

# SHAPES 64

SYS 48152, ADR, X, Y, Flag  
 ADR: \$C400+X+40\*Y | Flag: 2 def, 1 setzen  
 \$Ausdehnung | 0: ..., Flag, Chr, Col  
 füllen

programmiert von R. Löwenstein

kommentiert am 05.09.1984

```

..C000 20 FD AE JSR $AEFD auf Komma testen
..C003 20 EB B7 JSR $B7EB ADR, X holen
..C006 85 04 STX $04 X nach 4
..C008 A5 14 LDA $14
..C00A A6 15 LDX $15 ADR nach
..C00C 85 02 STA $02 2/3
..C00E 86 03 STX $03
..C010 20 FD AE JSR $AEFD Komma
..C013 20 EB B7 JSR $B7EB 2 Werte
..C016 A5 14 LDA $14 Y-Wert nach
..C018 85 05 STA $05 5
..C01A 8A TXA Flag nach Akku
..C01B F0 20 BEQ $C03D Bereich füllen? ja
..C01D C9 00 CMP #$01 setzen?
..C01F F0 0E BEQ $C02F ja
..C021 A2 07 LDX #$07 8 Bytes
..C023 BD 59 C0 LDA $C059, X definieren 8 Bytes
..C026 9D 92 C0 STA $C092, X kopieren
..C029 CA DEX
..C02A 10 BPL $C023 LOOP
..C02C 4C 71 C0 JMP $C071 kopieren
..C02F A2 07 LDX #$07 8 Bytes
..C031 BD 59 C0 LDA $C061, X setzen - Bytes
..C034 9D 92 C0 STA $C092, X kopieren
..C037 CA DEX
..C038 10 F7 BPL $C031 LOOP
..C03A 4C 71 C0 JMP $C071 kopieren
..C03D 20 FD AE JSR $AEFD Komma
..C040 20 EB B7 JSR $B7EB Zeichen, Farbe holen
..C043 8E 6E C0 STX $C06E Farbe speichern
..C046 A5 14 LDA $14 zeichnen
..C048 8D 6A C0 STA $C06A speichern
..C04B A2 07 LDX #$07 8 Bytes
..C04D BD 69 C0 LDA $C069, X "füllen"-Bytes
..C050 9D 92 C0 STA $C092, X kopieren
..C053 CA DEX
..C054 10 F7 BPL $C04D LOOP
..C056 4C 71 C0 JMP $C071 kopieren (besser: BMI)
..C059 B1 02 LDA ($02), Y
..C05B 91 06 STA ($06), Y
..C05D B1 08 LDA ($08), Y definieren
..C05F 91 0A STA ($0A), Y
..C061 B1 06 LDA ($06), Y
..C063 91 02 STA ($02), Y
..C065 B1 0A LDA ($0A), Y setzen
..C067 91 08 STA ($08), Y
..C069 A9 FF LDA #$FF Zeichen (C063)
..C06B 91 02 STA ($02), Y füllen
..C06D A9 00 LDA #$00 Farbe (C069)
..C06F 91 08 STA ($08), Y
..C071 A9 36 LDA #$36 Jump von C03A, C056, C02C BASIC-ROM
..C073 85 01 STA $01 aus
..C075 A9 00 LDA #$00 Bildschirm ab
..C077 A2 00 LDX #$A0
..C079 85 06 STA $06 $A000 ablegen,
..C07B 86 07 STX $07
..C07D A2 B0 LDX #$B0 Color-RAM
..C07F 85 0A STA $0A
..C081 86 0B STX $0B ab $B000
  
```

Programm für  
 den  
 Atari  
 400

```

..C083 A6 02 LDX $02
..C085 A5 03 LDA $03 Anfangsadresse
..C087 18 CLC
..C088 69 14 ADC #$14 + $1400 ergibt
..C08A 85 09 STA $09 Color-RAM-Position
..C08C 86 08 STX $08 nach 8/9
..C08E A6 05 LDX $05 Y-Länge
..C090 A0 00 LDY #$00
..C092 A9 FF LDA #$FF
..C094 91 02 STA ($02), Y kopieren
..C096 A9 03 LDA #$03
..C098 91 02 STA ($02), Y
..C09A 0B INY nächstes Byte
..C09C 04 04 CPY $04 X erreicht?
..C09E 90 F3 BCC $C092 nein, LOOP
..C09F A5 06 LDA $06 SCREEN-Zwischenspeicher
..C0A1 18 CLC
..C0A2 65 04 ADC $04 um X-Position erhöhen
..C0A4 85 06 STA $06 zurückspeichern
..C0A6 85 0A STA $0A (low-Byte identisch!)
..C0A8 90 04 BCC $C0AE kein Übertrag
..C0AA E6 07 INC $07 High-Bytes
..C0AC E6 0B INC $0B erhöhen
..C0AE A5 02 LDA $02 Pos im SCREEN
..C0B0 18 CLC
..C0B1 69 28 ADC #$28 + 40
..C0B3 85 02 STA $02 ablegen
..C0B5 85 08 STA $08
..C0B7 90 04 BCC $C0BD kein Übertrag
..C0B9 E6 03 INC $03 High-Bytes erhöhen
..C0BB E6 09 INC $09
..C0BD CA DEX Y decr.
..C0BE D0 D0 BNE $C090 nach nicht fertig
..C0C0 A9 37 LDA #$37 ROM
..C0C2 85 01 STA $01 ein
..C0C4 60 RTS Zeichenatz
..C0C5 EA JSR von C06F, C14E, C159 * kopieren
..C0C6 A9 30 LDA #$30 1/0 aus, Zeichenatz
..C0C8 85 01 STA $01 ein
..C0CA A9 0A LDA #$0A
..C0CC A2 0B LDX #$0B von $C0F4, $C156, $C148
..C0CE A0 D0 LDY #$D0 verändert
..C0D0 85 02 STA $02 $C800
..C0D2 86 03 STX $03 $C800
..C0D4 85 04 STA $04 $D000
..C0D6 84 01 STY $01
..C0D8 A2 0B LDX #$0B 8 Blocks
..C0DA AB TAY Y=0
..C0DB B1 04 LDA ($04), Y von $D000
..C0DD 91 02 STA ($02), Y nach $C800 kopieren
..C0DF 88 DEY
..C0E0 D0 F3 BNE $C0DB LOOP
..C0E2 E6 03 INC $03 High-Bytes
..C0E4 E6 05 INC $05 erhöhen
..C0E6 8A DEX Blocks erniedrigen
..C0E7 D0 F2 BNE $C0DB LOOP
..C0E9 A9 37 LDA #$37 1/0 an, charset aus
..C0EB 85 01 STA $01
..C0ED 58 CLI
..C0EE 60 RTS und Tschüss!!!
..C0EF 78 SET
  
```

Loop  
 Loop  
 Loop

\*1) wird vom Demoprogramm mit RTS  
 1 überschrieben

INIT-

```

COEF 78 SEI
COF0 A9 D0 LDA #D0 von $D000 kopieren
COF2 8D CF C0 STA #COCF
COF5 20 C5 C0 JSR #C0C5 Zeichensatz ins RAM kopieren
COF8 A9 00 LDA #00 vic-Daten von $C000-$FFFF
COFA 8D 00 DD STA #DD00
COFD A9 12 LDA #12 = %00010010
COFF 8D 18 D0 STA #D018 SCREEN ab $C400, Zeichensatz ab $C800
C102 A9 04 LDA #C4 aus Betriebssystem mitteilen
C104 8D 02 02 STA #02B8 auf dem Bildschirm ab $C400 liegt
C107 A9 32 LDA #32 }
C109 A2 11 LDX #C1 } neuen IRQ nach
C10B 8D 14 03 STA #0314 }
C10E 8E 15 03 STX #0315 } $C132 setzen
C111 A9 00 LDA #00
C113 8D 20 D0 STA #D020 } Bildschirm schwarz
C116 8D 21 D0 STA #D021
C119 A9 0E LDA #0E Cursorfarbe
C11B 8D 06 02 STA #02B6 hellblau
C11E A9 3C LDA #3C }
C120 A2 03 LDX #03 } 1082c, Kassettenspeicher
C122 85 2B STA #2B als BASIC-Anfang
C124 86 2C STX #2C
C126 A9 00 LDA #00 eine 0 gegen
C128 8D 3C 03 STA #033C "SYNTAX ERROR"
C12B A9 08 LDA #08 Zeichensatzumschaltung
C12D 20 00 FF JSR #FFD2 verriegeln
C130 58 CLI
C131 60 RTS

```

```

C132 AD 8D 02 LDA #02BD SHIFT, COMMO, CTRL
C135 C9 03 CMP #03 SHIFT & COMMODE?
C137 D0 18 BNE #C151 wenn
C139 A9 12 LDA #12
C13B 8D 02 02 STA #02BD
C13E 8D 0E 02 STA #02BE
C141 E6 FF INC #FF Flag umschalten
C143 A5 FF LDA #FF Flag holen
C145 29 01 AND #01 Großschrift eingeschaltet?
C147 D0 0E BNE #C154 wenn
C149 A9 DB LDA #DB, Kleinschrift
C14B 8D CF D0 STA #COCF in Routine schreiben und
C14E 20 C5 C0 JSR #C0C5 kopieren
C151 4C 31 EA JMP #EA31 IRQ-Ende
C154 A9 D0 LDA #D0 Großschrift
C156 8D CF D0 STA #COCF in Routine schreiben und
C159 20 C5 C0 JSR #C0C5 kopieren!
C15C 4C 31 EA JMP #EA31 IRQ-Ende

```

```

C15F 78 SEI
C160 A9 D0 LDA #D0 Großschrift
C162 8D CF C0 STA #COCF
C165 4C C6 C0 JMP #C0C6 kopieren
C168 78 SEI
C169 A9 DB LDA #DB Kleinschrift
C16B 8D CF C0 STA #COCF
C16E 4C C6 C0 JMP #C0C6 kopieren

```

unbenutzt

Zeropage-Belegung:

- 2/3 : ADR
- 4 : X
- 5 : Y
- 6/7 : Zeiger auf BS-Zwischenspeicher
- 8/9 : Color-RAM-Position
- A/B : Zeiger auf Col-RAM Zwischenspeicher
- FF : Zeichensatz-Flag