

Novaload Tape Transfer 1.0

```

;-----
;Novaload Tape Transfer 1.0
;
;(C) 2005 Nostalgia
;
;By Fungus / Nostalgia
;-----

        *= $1000

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      ;st clear

        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

pm1     ldx #$00      ;print msg 1
        lda $1300,x
        sta $0400,x
        lda #$05
        sta $d800,x
        inx
        cpx #$a0

```

```
        bne pm1

        lda #$1b
        sta $d011

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

        jsr lod          ;load header

        lda $01          ;motor off
        ora #$20
        sta $01

xl      ldx #$00
        lda loader1,x
        sta $04a0,x
        lda loader1+$0100,x
        sta $05a0,x
        lda loader1+$0200,x
        sta $06a0,x
        lda loader1+$0300,x
        sta $07a0,x
        inx
        bne xl

lod     jmp $04a0          ;begin real loader

        lda #$01          ;open tape file
        tax
        tay
        jsr $ffba
        lda #$00
        jsr $ffbd
        lda #$00
        sta $c0
        sta $93
        jsr $f7d7          ;load header/file
        jmp $f84f

;-----
;novaload tape routine

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

savename = $0341

moreload jsr clrmem      ;clear memory
```

```

        jsr putindex
        jsr printload
        jsr printname

        sei
        jsr i0345      ;loader
        lda $90
        beq loadok
        jmp loaderr

loadok  inc $a4
        bne nohi1
        inc $ad
nohi1  lda $a4      ;lo end
        sta $ae
        lda $ad      ;hi end
        sta $af
        lda #$01
        sta $ac
        lda #$08
        sta $ad
        inc header+$01
        jsr save

        ldx #$0f
        lda #$20
clrname sta savename,x
        dex
        bpl clrname

        lda #$99      ;change to sta
        sta mod2
        lda #<savename;after first
        sta mod1+1    ;file change name
        lda #>savename
        sta mod1+2
        lda #$11      ;filename 10 or <
        sta mod3+1
        lda #$b0      ;bne to bcs
        sta mod4
        jmp moreload

i0345  ;loader
        lda #$f4      ;timing
        sta $dc04
        lda #$81      ;$01f4+$80 = bit
        sta $dc05
        lda #$01      ;load addy
        sta $ac      ;$0801-$0100
        lda #$07
        sta $ad

```

```
        lda #$2f          ;io normal
        sta $00
        lda #$07          ;tape motor on
        sta $01

i0358   ldy #$00
        sty $a4          ;y index
        sty $a3          ;init chksum
        sty $90          ;init st
        lda #$05

i035e   jsr i03dd        ;get a bit
        bpl i035e        ;sync (#$80)
        jsr i03db        ;get a byte
        cmp #$aa        ;block id
        bne i035e
        jsr i03db        ;get a byte

mod3    cmp #$04        ;first file is
mod4    bne i035e        ;NOVA

        sta $b7          ;filename length

i0371   cpy $b7          ;filename present
        beq i037f        ;no name
        jsr i03d4        ;get filename
mod1    sta filename,y
mod2    lda $0480,y
        iny
        bne i0371

i037f   ldy #$00          ;get header
i0381   jsr i03d4
        sta header,y    ;6 bytes
        iny
        cpy #$06
        bne i0381
        ldy #$00
        lda header+$04
        sta $b0
        lda header+$05
        sta $b1

i038a   jsr i03d4        ;calc chksum+get
        ;byte

        cmp $a3
        bne i03ad        ;error
        lda $b1
        beq i03af        ;eof

        inc $ad          ;block hi
```

```

        dec $b1          ;blocks to load
        bne i03a1
        lda $b0
        beq i03af        ;eof
        sta $a4          ;y index

i03a1   jsr i03d4        ;get byte
        inc $01
        sta ($ac),y      ;store
        dec $01
        iny
        cpy $a4          ;y index
        bne i03a1
        beq i038a

i03ad   lda #$30         ;error
i03af   sta $90          ;eof

        lda #$27         ;motor off
        sta $01
        rts

i03d4   clc              ;calc checksum
        lda $a3
        adc $aa
        sta $a3

i03db   lda #$7f         ;get byte
i03dd   pha              ;get bit
        inc $d020
        ldx #$11
        lda #$10
i03e5   bit $dc0d
        beq i03e5
        lda $dc05
        stx $dc0e
        eor #$80
        asl a
        pla
        ror a
        bcs i03dd
        sta $aa
        rts

;-----
;save routine

save    lda #$00         ;save file
        sta $d020        ;border = black

```

```

jsr printsave

lda #$01      ;open file for
ldx cd        ;write
ldy #$01
jsr $ffbba   ;setlfs
lda #$10
ldx #$41     ;$0341 16 chars
ldy #$03
jsr $ffbd    ;setname
jsr $fffc0   ;open
ldx #$01
jsr $fffc9   ;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 $01
lda ($ac),y
dec $01
jsr $ffa8
inc $d020
dec $d020

inc $ac
bne b1
inc $ad
b1          lda $ad
cmp $af
bne saveb
lda $ac
cmp $ae
bne saveb

lda #$01
jmp $fffc3  ;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
ll      ldx #$07      ;print loading
        lda lt,x      ;text
        and #$3f
        sta $0478,x
        lda #$05
        sta $d878,x
        dex
        bpl ll
        rts

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

printname
ln      ldx #$0f      ;print name
        lda savename,x
        and #$7f
        sta savename,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 savename+$0e
        pla
        and #$0f
        jsr convert1
```

```
        sta savename+$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
        inx
        bne clrit
        inc clrit+$02
        bne clrit
        dec $01
        cli
        rts

;Transfer Messages

fileidx .byte $00

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

filename .text "                "
header  .byte $00,$00,$00,$00,$00,$00
```

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

Permanent link:
http://codebase64.org/doku.php?id=base:novaload_tape_transfer_1.0

Last update: **2015-04-17 04:33**

