

Burner Tape Transfer 1.0

```
;-----  
;Burner Tape Transfer 1.0  
;  
;(C) 2005 Nostalgia  
;  
;By Fungus / Nostalgia  
;-----  
  
    *= $0801  
  
    .byte $08,$0b,$d5,$07,$9e  
    .byte $32,$30,$36,$31,$00  
    .byte $00,$00  
  
cd    = $02  
  
    lda $ba  
    sta cd        ;save current  
                  ;disk drive  
  
    lda #$0b      ;setup  
    sta $d011  
    lda #$c1      ;no nmi  
    sta $0318  
    lda #$80  
    sta $0291     ;no char shift  
    lda #$17  
    sta $d018     ;this set  
    lda #$00  
    sta $9d       ;no msgs  
    sta $d020  
    sta $d021  
    lda #$2f      ;i/o config  
    sta $00       ;bit 4 = input  
    lda $01  
    and #$07      ;mem config  
    ora #$20      ;motor off  
    sta $01  
    lda #$00  
    sta $0289     ;no keys  
    sta $0286  
  
    lda #$93  
    jsr $ffd2     ;clr screen  
  
    ldx #$00      ;print msg 1  
pm1   lda $0b80,x  
      sta $0400,x
```

```
        lda #$05
        sta $d800,x
        inx
        cpx #$a0
        bne pm1

        lda #$1b      ;screen on
        sta $d011

wplay   lda $01      ;wait for play
        and #$10
        bne wplay

xl      ldx #$00
        lda loader1,x
        sta $04a0,x
        lda loader1+$0100,x
        sta $05a0,x
        lda loader1+$0200,x
        sta $06a0,x
        inx
        bne xl

        jmp $04a0    ;begin real loader

;-----
;burner tape routine

loader1 = *
        *= $04a0
        .offs loader1-*

moreload1
        cli
        jsr loadhead
        jsr namefile
        jsr idloader
moreload2
        jsr putindex
        jsr printload
        jsr printname
        jsr clrmem
        jsr load
        jsr save
        lda files
        beq moreload1
        bne moreload2

load    sei
        lda #$0b    ;screen off
        sta $d011
```

```
    lda #$80      ;threshold = $0180
    sta $dd04
    lda #$01
    sta $dd05
    lda #$19
    sta $dd0e
    lda #$00
    sta $fb      ;relocate load to
    lda #$08      ;$0800
    sta $fc

    lda #$07      ;motor on
    sta $01

restart ldy #$00
      jsr sync
      jsr fbyte
      sta $20      ;load addy low
      sta $c1
      jsr fbyte
      sta $21      ;load addy high
      sta $c2
      jsr fbyte
      sta $22      ;end addy low
      sta $c3
      jsr fbyte
      sta $23      ;end addy high
      sta $c4

floop  jsr fbyte
      sta ($fb),y
      inc $fb
      bne aload
      inc $fc
aload  inc $c1
      bne cend
      inc $c2
cend   lda $c1
      cmp $c3
      lda $c2
      sbc $c4
      bcc floop

exit1  bit v3exit
      jsr fbyte    ;v1/2 more files
      sta files
      jsr fbyte    ;v1/2 jump addy
      sta jaddy
      jsr fbyte
      sta jaddy+1
      lda #$27     ;motor off
      sta $01
```

```
        rts

v3exit  lda #$27      ;motor off
        sta $01
        dec files
        lda $03b3
        sta jaddy
        lda $03b6
        sta jaddy+1
        rts

sync    jsr fbit      ;sync
endian1 ror $bd
        lda $bd
sync1   = *+1
        cmp #$96
        bne sync
wsync   jsr fbyte
sync2   = *+1
        cmp #$96
        beq wsync
sync3   = *+1
        cmp #$81
        bne sync
        rts

fbyte   ldx #$08      ;get byte
gbyte   jsr fbit
endian2 ror $bd
        inc $d020
        dex
        bne gbyte
        lda $bd
        rts

fbit    lda #$10      ;get bit
gbit    bit $dc0d
        beq gbit
        lda $dd0d
        lsr a
        lda #$19
        sta $dd0e
        rts

files   .byte $00     ;for burner 3/4
jaddy   .byte $00,$00

;-----
;load header

loadhead
```

```
    lda #$01      ;open tape file
    tax
    tay
    jsr $ffba
    lda #$00
    jsr $ffbd
    lda #$00
    sta $c0
    sta $93
    jsr $f7d7     ;load header/file
    jsr $f84f
    lda $01      ;motor off
    ora #$20
    sta $01
    rts

;-----
;id loader type

idloader
    ldx #$51
headdec lda $0300,x ;decode header
    eor #$59
    sta $0300,x
    inx
    bne headdec

    lda $035e
    cmp #$a9
    beq version1
    cmp #$85
    beq version2
    cmp #$d0
    beq version3
    jmp loaderr ;unknown loader

version1
    lda $03c3     ;version 1 loader
    sta sync1     ;mods
    sta sync2
    lda $03ce
    sta sync3
    lda $03be
    sta endian1
    lda $03d7
    sta endian2
    rts

version2
    lda $03c5     ;version 2 loader
    sta sync1     ;mods
    sta sync2
```

```
    lda $03d0
    sta sync3
    lda $03c0
    sta endian1
    lda $03d9
    sta endian2
    rts
version3
    lda $03c4    ;version 3/4 loader
    sta sync1    ;mods
    sta sync2
    lda $03cf
    sta sync3
    lda $03bf
    sta endian1
    lda $03d8
    sta endian2
    lda #$4c
    sta exit1
    lda $03bb
    sta files
    rts

;-----
;save routine

save
    lda #$00    ;save file
    sta $d020   ;border = black
    lda #$1b
    sta $d011   ;screen on
    lda #$00
    sta $fb     ;init save addy
    lda #$08
    sta $fc
    jsr printsave
    lda $20     ;init end counter
    sta $c1
    lda $21
    sta $c2

    lda #$01    ;open file for
    ldx cd      ;write
    ldy #$01
    jsr $ffb8   ;setlfs
    lda #$10
    ldx #<filename
    ldy #>filename
    jsr $ffbd   ;setname
    jsr $ffc0   ;open
    ldx #$01
```

```

        jsr $ffc9    ;chkout
        lda $ac      ;start addy low
        jsr $ffa8    ;send
        lda $ad      ;start addy high
        jsr $ffa8    ;send

saveb   ldy #$00     ;save the file
        sei
        inc $01
        lda ($fb),y
        dec $01
        jsr $ffa8
        inc $d020
        dec $d020

        inc $fb
        bne asave
        inc $fc

asave   inc $c1
        bne bsave
        inc $c2

bsave   lda $c1
        cmp $c3
        bne saveb
        lda $c2
        sbc $c4
        bne saveb

        lda #$01
        jmp $ffc3    ;close the file

;-----
;utility subroutines

loaderr
        lda #$02     ;print tape load
        sta $d020    ;error msg
        ldx #$17

perr1   lda et1,x
        and #$3f
        sta $0478,x
        lda #$01
        sta $d878,x
        dex
        bpl perr1
        jmp *        ;endless loop

printload
        ldx #$07     ;print loading
ll      lda lt,x     ;text

```

```
        and #$3f
        sta $0478,x
        lda #$05
        sta $d878,x
        dex
        bpl ll
        rts

printsave
        ldx #$07
dl       lda st,x      ;print save text
        and #$3f
        sta $0478,x
        lda #$05
        sta $d878,x
        dex
        bpl dl
        rts

printname
        ldx #$0f      ;print name
ln       lda filename,x
        and #$7f
        sta filename,x
        and #$3f
        sta $0480,x
        dex
        bpl ln
        rts

putindex
        lda fileidx
        inc fileidx
        pha
        lsr a
        lsr a
        lsr a
        lsr a
        jsr convert1
        sta filename+$0e
        pla
        and #$0f
        jsr convert1
        sta filename+$0f
        rts

convert1 ora #$30
        cmp #$3a
        bcc convert2
        adc #$06
convert2 rts
```



```

clrmem      sei
            inc $01
            lda #$08
            sta clrit+$02
            lda #$00
            tax
clrit       sta $0800,x
            inc
            bne clrit
            inc clrit+$02
            bne clrit
            dec $01
            cli
            rts

namefile
            lda #$20
            ldx #$0f
clrname     sta filename,x
            dex
            bpl clrname
namer       ldy #$00      ;name file
            ldx #$00
dname       lda $0341,y
            beq skip
            cmp #$41      ;less than A
            bcc check2
            cmp #$5b      ;more than Z
            bcs skip
            bcc putname
check2      cmp #$30      ;less than 0
            bcc skip
            cmp #$3a      ;more than 9
            bcs skip
putname     sta filename,x
            inc
skip        iny
            cpy #$10
            bne dname
            rts

;-----
;Transfer Messages

fileidx     .byte $00

et1         .text "Load Error!      "
            .text "                "
lt          .text "Loading "
st          .text "Saving  "

```

filename .text "

From:

<http://www.codebase64.org/> - **Codebase 64 wiki**

Permanent link:

http://www.codebase64.org/doku.php?id=base:burner_tape_transfer_1.0

Last update: **2015-04-17 04:30**

