

```
n "Rasterbars"  
t "G3_SymMacExt"
```

```
a "M. Kanet"  
f SYSTEM  
o LD_ADDR_SCRSAVER
```

```
i
```



```
c "ScrSaver64 V1.0"  
z $80 ;nur GEOS64 bei MP3-64
```

```
;*** ScreenSaver aufrufen.  
:MainInit      jmp      InitScreenSaver
```

```
;*** ScreenSaver installieren.  
; Laufwerk von dem ScreenSaver geladen wurde muß noch aktiv sein!  
; Rückgabe eines Fehlers im xReg ($00=Kein Fehler).  
; ACHTUNG! Nur JMP-Befehl oder "LDX #$00:RTS", da direkt im Anschluß  
; der Name des ScreenSavers erwartet wird! (Addr: G3_ScrSave +6)  
:InstallSaver   ldx      #$00  
               rts
```

```
;*** Name des ScreenSavers.  
; Direkt nach dem JMP-Befehl, da über den GEOS.Editor der Name  
; an dieser Stelle ausgelesen wird.  
; Der Name muss mit dem Dateinamen übereinstimmen, da der  
; Bildschirmschoner über diesen Namen beim Systemstart geladen wird.  
:SaverName      b "Rasterbars",NULL
```

;*** ScreenSaver aufrufen.

:InitScreenSaver	php		;IRQ sperren.
	sei		;Screener läuft in der MainLoop!
::51	ldx	##\$1f	;Register ":r0" bis ":r3"
	lda	r0L,x	;zwischenspeichern.
	pha		
	dex		
	bpl	:51	
	jsr	DoSaverJob	;Bildschirmschoner aktivieren.
	lda	##%01000000	;Bildschirmschoner neu starten.
	sta	Flag_ScrSaver	
::52	ldx	##\$00	;Register ":r0" bis ":r3"
	pla		;zurückschreiben.
	sta	r0L,x	
	inx		
	cpx	##\$20	
	bne	:52	
	sei		;IRQ abschalten.
	ldx	CPU_DATA	;CPU-Register speichern und
	lda	##\$35	;I/O-Bereich einblenden.
	sta	CPU_DATA	
::53	lda	##\$00	
	sta	\$dc00	;Tastenregister aktivieren.
	lda	\$dc01	;Tastenstatus einlesen.
	eor	##\$ff	;Taste noch gedrückt ?
	bne	:53	;Ja, Warteschleife...
	stx	CPU_DATA	;CPU-Register zurücksetzen.
	plp		;IRQ zurücksetzen und
	rts		;Ende...

;*** Bildschirmschoner-Grafik.

:DoSaverJob	lda	CPU_DATA	;I/O-Bereich aktivieren.
	pha		
	lda	##\$35	
	sta	CPU_DATA	
	lda	\$d011	;Bildschirm abschalten.
	and	##%11101111	
	sta	\$d011	

```

lda    $d015    ;StriteOn-Register speichern.
pha
lda    #$00     ;Sprites abschalten.
sta    $d015

lda    $d020    ;Rahmenfarbe sichern.
pha
lda    #$00     ;Rahmenfarbe löschen.
sta    $d020

::51    lda    #$00     ;Warten bis keine Taste gedrückt.
        sta    $dc00
        lda    $dc01
        eor    #$ff
        bne    :51

::52    ldx    #$00     ;Zeiger auf erste Rasterzeile.
        lda    CurColTab
        tay
        clc
        adc    #$02
        cmp    #$08
        bcc    :53
        lda    #$00
::53    sta    CurColTab

        lda    ColTabVec +0,y
        sta    r0L
        lda    ColTabVec +1,y
        sta    r0H
        tya
        lsr
        tay
        lda    ColTabLen +0,y
        sta    r1L
::54    cpx    $d012    ;Rasterzeile erreicht ?
        bne    :54     ;Nein, warten.
        stx    r1H     ;Rasterzeile merken.

::55    cpx    $d012    ;Am Beginn der nächsten Zeile ?
        beq    :55     ;Nein, warten.

```

```

::56      ldy      #$00      ;Farbbalken erzeugen.
          lda      (r0L),y
          inc      r0L
          cpx      $d012
          beq      :57
          sta      $d020
          iny
          cpy      r1L
          bcc      :56

          lda      #$00      ;Rahmenfarbe löschen.
          sta      $d020

          dey
          dey
          lda      (r0L),y
          pha
          dey
          lda      (r0L),y
          iny
          sta      (r0L),y
          dey
          bne      :61
          pla
          sta      (r0L),y

          lda      #$00
          sta      $dc00      ;Tastenregister aktivieren.
          lda      $dc01      ;Tastenstatus einlesen.
          eor      #$ff      ;Wurde Taste gedrückt ?
          bne      :58      ;Ja, weiter...

          ldx      r1H      ;Zeiger auf Rasterzeile einlesen.
          inc      r1H      ;Zeiger auf nächste Zeile.
          bne      :54
          jmp      :52      ;Schleife...

::58      pla
          sta      $d020      ;Rahmenfarbe zurücksetzen.

          pla
          sta      $d015      ;Sprites wieder aktivieren.

          lda      $d011      ;Bildschirm wieder einschalten.
          ora      #%00010000
          sta      $d011

          pla
          sta      CPU_DATA    ;I/O-Bereich zurücksetzen.
          rts

```

;* Farbtabellen.**

; **Am Anfang/Ende muß ein NULL-Byte stehen um klare**
; **übergänge Balken/Bildschirm zu erzeugen!**

:ColGrfx1a **b \$00**
 b \$06,\$06,\$06,\$06,\$06
 b \$0e,\$0e,\$0e,\$0e
 b \$03,\$03,\$03
 b \$0d,\$0d
 b \$01
 b \$0d,\$0d
 b \$03,\$03,\$03
 b \$0e,\$0e,\$0e,\$0e
 b \$06,\$06,\$06,\$06,\$06
 b \$00

:ColGrfx1b

:ColGrfx2a **b \$00**
 b \$09,\$09,\$09,\$09,\$09
 b \$08,\$08,\$08,\$08
 b \$07,\$07,\$07
 b \$0f,\$0f
 b \$01
 b \$0f,\$0f
 b \$07,\$07,\$07
 b \$08,\$08,\$08,\$08
 b \$09,\$09,\$09,\$09,\$09
 b \$00

:ColGrfx2b

:ColGrfx3a **b \$00**
 b \$06,\$06,\$06,\$06,\$06
 b \$04,\$04,\$04,\$04
 b \$02,\$02,\$02
 b \$0a,\$0a
 b \$01
 b \$0a,\$0a
 b \$02,\$02,\$02
 b \$04,\$04,\$04,\$04
 b \$06,\$06,\$06,\$06,\$06
 b \$00

:ColGrfx3b

```
:ColGrfx4a      b $00
                 b $0b,$0b,$0b,$0b
                 b $0c,$0c,$0c
                 b $0f,$0f
                 b $01
                 b $0f,$0f
                 b $0c,$0c,$0c
                 b $0b,$0b,$0b,$0b
                 b $00

:ColGrfx4b

;*** Variablen.
:CurColTab      b $00
:ColTabVec       w ColGrfx1a
                 w ColGrfx2a
                 w ColGrfx3a
                 w ColGrfx4a
:ColTabLen       b ColGrfx1b-ColGrfx1a
                 b ColGrfx2b-ColGrfx2a
                 b ColGrfx3b-ColGrfx3a
                 b ColGrfx4b-ColGrfx4a
```

```
;*****
;*** Endadresse testen.
;*****
          g LD_ADDR_SCRSAVER + R2_SIZE_SCRSAVER -1
;*****
```