f34 28 1092:          plp
4f35 10f4 1093:          bpl   .aprbloop
4f37 60 1094:          rts
4f38      1095:
4f38      1096: ;
4f38      1097: ;
4f38      1098: ;~~~~~
4f38      1099: ; help menu section
4f38      1100: ;~~~~~
4f38      1101: help_menu:
4f38      1102:          .include  notice
5416      1103:
5416 60 1104:          rts
5417      1105: ;
5417      1106:
5417      1107: win_end:      ds.b  0
5417      1108: ;
5417      1109: ;
```

End assembly: no errors