

```

; -----
;
; Sprite-Scroller Routine
; -----
;
; coding: testicle/payday
; logo: fabu/payday
;
;
; contact and payday-releases:
; -----
;
; daniel@popelganda.de
; www.popelganda.de
;
;
; this source code is part of an intro, so many code is missing here,
; while only the interesting parts for the sprite scroller are shown.
; it shows how to use sprites for text scrolling, so the scroll text
; can easily be placed above pictures.
;
; this sourcecode is best view with the font "tahoma", font size 9.
; you can compile this code using the ACME crossassembler.
;
; the code was written with Relaunch64, the c64-crossassembler-tool
; for windows-pc. grab it at www.popelganda.de!
; -----
; -----

;-----
;----- Paragraph @Globale Variablen@ -----
;-----

spritexpos = 128
spriteypos = 140 ;sprite y-position
spritechar = $3300 ;here's the char located, that "rolls" into the
spritescoller
text = $3a00

;-----
;----- Paragraph @Includes@ -----
;-----

; init text pointer

lda #<text

```

```

        sta $50
        lda #>text
        sta $51

;-----
;----- Paragraph @clear sprite-memory@ -----
;-----

        ldx #0
        lda #0
.loop4   sta $3800,x
        inx
        bne .loop4
        ldx #0
        lda #0
.loop6   sta $3900,x
        inx
        cpx #64
        bne .loop6

        ldx #7
        lda #0
.loop5   sta spritechar,x
        dex
        bpl .loop5

        jsr sprscrollinit    ;jump to subroutine to initialize sprite
positions

        cli                ;cli, for this is the end of the setup, which is missing
here

                                ;(see comment on top for further information)
loading lda #0
        beq loading        ;endless branch, the "loading" pointer is changed
                                ;to a value of 1 when the user presses the space-bar
                                ;add something here to clear screen/move next demo
part/whatever

;-----
;----- Paragraph @init sprites above@ -----
;-----

!zone
sprscrollinit  lda #spritexpos
               sta $d000
               lda #spritexpos+24
               sta $d002
               lda #spritexpos+48

```

```

    sta $d004
    lda #spritexpos+72
    sta $d006
    lda #spritexpos+96
    sta $d008
    lda #spriteypos
    sta $d001
    sta $d003
    sta $d005
    sta $d007
    sta $d009
    lda #%00011111    ;switch on 5 sprites
    sta $d015
    lda #0
    sta $d01b
    sta $d01c
    lda #11
    sta $d027
    sta $d028
    sta $d029
    sta $d02a
    sta $d02b

```

```

;-----
;    sprites at $3800
;-----

    lda #224
    sta $07f8
    lda #225
    sta $07f9
    lda #226
    sta $07fa
    lda #227
    sta $07fb
    lda #228
    sta $07fc
    rts

```

```

;-----
;
;----- Paragraph @Sub-Route: Spritescrolling@ -----
;
;-----

```

```

    ;this is the main routine which is responsible for scrolling
    ;a text through sprites

```

```

!zone
spritescroll    dec .cnt+1
.cnt            lda #8            ;already 8 pixel moved?

```

```

        beq .neuchar          ;if yes, read in new char
        jmp .softscroll      ;else jump to the softscroller and return to
the main routine

.neuchar  ldy #0              ;read new char
         lda ($50),y         ;this is the text-pointer
         bne .undlos        ;end-sign?

;-----

        lda #<next
         sta jumper+1
         lda #>next
         sta jumper+2

;-----

         lda #0              ;if yes, reset text-vector
         sta $50
         lda #>text
         sta $51
         lda # $20

.undlos  clc                  ;clear carry-bit
         rol                  ;char-value * 8
         rol                  ;(this is the offset for the pixeldata of a char in
the charset)
         rol
         sta .loop2+1
         bcc .weiter
         inc .loop2+2

.weiter  ldx #7              ;read 8 bytes (one char from the charset)
.loop2   lda $3000,x         ;from charset-memory
         sta spritechar,x    ;and store to that memory-adress where the
char is located,
         dex                  ;that "roles" next into the spritescroll
         bpl .loop2

         lda #0              ;reset adresses
         sta .loop2+1
         lda # $30
         sta .loop2+2

         inc $50              ;increase scrolltext-counter
         lda $50
         bne .nixneu
         inc $51

.nixneu  lda #8              ;reset scrolltext-counter
         sta .cnt+1

```

```

.softscroll ldy #0
            ldx #0

;-----
;   move chars in sprites
;   to the left (soft-scrolling)
;-----

.loop1      clc
.origin     rol spritechar      ;"read" left bit of new sign
            rol $3902,x         ;move sprite-char - sprite5
            rol $3901,x
            rol $3900,x
            rol $38c2,x         ;move sprite-char - sprite4
            rol $38c1,x
            rol $38c0,x
            rol $3882,x         ;move sprite-char - sprite3
            rol $3881,x
            rol $3880,x
            rol $3842,x         ;move sprite-char - sprite2
            rol $3841,x
            rol $3840,x
            rol $3802,x         ;move sprite-char - spritel
            rol $3801,x
            rol $3800,x
            iny
            inc .origin+1       ;increase counter and set to next "pixel-row"
of that char
            txa
            clc:adc #3
            tax
            cpy #8
            bne .loop1
            lda #<spritechar    ;restore original value
            sta .origin+1
            rts

;-----
;
;----- Paragraph @Scrolltext@ -----
;
;-----

*= text

!ct scr
!tx "      out of the dark, into the blue... the symbol of our complex has
been banned to bytes."
!tx "      now we in - payday - proudly present you some oldschoool-stuff,
codename: press space to continue."
!tx "      this marvellous piece of art has a name! it's called..."

```

```
!tx "      "  
!byte 0
```

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

Permanent link:  
[https://codebase64.org/doku.php?id=base:scrolltext\\_using\\_sprites](https://codebase64.org/doku.php?id=base:scrolltext_using_sprites)

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

