

Flexloader 3.0

WARNING: This is NOT an IRQ loader, it's only stable without IRQ and screen off.

Non drive locking IRQ/FAST loader 1541 only

This source can be compiled with tass/taboo.

The drive initialize routine is to slow to run within a irq.

The original routine sets # $\$0f$ to $\$1c07$, the frequency of the stepper motor. This is unusually fast! According to the [this thread on CSDb](#), the absolute minimum is $\$18$, but it's safer to use values between $\$20$ and $\$28$.

```

;-----
; loader v3.0 (c) Jayce/Focus 1993!
; finally converted to turboassembler/windows march 2007!
;-----
      *= $6000
      jsr faster ;for faster acces to 1541
      jsr init   ;warm reset vectors
      lda #$00
      sta $d020
      sta $d011
      jsr fopen  ;open logical file

;-----
;-remove this sei if a irq is used   -
;-----
      sei

;-----

      jmp load   ;and go loading

;-----filename + lengte-----
fnln   .byte 16
fnm    .text "p>anim*"
;-----

fopen  lda fnln
       ldx #<fnm
       ldy #>fnm
       jsr $ffbd
       lda #1
       ldy #0
       ldx #8
       jsr $ffba
       jsr $ffc0
       bcc fopeno

```

```

fopener  lda #1
         jsr $ffc3
         jsr $ffcc

         jmp fopen;keep jumping on error

fopeno   ldx #1
         jsr $ffc6
         bcs fopener
         jsr $ffa5
         lda $90
         bne fopener
         lda #0
         sta getchara
         sta getcharb
         sta getcharc
         jsr insdisk
         lda #$08
         jsr $ffb1
         lda #$6f
         jsr $ff93
         ldy #$00
fopen1   lda fstt,y
         jsr $ffa8
         iny
         cpy #$20
         bne fopen1
         jsr $ffae
         lda #$01      ;reload address lo byte
         sta $ae
         lda #$08      ;reload address hi byte (here it's $0801)
         sta $af
         rts

getchara .byte 0
getcharb .byte 0
getcharc .byte 0

load     jsr getchar
         jsr getchar
load1    jsr getchar
         jsr putch
         lda $90
         beq load1
stop     jsr init
         jmp $080d ; jump adress after loading

getchar  lda getcharb
         bne getchar1

```

```

        lda #2
        sta getcharb
        jsr charin
        sta getchara
        bne getchar1
        jsr charin
        sta getcharc
getchar1 lda getchara
        bne getchar2
        lda getcharb
        cmp getcharc
        bne getchar2
        jsr charin
        ldx #$80
        stx $90
        rts
getchar2 lda #$00
        sta $90
        jsr charin
        inc getcharb
        inc $ac
        bne getchar3
        inc $ad
getchar3 rts

charin  ldx #$03
        lda #$0b
        sta $dd00
charin0 bit $dd00
        bpl charin

;-----
;-remove this if no irq is used      -
;-----
charin1 ; lda $d012
        ; cmp #$31 ;number of free rasterlines the higher the number, the
slower the loader
        ; bcs charin1
;-----

charin2 stx $dd00
        nop
        nop
        nop
        nop
        nop
        lda $dd00
        lsr a
        lsr a
        eor $dd00
        lsr a

```

```

        lsr a
        eor $dd00
        lsr a
        lsr a
        eor #3      ;$dd00 bankswitch value
        eor $dd00
        rts

putch   tax
        lda $af
        beq putche
        txa
        ldx #$00
        inc $01 ; use the Y to set $01 manually if needed
        sta ($ae,x)
        dec $01
        ;sta $d020 ;(flash aaahaaaaa!)
        inc $ae
        bne putche
        inc $af
putche  rts

fstt   .byte $4d,$2d,$45,$05,$02
        cli
        lda $f9
        asl a
        tax
        lda $06,x
        sta $08
        lda $07,x
        sta $09
        lda #$80
        sta $01
        jmp $030d

;insert the code into the 1541 #08
insdisk lda #$00
        sta $fb
        lda #$03
        sta $fc
        lda #<diskdat
        sta $fd
        lda #>diskdat
        sta $fe
insd0  lda #$08 ;<< device ID #08
        jsr $ffb1
        lda #$6f
        jsr $ff93
        lda #$4d

```

```

        jsr $ffa8
        lda #$2d
        jsr $ffa8
        lda #"w"
        jsr $ffa8
        lda $fb
        jsr $ffa8
        lda $fc
        jsr $ffa8
        lda #$20
        jsr $ffa8
        ldy #$00
insd1   lda ($fd),y
        jsr $ffa8
        iny
        cpy #$20
        bne insd1
        jsr $ffae
        clc
        lda $fd
        adc #$20
        sta $fd
        bcc insd2
        inc $fe
insd2   clc
        lda $fb
        adc #$20
        sta $fb
        cmp #$e0
        bcc insd0
        rts

```

;---a byte dump of the in-drive code ---

```

diskdat .byte $00,$00,$53,$54,$49,$20
        .byte $5b,$03,$a5,$00,$4c,$0a
        .byte $e6,$a5,$09,$a2,$01,$85
        .byte $09,$58,$20,$85,$03,$78
        .byte $c9,$02,$b0,$ec,$a0,$00
        .byte $ad,$00,$05,$d0,$03,$ac
        .byte $01,$04,$8c,$ff,$03,$a0
        .byte $02,$c9,$00,$d0,$01,$88
        .byte $20,$47,$03,$b9,$00,$05
        .byte $20,$47,$03,$c8,$cc,$ff
        .byte $03,$d0,$f4,$ad,$00,$05
        .byte $d0,$cf,$4c,$a9,$03,$78
        .byte $aa,$4a,$4a,$4a,$4a,$85
        .byte $85,$8a,$29,$0f,$aa,$bd
        .byte $c0,$03,$2c,$00,$18,$10
        .byte $fb,$a2,$10,$8e,$00,$18
        .byte $a6,$85,$2c,$00,$18,$30
        .byte $fb,$8d,$00,$18,$0a,$29

```

```

.byte $0f,$8d,$00,$18,$bd,$c0
.byte $03,$8d,$00,$18,$0a,$29
.byte $0f,$8d,$00,$18,$ea,$ea
.byte $a9,$00,$8d,$00,$18,$58
.byte $60,$a5,$01,$30,$fc,$48
.byte $a0,$00,$b9,$00,$04,$99
.byte $00,$05,$c8,$d0,$f7,$ad
.byte $00,$04,$f0,$0b,$85,$08
.byte $ad,$01,$04,$85,$09,$a9
.byte $80,$a2,$01,$85,$01,$68
.byte $60,$b9,$00,$05,$20,$47
.byte $03,$20,$42,$d0,$4c,$94
.byte $c1,$00,$00,$00,$00,$00
.byte $00,$00,$00,$00,$00,$00
.byte $0f,$07,$0d,$05,$0b,$03
.byte $09,$01,$0e,$06,$0c,$04
.byte $0a,$02,$08,$00

```

;-- a faster drive access stuff --

```

faster   lda #$08      ; Memory write to drive 8
        jsr $ffb1
        lda #$6f
        jsr $ff93
        lda #$4d      ; M
        jsr $ffa8
        lda #$2d      ; -
        jsr $ffa8
        lda #$57      ; W
        jsr $ffa8
        lda #$07      ; <$1c07, stepper motor frequency
        jsr $ffa8
        lda #$1c      ; >$1c07
        jsr $ffa8
        lda #$01      ; data length 1 byte
        jsr $ffa8
        lda #$0f      ; frequency = $0f
        jsr $ffa8
        jsr $ffae
        rts

```

init

```

sei
jsr $fd15
jsr $fda3
ldx #$00
jsr $e5aa
cli
rts

```

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

Permanent link:
https://codebase64.org/doku.php?id=base:flexloader_3.0

Last update: **2015-05-17 11:36**

