



```

0000 1: ;-----
0000 2: ;          TCPIP Express!
0000 3: ;          Copyright (c) 2012 by Intergrated Logic Sy
0000 4: ;
0000 5: ;
0000 6: ;
0000 7: ;
0000 8: ;          Source Code is Copyright 1997 Steven J. Tuc
0000 9: ;
0000 10: ; Converted to xasm  2/9/2005  By Stephen J. Carden
0000 11: ;
0000 12: ;
0000 13: ;-----
0000 14: ;          Apetime.S
0000 15: ;          Written By Steven J. Tucker
0000 16: ;          Apetime.com  A status file for the multiple
0000 17: ;          ~~~~~
0000 18: ;
0000 19: ;
0000 20: ;
0000 21: ;
0000 22: ; APETIME Time Downloader
0000 23: ;
0000 24: ; (c)1997 Steven J Tucker
0000 25: ;
0000 26: ;          9731 Sunrise M-33
0000 27: ;          North Royalton, Ohio  44133
0000 28: ;          440-237-5308
0000 29: ;
0000 30: ; Uses the spartados comtab to set the time and date via t
0000 31: ; under the OS that are installed by Spartados and Sparta
0000 32: ; should work in any version of Spartados.  Routines fot
0000 33: ; with memory under the OS and those without are included
0000 34: ;
0000 35: ; The programmed time is downloaded from the attached APE
0000 36: ; is not present, the loader complains & exits without di
0000 37: ;
0000 38: ;-----
0000 39: ; Notes: o This source code MAY NOT be placed for download
0000 40: ;        o "mea" is a macro that loads the address of the
0000 41: ;        into the pointer specified by the second field.
0000 42: ;-----
0000 43: ; Assembler:  MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 44: ;          XASM  st and IBM VERSIONS
0000 45: ;-----
0000 46: ;=====
0000 47:
0000 48:
0000 49: com_file_name:      .macro
0000 50: ;          dc.b  12,"filename.ext",$9b
0000 51: ;          dc.b  12,"Ape_hnd.com ", $9b
0000 52: ;          .endm
0000 53:

```

```

0000 54:
0000 55: ;~~~~~
0000 56: ; File revision history. Version Number in hex
0000 57: ;
0000 58: file_ver:      equ $10
0000 59: ;~~~~~
0000 60: ; the Month in dec for the first time this revision
0000 61: ;was compiled
0000 62: ;
0000 63: c_month:      equ 10
0000 64: ;~~~~~
0000 65: ; the day in dec for the first time this revision
0000 66: ;was compiled
0000 67: c_day:      equ 11
0000 68: ;~~~~~
0000 69: ; the year in dec for the first time this revision
0000 70: ;was compiled
0000 71: c_year:      equ 2012
0000 72: ;~~~~~
0000 73: ; Type of compiler used in program
0000 74: ;
0000 75: ; 0 = unknown Compiler
0000 76: ; 1 = xasm
0000 77: ; 2 = mac_65
0000 78: ; 3 = basic
0000 79: ; 4 = compiled basic xl
0000 80: ; 5 = C65
0000 81: ; 6 = Action
0000 82: ;
0000 83: ; We can add more as time goes on!
0000 84: ;
0000 85: ;
0000 86: xasm:      equ 1
0000 87: mac_65:    equ 2
0000 88: basic:     equ 3
0000 89: basicxl:   equ 4
0000 90: c65:      equ 5
0000 91: action:    equ 6
0000 92:
0000 93: file_compiler: equ xasm
0000 94: ;~~~~~
0000 95: ; is this relocatable code ?
0000 96: ; anyother value other than 1 or 2 would be unknown
0000 97: r..yes:    equ 1
0000 98: r..no:     equ 2
0000 99: ;
0000 100: ;~~~~~
0000 101: ; Gotta define if it can be relocated
0000 102: l_relocatable: equ r..yes
0000 103: ;
0000 104: ;
0000 105: ;~~~~~
0000 106: ; Gotta define if it can be relocated

```

```

0000    107: r..crl:          equ   $7d
0000    108: r..crlf:         equ   $9b
0000    109: r..space:       equ   $20
0000    110:
0000    111: l_frstscreenbyte:    equ   r..crl
0000    112: ;~~~~~
0000    113: ; The language the output file is in.
0000    114: ;
0000    115: ;
0000    116: ; 0 =   Undefined
0000    117: ; 1 =   English
0000    118: ; 2 =   German
0000    119: ;
0000    120: r..language:        equ   1
0000    121: ;~~~~~
0000    122: ;
0000    123: ;
0000    124: ;           program constants
0000    125: ;           ~~~~~
0000    126: ;
0000    127: dateoff:          equ   $000D ;; DATER offset
0000    128:
0000    129:
0000    130:                .include   equates
0000    131:                .include   globals
0000    132:                .include   macros
0000    133:
0000    134:                .org   $4000
0000    135:
0000    136: win_start:
0000    137: header_info:
0000    138:                .include   header      ; t
42a7 20ae42 139:                jsr   get_xsio      ; G
42aa 203843 140:                jsr   Ape_time
42ad 60    141:                rts
42ae      142:
42ae      143:
42ae      144:
42ae a50a 145: get_xsio:      LDA   COMTAB          ; c
42b0 38    146:                SEC
42b1 e90a 147:                SBC   #low lsio      ; T
42b3 8dc442 148:                STA   XSIO+1          ; T
42b6 a50b 149:                LDA   COMTAB+1        ; N
42b8 e900 150:                SBC   #high lsio     ; S
42ba 8dc542 151:                STA   XSIO+2          ; p
42bd a96c 152:                lda   #$6c            ; h
42bf 8dc342 153:                sta   xsio
42c2 60    154:                rts
42c3      155:
42c3 4c59e4 156: XSIO:          jmp   siov
42c6      157:
42c6 00    158: hold_stack:    dc.b   0
42c7      159:

```

```

42c7      160: ;
42c7      161: ;~~~~~
42c7      162: ;load the acc with the flag you want to set
42c7      163: ;   lda  #Value
42c7      164: ;   jsr  set_hw
42c7      165: ;~~~~~
42c7      166:
42c7      167: set_hw:
42c7 8ddf42 168:           sta  .hold
42ca 20e042 169:           jsr  os_off
42cd      170:           mea  int.hardware,s_ptr
42cd a9ed   ----:       lda  #low int.hardware
42cf 8582   ----:       sta  s_ptr
42d1 a9ff   ----:       lda  #high int.hardware
42d3 8583   ----:       sta  s_ptr + 1
42d5 addf42 171:           lda  .hold
42d8 20f642 172:           jsr  setlvl
42db 20eb42 173:           jsr  os_on
42de 60     174:           rts
42df      175:
42df 00     176: .hold:           dc.b  0
42e0      177:
42e0      178:
42e0      179: ;~~~~~
42e0      180: ; OS on and off ROUTEEN
42e0      181: ;~~~~~
42e0      182: ;
42e0      183:
42e0      184: os_off:
42e0 78     185:           SEI
42e1 ad01d3 186:           LDA  PORTB
42e4 29fe   187:           AND  #$FE
42e6 8d01d3 188:           STA  PORTB
42e9 58     189:           CLI
42ea 60     190:           RTS
42eb      191: ;
42eb      192: os_on:
42eb 78     193:           SEI
42ec ad01d3 194:           LDA  PORTB
42ef 0901   195:           ORA  #1
42f1 8d01d3 196:           STA  PORTB
42f4 58     197:           CLI
42f5 60     198:           RTS
42f6      199:
42f6      200:
42f6      201:
42f6      202:
42f6      203: ;   sets the bit specified (in the A register) of the se
42f6      204: ;   level pointed to by S_PTR to a TRUE value.
42f6      205: ;   ~~~~~
42f6      206:
42f6 8c3243 207: setlvl:           sty  .tempy
42f9 8e3343 208:           stx  .tempx

```

```

42fc c900 209:      cmp    #0          ;se
42fe f031 210:      beq    .return        ;ye
4300 a8 211:      tay          ;:-
4301 a980 212:      lda    #128
4303 8d3443 213:      sta    .temp          ;in
4306 a900 214:      lda    #0
4308 8d3543 215:      sta    .temp+1
430b 8d3643 216:      sta    .temp+2
430e 8d3743 217:      sta    .temp+3
4311 88 218:      dey
4312 f011 219:      beq    .mask          ;if
4314 4e3443 220: .shloop:      lsr    .temp          ;sh
4317 6e3543 221:      ror    .temp+1
431a 6e3643 222:      ror    .temp+2
431d 6e3743 223:      ror    .temp+3
4320 88 224:      dey
4321 d0f1 225:      bne    .shloop
4323 a000 226:      ldy    #0
4325 b93443 227: .mask:      lda    .temp,y          ;no
4328 1182 228:      ora    (s_ptr),y
432a 9182 229:      sta    (s_ptr),y
432c c8 230:      iny
432d c004 231:      cpy    #4
432f d0f4 232:      bne    .mask
4331 60 233: .return:      rts
4332 234:
4332 00 235: .tempy:      dc.b 0
4333 00 236: .tempx:      dc.b 0
4334 000000 237: .temp:      dc.b 0,0,0,0
4338 238:
4338 239:
4338 240:
4338 241:
4338 242:
4338 243: ;~~~~~
4338 244: ; Ape time Source
4338 245: ;~~~~~
4338 246:
4338 247: Ape_time:
4338 206944 248:      jsr    printsi          ; d
433b 9b9b 249:      dc.b $9b,$9b
433d 50756c 250:      dc.b "Pulling Time/Date from APE interfa
4364 415045 251:      dc.b "APE Interface is Copyright",$9b
437f 252:
437f 313939 253:      dc.b "1997 Steven J. Tucker",$9b,-1
4396 254:
4396 202f44 255:      jsr    gettime
4399 302d 256:      bmi    tranerr          ; e
439b 205844 257:      jsr    settime          ; s
439e 258:
439e a911 259:      lda    #G_Ape
43a0 20c742 260:      jsr    set_hw
43a3 261:

```

```

43a3 206944 262:      jsr  printsi
43a6 9b 263:      dc.b  $9b
43a7 d4e9ed 264:      dc.b  <+128>"Time/Date"
43b0 205570 265:      dc.b  " Updated from IBM PC", $9b, $9b, -1
43c7 60 266:      rts          ; e
43c8 267:
43c8 268: tranerr:
43c8 269:
43c8 206944 270:      jsr  PrintsI
43cb 9b 271:      dc.b  $9b
43cc c4eff7 272:      dc.b  <+128>,"Download failed!"
43dc 9b2043 273:      dc.b  $9b," Check Your cables and your",$
43f9 536f66 274:      dc.b  "Software on the IBM PC",$9b
4410 52756e 275:      dc.b  "Run 'Apetime' to try again!"
442b 276:
442b 277:
442b 9b9bff 278:      dc.b  $9b, $9b, -1
442e 60 279:      rts          ; e
442f 280:
442f 281: ; ~~~~~
442f 282: ;
442f 283: ;      Attempt to download time from APE.
442f 284: ; ~~~~~
442f 285:
442f a20b 286: gettime:  ldx  #$0B          ; setup the
4431 bd4c44 287: setloop:  lda  cmdtab,x      ;
4434 9d0003 288:          sta  $300,x        ;
4437 ca 289:      dex          ;
4438 10f7 290:      bpl  setloop        ;
443a 18 291:      clc          ; download
443b a50a 292:      lda  comtab          ; calculate
443d 690d 293:      adc  #dater          ;
443f 8d0403 294:      sta  $304            ;
4442 a50b 295:      lda  comtab+1        ;
4444 6900 296:      adc  #$00            ;
4446 8d0503 297:      sta  $305            ;
4449 4cc342 298:      jmp  xsio            ; get time
444c 299:
444c 450193 300: cmdtab:  dc.b  $45, $01, $93, $40, $00, $00, $00, $00
4458 301:
4458 302:
4458 303: ; ~~~~~
4458 304: ;
4458 305: ; Enters spartados kernel routine to set time.
4458 306: ; ~~~~~
4458 307:
4458 ad01d3 308: settime:  lda  portb        ; get into
445b 48 309:      pha          ;
445c 29fe 310:      and  #$FE          ;
445e 8d01d3 311:      sta  portb        ;
4461 20c3ff 312:      jsr  vsettd        ; enter spa
4464 68 313:      pla          ;
4465 8d01d3 314:      sta  portb        ; restore O

```

```

4468 60 315:      rts          ; exit
4469      316:
4469      317: ;
4469      318: ;-----
4469      319: ;prints Routeen macro
4469      320: ;this is a slow print to the screen
4469      321: ;usage jsr prints
4469      322: ; .byte $9b," string to be printed",$ff
4469      323: ;-----
4469      324: ;
4469 68 325: prints:      pla
446a 8d7a44 326:      sta pstr+1
446d 68 327:      pla
446e 8d7b44 328:      sta pstr+2
4471 ee7a44 329: prsl:   inc pstr+1
4474 d003 330:      bne pstr
4476 ee7b44 331:      inc pstr+2
4479 adffff 332: pstr:   lda $ffff
447c c9ff 333:      cmp #$ff
447e f006 334:      beq estri
4480 208f44 335:      jsr echo
4483 4c7144 336:      jmp prsl
4486 ad7b44 337: estri:  lda pstr+2
4489 48 338:      pha
448a ad7a44 339:      lda pstr+1
448d 48 340:      pha
448e 60 341:      rts
448f      342: ;
448f      343: ;
448f      344: ;-----
448f      345: ; PRINT BYTE
448f      346: ;-----
448f      347: echo:
448f 85b8 348: sprint:      sta $b8
4491 a904 349:      lda #$04
4493 8d4a03 350:      sta $034a
4496 86b6 351:      stx $b6
4498 84b7 352:      sty $b7
449a a206 353:      ldx #$06
449c bd01e4 354:      lda $e400+1,x
449f 85b3 355:      sta $b3
44a1 bd00e4 356:      lda $e400,x
44a4 85b2 357:      sta $b2
44a6 a5b8 358:      lda $b8
44a8 20b744 359:      jsr .a
44ab a6b6 360:      ldx $b6
44ad a4b7 361:      ldy $b7
44af a90c 362:      lda #$0c
44b1 8d4a03 363:      sta $034a
44b4 a5b8 364:      lda $b8
44b6 60 365:      rts
44b7      366: ;
44b7 e6b2 367: .a:      inc $b2

```



```
44b9 6cb200 368:          jmp  ($b2)
44bc 60      369:          rts
44bd      370:
44bd      371:
44bd      372:
44bd      373:
44bd      374:
44bd      375: Data_buffer:      ds.b  128
453d      376:
453d      377: win_end:
453d      378:      .end
```

End assembly: no errors