```
; ---------------------------------------------------------------------------------------------
;  Auto-set time from Ultimate 64 or 1541 UII+.
; ---------------------------------------------------------------------------------------------
DEBUG           =       0
.if             Pass1
                .include geosSym
                .include geosMac
                .include ultimate.inc
.endif
; ---------------------------------------------------------------------------------------------
                lda     #2              ;50% stipple
                jsr     SetPattern
                LoadB   r2L,0
                LoadB   r2H,199
                LoadW   r3,0
                LoadW   r4,319
                jsr     Rectangle       ;clear screen
                jsr     reset           ;start from clean slate
                jsr     ultProbe
                bcc     10$
                LoadW   r0,errorDB      ;device not found
                jsr     DoDlgBox
                bra     20$
10$             jsr     setTime
20$             jmp     EnterDeskTop
; ---------------------------------------------------------------------------------------------
;  Probe for presence of an Ultimate device.
;               pass:   nothing
;               return: carry clear if device present, set otherwise
; ---------------------------------------------------------------------------------------------
ultProbe:       jsr     enableIO
                lda     CMD_REG
                pha
                jsr     restoreIO
                pla
                cmp     #$c9            ;default read value
                bne     10$
                clc
                rts
10$             sec
                rts
```

```
; ---------------------------------------------------------------------------------------------
;  Set GEOS time from Ultimate device.
; ---------------------------------------------------------------------------------------------
setTime:        ldx     #<getTime
                ldy     #>getTime
                jsr     sendcmd
                lda     #19             ;yyyy/mm/dd hh:mm:ss
                jsr     drain
                ldx     #2
                jsr     cvtToBin
                sta     uYear
                ldx     #5
                jsr     cvtToBin
                sta     uMonth
                ldx     #8
                jsr     cvtToBin
                sta     uDay
                ldx     #11
                jsr     cvtToBin
                jsr     cvtMil
                sta     uHour
                jsr     cvtToCIA
                sta     cHour
                ldx     #14
                jsr     cvtToBin
                sta     uMinutes
                ldx     #14
                jsr     cvtToBCD
                sta     cMinutes
                ldx     #17
                jsr     cvtToBin
                sta     uSeconds
                ldx     #17
                jsr     cvtToBCD
                sta     cSeconds
                php
                sei
                MoveB   uYear,year
                MoveB   uMonth,month
                MoveB   uDay,day
                MoveB   uHour,hour
                MoveB   uMinutes,minutes
                MoveB   uSeconds,seconds
                plp                     ;enable/restoreIO also does php
                jsr     enableIO
                php
                sei
                lda     $dc0f           ;CIA #1
                and     #$7f            ;set time, not alarm
                sta     $dc0f
;               Note that writing to the hours register stops the TOD clock,
;               and writing to the tenths-of-second register restarts it.
                MoveB   cHour,$dc0b     ;this stops the clock!
                MoveB   cMinutes,$dc0a
                MoveB   cSeconds,$dc09
                MoveB   0,$dc08         ;this restarts the clock
                plp
                jsr     restoreIO
                rts
```

```
; ---------------------------------------------------------------------------------------------
; Convert two-digit ASCII number to binary.
;               pass:       .X pointer into dataBuf
;               return:     .A binary number
; ---------------------------------------------------------------------------------------------
cvtToBin:       lda         dataBuf,x
                and         #$0f
                sta         temp
                asl         a               ;X 2
                asl         a               ;X 4
                asl         a               ;X 8
                clc
                adc         temp            ;X 9
                adc         temp            ;X 10
                sta         temp
                inx
                lda         dataBuf,x
                and         #$0f
                adc         temp
                rts
; ---------------------------------------------------------------------------------------------
;  Convert military hours in binary to AM/PM
;               pass:       .A binary hours in military format (0-23)
;               return:     .A binary hours; carry set if PM, clear otherwise
;               note:       Reverse AM/PM flag for 0h and 12h.
; ---------------------------------------------------------------------------------------------
cvtMil:         clc                         ;AM
                tay
                bne         10$
                lda         #12             ;0h is 12:00 AM
                sec                         ;reverse AM/PM flag
                bcs         30$
10$             cmp         #12
                bne         20$
                clc                         ;reverse AM/PM flag
                bcc         30$
20$             cmp         #13
                bcc         30$             ;01:00 - 11:00 is AM
                sbc         #12             ;e.g. 13:00 is 01:00 PM
30$             rts
; ---------------------------------------------------------------------------------------------
; Convert hours to CIA TOD format.
;               pass:       .A hours (call cvtToBin, cvtMil)
;                           carry set if PM, clear if AM
;               return:     hours in format expected by TOD clock
; ---------------------------------------------------------------------------------------------
cvtToCIA:       bcc         10$
                ldy         #$80            ;AM/PM flag
                bne         20$
10$             ldy         #0
20$             sty         temp
                cmp         #10
                bcc         30$
                tay
                lda         #$10            ;10 in BCD
                ora         temp
                sta         temp
                tya
                sec
                sbc         #10             ;one’s place
30$             ora         temp
                rts
```

```
; ---------------------------------------------------------------------------------------------
; Convert two-digit ASCII number to BCD.
;                   pass:       .X pointer into dataBuf
;                   return:     .A BCD number
; ---------------------------------------------------------------------------------------------
cvtToBCD:   lda     dataBuf,x
            and     #$0f
            asl     a
            asl     a
            asl     a
            asl     a
            sta     temp
            inx
            lda     dataBuf,x
            and     #$0f
            ora     temp
            sta     temp
            rts
; ---------------------------------------------------------------------------------------------
;  Send command and params to Ultimate.
;                   pass:       .X/.Y address of command (first byte is length)
;                   return:     carry set on error, clear otherwise
; ---------------------------------------------------------------------------------------------
sendcmd:    jsr     enableIO
            stx     a0L
            sty     a0H
            ldy     #0
            lda     (a0),y
            sta     length
10$         iny
            lda     (a0),y
            sta     CMD_REG
            cpy     length
            bne     10$
            lda     #PUSH_CMD
            sta     CTRL_REG
            lda     STAT_REG
            and     #CMD_ERR
            beq     20$
            lda     #CLR_ERR        ;clear error state
            sta     CTRL_REG
            jsr     restoreIO
            sec
            rts
20$         jsr     restoreIO
            clc
            rts
```

```
; ------------------------------------------------------------------------------------------
;  Drain data and status queues and print to screen.
;                   pass:       .A number of data bytes to save
;                   return:     dataBuf: data, statCode: status code
; ------------------------------------------------------------------------------------------
drain:              sta         dataSize
                    lda         #0
                    sta         dataNdx
                    jsr         waitbusy
.if                 DEBUG
                    LoadW       r11,3
                    MoveB       ypos,r1H
.endif
                    jsr         enableIO
10$                 lda         CTRL_REG
                    and         #DATA_AVAIL
                    beq         20$
                    lda         DATA_REG
                    ldx         dataNdx
                    cpx         dataSize
                    bcs         15$
                    sta         dataBuf,x
                    inc         dataNdx
15$
.if                 DEBUG
                    pha
                    jsr         restoreIO
                    pla
                    jsr         PutChar
                    jsr         enableIO
.endif
                    bra         10$
20$                 ldx         #0
                    stx         statNdx
.if                 DEBUG
                    LoadW       r11,3
                    AddVB       9,ypos              ;new line
                    MoveB       ypos,r1H
.endif
30$                 lda         CTRL_REG
                    and         #STAT_AVAIL
                    beq         50$
                    lda         STAT_REG
                    ldx         statNdx
                    cpx         #2                  ;only save status code
                    bcs         40$
                    sta         statCode,x
                    inc         statNdx
40$
.if                 DEBUG
                    pha
                    jsr         restoreIO
                    pla
                    jsr         PutChar
                    jsr         enableIO
.endif
                    bra         30$
50$
.if                 DEBUG
                    AddVB       9,ypos              ;new line
.endif
                    lda         #ACK_DATA
                    sta         CTRL_REG
                    jsr         restoreIO
                    rts
```

```
; ---------------------------------------------------------------------------------------------
;  Wait for Ultimate device not to be busy processing a command.
; ---------------------------------------------------------------------------------------------
waitbusy:       jsr     enableIO
10$             lda     CTRL_REG
                and     #CMDBUSY
                bne     10$
                jsr     restoreIO
                rts
; ---------------------------------------------------------------------------------------------
;  Reset Ultimate device.
; ---------------------------------------------------------------------------------------------
reset:          jsr     enableIO
                lda     #ABORT
                sta     CTRL_REG        ;abort command
10$             lda     CTRL_REG
                and     #ABRT_PND       ;wait for abort to finish
                bne     10$
                lda     #CLR_ERR        ;clear error flag
                sta     CTRL_REG
                jsr     restoreIO
                rts
; ---------------------------------------------------------------------------------------------
;  Enable access to I/O bank (or restore previous setting). Not re-entrant!
; ---------------------------------------------------------------------------------------------
enableIO:       php
                sei
                lda     $01
                sta     ioSave          ;save memory configuration
                and     #$f8
                ora     #$05            ;bank in I/O
                sta     $01
                plp
                rts
restoreIO:      php
                sei
                lda     ioSave
                sta     $01             ;restore previous state
                plp
                rts
```

```
; ------------------------------------------------------------------------------------------------
errorDB:        .byte       DEF_DB_POS | 1
                .byte       OK,DBI_X_2,DBI_Y_2
                .byte       DBTXTSTR,14,28
                .word       errorMsg
                .byte       0
errorMsg:       .byte       "UltimateTime: device not found.",0
; ------------------------------------------------------------------------------------------------
xpos:           .word       3
ypos:           .byte       23
length:         .byte       0
;first byte of each command is length
identify:       .byte       2,TGT_NET,IDENTIFY
getTime:        .byte       3,TGT_DOS1,GET_TIME,0
uYear:          .block      1
uMonth:         .block      1
uDay:           .block      1
uHour:          .block      1
uMinutes:       .block      1
uSeconds:       .block      1
cHour:          .block      1
cMinutes:       .block      1
cSeconds:       .block      1
ioSave:         .block      1
statCode:       .block      2
statNdx:        .block      1
dataBuf:        .block      254
dataSize:       .block      1
dataNdx:        .block      1
temp:           .block      1
```