

ALPHABIT by nom de nom & shifty of TROPE ... a 64 byte C64 intro, runs for 0:35

```
02 08 01 00 00 9E 32 30
36 31 00 00 00 20 81 FF
C8 8C 12 D4 8C 14 D4 C8
8C 20 D0 AD 12 D0 9D F4
D3 8C 18 D4 D0 F5 8A 8E
0F D4 AE 1B D4 E0 F0 B0
F6 9D 90 05 9D 90 D9 AA
88 D0 E0 E8 E0 1B D0 DB
```

Load address of file: \$0802, BASIC region. BASIC actually starts at \$0801, but if there is any nonzero value in either \$0801 or \$0802 a program has a line number, so we can load from \$0802 and save a byte.

BASIC bootloader: Puts "0 SYS2061" into memory to start the machine code program at decimal 2016, \$80D.

Initialization:

```
jsr scinit ; returns with y=$84
iny        ; $85 works for the next two...
sty $d412  ; voice 3 noise waveform
sty $d414  ; voice 3 SR
iny        ; $86 works next; The low nybble just needs be $06
sty $d020  ; border to dark blue
```

Outer loop, first part:

```
lda $d012 ; get current raster line
sta $d3f4,x
sty $d418 ; put # of chars left to write in the volume register
bne raster
txa
```

Inner loop:

```
stx $d40f ; put the current letter, 1-26, in the freq register
ldx $d41b ; get random byte from SID voice 3
cpx #240 ; in the 240 character "block"?
bcs random ; if not, pick again
```

Outer loop, second part:

```
sta $590,x ; jam the current letter on screen
sta $d990,x ; make some colors with the value
tax
dey
bne raster
```

Final letter loop:

```
inx
cpx #27 ; have we gotten past 'Z'?
bne raster
```