

Neues vom SMON

Diese Erweiterung stellt elf weitere Befehle zur Verfügung. So läßt sich der Monitor zum Beispiel frei im Speicher verschieben und Sprites oder Zeichensätze können sehr einfach erstellt und geändert werden.

Um die Befehlserweiterung zu initialisieren, geht man folgendermaßen vor:

1. SMON absolut laden und NEW eingeben.
2. Den Basic-Lader (Listing 1) eintippen und speichern.
3. Nach dem Start des Laders die Startadresse (dezimal) Ihrer SMON-Version eingeben: zum Beispiel 49152 (= \$C000).
4. Den erweiterten SMON zum Beispiel mit "SMONEX" Startadresse Endadresse speichern.

Die neuen Routinen werden, genau wie die meisten bereits vorhandenen, durch einen Buchstaben, zum Teil gefolgt von Adressenangaben, aufgerufen. Bei den ersten drei Ausgabebefehlen kann der Speicherinhalt durch Überschreiben der Zeile geändert werden.

Z 4000 (4100) (Zeichendaten)

gibt den Speicherinhalt von \$4000 (bis \$40FF) folgendermaßen aus: Jeweils ein Byte pro Zeile wird in 8-Bit-Form dargestellt. Dabei ist ein »*« ein gesetztes, ein ».« dagegen ein nicht gesetztes Bit. Die beiden Zeichen sind willkürlich gewählt und können durch Überschreiben der Speicherzellen \$xE65, \$xE2D (Bit = 1) und \$xE69, \$xE30 (Bit = 0) in den Bildschirm-Code (!) der gewünschten Zeichen geändert werden.

Die Anwendung dieses Befehls liegt beispielsweise in der gezielten und anschaulichen Beeinflussung bestimmter Steuerbits in VIC, CIA etc. Andererseits lassen sich — besonders in Verbindung mit dem Kommando »Q« — Zeichendaten leicht modifizieren.

H 4000 (4100)

entspricht dem Befehl »Z« mit dem Unterschied, daß jeweils drei Byte pro Zeile ausgegeben werden. Das entspricht dem Format für Spritedaten. Auf diese Weise steht mit dem erweiterten SMON ein kleiner »Sprite-Editor« zur Verfügung.

N 4000 (4100) (Normaldarstellung)

interpretiert den Speicherinhalt von \$4000 (bis \$40FF) als Bildschirm-Code und gibt 32 Zeichen pro Zeile aus.

U 4000 (4100) (Übersicht)

Wie »N«, jedoch werden in einer Zeile 40 Zeichen dargestellt. Änderungen sind nur mit »N« möglich. Dieser Befehl dient hauptsächlich dazu, im Speicher abgelegte Bildschirminformationen so auszugeben, wie sie tatsächlich im 40-Zeichen/Zeile-Format aussehen würden. Dieser Befehl ist recht nützlich, um professionelle Videospiele zu analysieren, da hier Spielszenen oft im Bildschirm-Code gespeichert sind.

E 4000 (4100) (Erase)

ist der bereits im 64'er, Ausgabe 2/85 vorgeschlagene Erase-Befehl zum Füllen des Speicherbereiches von \$4000 bis \$40FF mit \$00.

Y 40

kopiert die vorhandene SMON-Version in nur drei Sekunden nach \$4000 bis \$4FFF und nimmt dabei alle notwendigen Anpassungen vor. Die ursprüngliche Speicherversion

des Monitors bleibt unverändert. Mit »G 4000« kann man in den neuen SMON springen. Von dem Byte-Wert, der übergeben werden muß, wird nur das obere Nibble (\$4) gewertet, so daß sich theoretisch 16 SMON-Versionen im Speicher unterbringen lassen, wobei natürlich nicht alle Möglichkeiten sinnvoll sind.

Auf diese Weise läßt sich stets die erforderliche Speicherversion herstellen, ohne daß langwierige Änderungen notwendig sind.

Q 2000

kopiert den Zeichensatz aus dem ROM von \$D000 bis \$DFFF in das RAM nach \$2000. Dort kann er mit dem Befehl »Z« nach Belieben geändert werden. Möchte man zum Beispiel das Zeichen »A« in ein »Ä« umdefinieren, so ist der Zeichensatz mit »Q 2000« ins RAM zu kopieren. Anschließend kann mit »Z 2000 2015« der Bereich in binärer Form auf dem Bildschirm ausgegeben werden, in dem auch das Zeichen »A« steht. Dieses kann nun in ein »Ä« geändert werden, indem man mit dem Cursor an die zu ändernde Stelle fährt und für einen Punkt, der gesetzt werden soll, ein »*« und für einen Punkt der nicht gesetzt werden soll ein ».« setzt. So, jetzt ist der Zeichensatz umdefiniert, aber noch nicht aktiviert. Als nächstes muß dem Videocontroller die Startadresse des neuen Zeichensatzes mitgeteilt werden. Dazu ist die Adresse \$D018, in der eine hexadezimale 15 steht, durch eine hexadezimale 18 zu ersetzen.

J

bringt den letzten Ausgabebefehl (K, D, M, Z, H, N, U) auf den Bildschirm zurück. Mit RETURN wird der letzte Befehl noch einmal ausgeführt.

Zum Schluß noch ein Tip:

DATA-Zeilen in Hex-Byte-Darstellung sind wegen ihrer konstanten Länge (immer zwei Ziffern pro Wert!) übersichtlicher als solche mit dezimalen Zahlen. Da für die Ausgabe von Hex-Werten bereits alle Routinen im SMON integriert sind, kann der »B«-Befehl (Basic-DATA-Zeilen erzeugen) durch Verändern eines einzigen Sprungbefehles dahingehend manipuliert werden, daß der Speicherinhalt künftig in Form von Hex-Byte ausgegeben wird:

Disassemblieren Sie dazu den Byte-Ausgabebefehl mit »D x99F« und ersetzen »JSR BDD1« durch »JSR x32A«. Für das »x« muß der 4-KByte-Block, in dem die zu ändernde SMON-Version steht, eingesetzt werden. Liegt Ihre SMON-Version bei \$C000, so ersetzen Sie das »x« durch ein »C«.

Die Gesamtlänge der DATA-Zeile kann außerdem durch Verändern der Speicherzelle \$x9AE variiert werden. Bei dem Wert \$1C werden zum Beispiel genau acht Hex-Byte pro Zeile ausgegeben. Das Assembler-Listing zu dieser Erweiterung zeigt Listing 2.

(Mark Richters/ah)

```

100 REM *****
110 REM *                               * <238>
120 REM *                               * <159>
130 REM *                               * <222>
140 REM *                               * <179>
150 REM * VON MARK RICHTERS           * <005>
160 REM * ALLERSTR.4                   * <068>
170 REM * 2806 OYTEN                   * <037>
180 REM * TEL.: 04207/1870             * <078>
190 REM *                               * <229>
200 REM *****                       * <072>
210 :                                  * <176>
220 PRINT"BITTE DIE STARTADRESSE"    * <248>
230 PRINT"IHRER SMON-VERSION EINGEBEN!" <189>
240 INPUT H : AL=H:H=H/4096          * <234>
250 IF H<>INT(H) THEN 210             * <052>
260 POKE AL+38,0                     * <239>
270 :                                 * <238>
280 DIM W(75)                        * <045>

```

Listing 1. Erweiterung zum SMON. Bitte die Eingabe-Hinweise auf Seite 109 beachten.

```

280 FOR I=0 TO 9
290 : W(48+I)=I
300 : W(65+I)=I+10
310 NEXT I
320 :
330 FOR I=1 TO 4
340 : READ K,Z : K=K+H*4096
350 : FOR J=K TO K+Z-1
360 : READ A$
370 : A=W(ASC(A$))
380 : B=W(ASC(RIGHT$(A$,1)))
390 : S=S+A+B
400 : IF ASC(A$)=42 THEN A=H
410 : P=A*16+B : POKE J,P
420 : NEXT J
430 NEXT I
440 :
450 IF S=7031 THEN PRINT "OK!":GOTO 470
460 PRINT "FEHLER IN DATAS !"
470 STOP
480 :
490 DATA 97,7
500 DATA 27,3A,3B,2C,2B,29,21
510 :
520 DATA 739,2B
530 DATA A2,06,DD,61,*0,F0,0B,CA
540 DATA 10,F8,A9,2E,20,D2,FF,20
550 DATA CA,*2,C9,2E,F0,F9,4C,85
560 DATA *F,4C,B2,*F
570 :
580 DATA 781,2
590 DATA F0,ED
600 :
610 DATA 3593,503
620 DATA A9,80,2C,A9,00,85,AB,20
630 DATA 64,*2,24,AB,10,03,A2,29
640 DATA 2C,A2,2B,20,40,*3,20,23
650 DATA *3,A0,06,A2,00,A1,FB,0A
660 DATA 4B,80,03,A9,2E,2C,A9,2A
670 DATA 91,D1,AD,86,02,91,F3,6B
680 DATA C8,E8,E0,0B,D0,E9,20,67
690 DATA *3,24,AB,30,04,C0,1E,90
700 DATA DA,20,5D,*4,90,C4,60,A0
710 DATA 0B,2C,A0,1B,20,7E,*2,20
720 DATA B8,*2,A2,0B,A9,00,85,AA
730 DATA 20,CA,*2,C9,2E,F0,07,C9
740 DATA 2A,F0,04,4C,D1,*2,1B,26
750 DATA AA,B8,CA,D0,EB,A5,AA,B1
760 DATA FB,C1,FB,D0,EE,20,67,*3
770 DATA C0,00,D0,D6,60,A9,80,2C
780 DATA A9,00,85,AB,20,64,*2,20
790 DATA 51,*3,24,AB,10,0B,A9,21

```

```

<096>
<224>
<069>
<140>
<042>
<048>
<123>
<235>
<038>
<253>
<129>
<023>
<010>
<118>
<070>
<004>
<162>
<200>
<071>
<026>
<202>
<136>
<025>
<232>
<137>
<159>
<187>
<184>
<109>
<038>
<220>
<141>
<068>
<202>
<085>
<117>
<239>
<071>
<144>
<079>
<139>
<233>
<133>
<100>
<236>
<052>
<202>
<014>
<071>
<218>
<243>
<138>
800 DATA 20,D2,FF,20,23,*3,A0,0B
810 DATA 2C,A0,00,A2,00,A1,FB,20
820 DATA 4F,*4,D0,F9,20,5D,*4,90
830 DATA DE,60,20,7E,*2,A2,00,A0
840 DATA 0B,B1,D1,B1,FB,C1,FB,D0
850 DATA AA,20,56,*4,90,F3,60,20
860 DATA 8D,*2,29,F0,85,FF,20,3C
870 DATA *F,20,D6,*9,20,3C,*F,A9
880 DATA 14,85,FB,A9,02,05,FF,85
890 DATA FC,A9,D1,85,FD,A9,0F,05
900 DATA FF,85,FE,20,68,*A,A0,00
910 DATA A2,00,8D,F2,*F,05,FF,85
920 DATA FC,CA,BD,F2,*F,85,FB,B1
930 DATA FB,29,0F,05,FF,91,FB,CA
940 DATA 10,E8,A9,2B,85,FB,A9,00
950 DATA 05,FF,85,FC,A0,35,B1,FB
960 DATA 29,0F,05,FF,91,FB,8B,8B
970 DATA 10,F4,A9,DE,85,FB,A9,0F
980 DATA 05,FF,85,FC,A0,13,B1,FB
990 DATA 29,0F,05,FF,91,FB,8B,8B
1000 DATA 10,F4,60,A5,FF,85,A9,20
1010 DATA 43,*F,6B,6B,29,F0,85,A5
1020 DATA 1B,69,10,85,A7,00,85
1030 DATA A4,85,A6,85,AB,60,20,7A
1040 DATA *2,A9,00,4C,C7,*9,20,7E
1050 DATA *2,A0,00,A9,D0,84,FD,85
1060 DATA FE,7B,A9,03,85,01,A2,10
1070 DATA B1,FD,91,FB,C8,D0,F9,E6
1080 DATA FC,E6,FE,CA,D0,F2,A9,27
1090 DATA 85,01,5B,60,4B,C9,4A,D0
1100 DATA 10,A0,27,B9,00,02,91,D1
1110 DATA 8B,10,F8,6B,C6,D6,4C,D6
1120 DATA *2,A0,06,D9,D7,*F,D0,0A
1130 DATA A0,27,B1,D1,99,00,02,8B
1140 DATA 10,F8,8B,10,EE,6B,4C,FF
1150 DATA *2,A2,0A,DD,D0,*F,F0,06
1160 DATA CA,D0,F8,4C,D1,*2,20,C5
1170 DATA *F,4C,D6,*2,8A,0A,AA,BD
1180 DATA DD,*F,4B,8D,DC,*F,4B,60
1190 DATA 2B,29,21,45,59,51,4B,5A
1200 DATA 4E,55,44,4B,4D,52,*E,4F
1210 DATA *E,B2,*E,56,*F,C7,*E,5E
1220 DATA *F,0B,*E,0B,*E,85,*E,8B
1230 DATA *E,06,00,87,03,2D,0C,5C
1240 DATA 0C,F5,0C,A2,0D,04,0E

```

6 64'er

Listing 1. Erweiterung zum SMON (Schluß)

```

110: C026 .OPT P4,00
;
; SMON-ERWEITERUNG
; ERWEITERT SMON UM FOLGENDE
; BEFEHLE - H ADR1 (ADR2)
; - Z ADR1 (ADR2)
; - N ADR1 (ADR2)
; - U ADR1 (ADR2)
; - E ADR1 ADR2
; - Y BYTE
; - Q ADR1
; - J
;
; -----
700: C026 S = $C000 ;BASISADRESSE SMON
;
720: C026 HCH = " " ;HIDDEN COMMANDS
730: C026 HCZ = " "
740: C026 HCN = "!"
;
760: C026 FLAG = $AA
770: C026 TEMP = $AB
780: C026 PCL = $FB
790: C026 PCH = $FC
800: C026 HINIBBLE = $FF
;
820: C026 CMDS = S+$02B
830: C026 GET12ADR = S+$264
840: C026 GET2ADR = S+$27A
850: C026 GETADR1 = S+$27E
860: C026 GETBYT = S+$28D
870: C026 SKIPSPAC = S+$2B8
880: C026 GETRET = S+$2C2
890: C026 GETCHRR = S+$2CA
900: C026 ERROR = S+$2D1
910: C026 EXECUTE = S+$2D6

```

```

920: C026 CMDSTORE = S+$2FF
930: C026 CHARRET = S+$340
940: C026 RETURN = S+$351
950: C026 HEXOUT = S+$323
960: C026 PCINC = S+$367
970: C026 ASCI14 = S+$44F
980: C026 ASCI15 = S+$456
990: C026 CONTIN = S+$45D
1000: C026 OCCUPY = S+$9C7
;
1011: C026 ; S+$3B
1012: C026 00 .BYTE00 ;ALTEN Z-BEFEHL LOESCHEN
;
1020: C061 ; S+$61
;
1040: C061 27 3A 3B HCMDTAB .ASC ":", "
1050: C065 28 29 21 .BYTEHCH,HCZ,HCN
;
1070: C2E3 ; S+$2E3
;
1090: C2E3 A2 06 LDX #6 ;ZEILENANFANG
1100: C2E5 DD 61 C0 F1 CMP HCMDTAB,X
1110: C2E8 F0 0B BEQ EXEC1
1120: C2EA CA DEX
1130: C2E8 10 FB BPL F1
1140: C2ED A9 2E LDA #".
1150: C2EF 20 D2 FF JSR $FFD2
;
1170: C2F2 20 CA C2 EXEC1 JSR GETCHRRR
1180: C2F5 C9 2E CMP #".
1190: C2F7 F0 F9 BEQ EXEC1
1200: C2F9 4C 85 CF JMP LINSTORE
1210: C2FC 4C B2 CF WEITER JMP MORECMD
;

```

Listing 2. Assembler-Listing zur SMON-Erweiterung

```

1230: C30D      ** S+$30D      ;WEITERE BEFEHLE
1250: C30D F0 ED      ;      BEQ WEITER
1270: CE09      ** S+$E09      ;NEUE ROUTINEN
;
;-----
1310: CE09 A9 80      ZCMD      LDA #80      ;FLAG SETZEN
1320: CE0B 2C          ;      .BYTE$2C
;
1340: CE0C A9 00      HCMD      LDA #0
1350: CE0E 85 AB      STA TEMP
1360: CE10 20 64 C2    JSR GET12ADR ;START/END-ADRESSE
;
1380: CE13 24 AB      L1        BIT TEMP
1390: CE15 10 03      BPL W0
1400: CE17 A2 29      LDX #HCZ      ;HIDDEN COMMAND
1410: CE19 2C          ;      .BYTE$2C
1420: CE1A A2 28      W0        LDX #HCH
1430: CE1C 20 40 C3    JSR CHARRET ;AUSGEBEN
1440: CE1F 20 23 C3    JSR HEXOUT  ;PC AUSGEBEN
1450: CE22 A0 06      LDY #6      ;SPALTE 6
1470: CE24 A2 00      L2        LDX #0
1480: CE26 A1 FB      LDA (PCL,X)
;
1500: CE28 0A          L3        ASL
1510: CE29 48          PHA
1520: CE2A B0 03      BCS BITSET ;BYTE MERKEN
1530: CE2C A9 2E      LDA #".      ;BIT=1, DANN *
1540: CE2E 2C          ;      .BYTE$2C
1550: CE2F A9 2A      BITSET     LDA #"*
1560: CE31 91 D1      STA ($D1),Y
1570: CE33 A0 86 02    LDA $286
1580: CE36 91 F3      STA ($F3),Y
;
1600: CE38 68          PLA
1610: CE39 C8          INY
1620: CE3A EB          INX
1630: CE3B E0 08      CPX #8
1640: CE3D D0 E9      BNE L3
;
1660: CE3F 20 67 C3    JSR PCINC ;ZAEHLER ERHOEHEN
1670: CE42 24 AB      BIT TEMP ;FLAG FUER 1*8 BIT
1680: CE44 30 04      BHI W9
1690: CE46 C0 1E      CPY #30
1700: CE48 90 DA      BCC L2
1710: CE4A 20 5D C4 W9 JSR CONTIN ;TASTEN-ABFRAGE
1720: CE4D 90 C4      BCC L1
1730: CE4F 60          RTS
;
;-----
1770: CE50 A0 08      ZCMDH     LDY #8
1780: CE52 2C          ;      .BYTE$2C
1800: CE53 A0 18      HCMDH     LDY #24 ;3 BYTE
1810: CE55 20 7E C2    JSR GETADR1
1820: CE58 20 B8 C2    JSR SKIPSPACE ;SPACES UEBERLESEN
1830: CE5B A2 08      A1        LDX #8
1840: CE5D A9 00      LDA #0
1850: CE5F 85 AA      STA FLAG
;
1870: CE61 20 CA C2 A2 JSR GETCHERR ;. => BIT=0
1880: CE64 C9 2E      CMP #".      ;* => BIT=1
1890: CE66 F0 07      BEQ BIT0
1900: CE68 C9 2A      CMP #"*
1910: CE6A F0 04      BEQ BIT1
1920: CE6C 4C D1 C2 ERR1 JMP ERROR ;ANDERES ZEICHEN
;
1940: CE6F 18          BIT0      CLC
1950: CE70 26 AA      BIT1      ROL FLAG
1960: CE72 88          DEY
1970: CE73 CA          DEX
1980: CE74 D0 EB      BNE A2 ;BYTE
;
2000: CE76 A5 AA      LDA FLAG ;IN SPEICHER
2010: CE78 B1 FB      STA (PCL,X) ;SCHREIBEN
2020: CE7A C1 FB      CMP (PCL,X)
2030: CE7C D0 EE      BNE ERR1
2040: CE7E 20 67 C3    JSR PCINC
;
2060: CE81 C0 00      CPY #0
2070: CE83 D0 D6      BNE A1 ;NOCH NICHT FERTIG
2080: CE85 60          RTS
;
;-----
2120: CE86 A9 80      NCMD      LDA #80
2130: CE88 2C          ;      .BYTE$2C
2140: CE89 A9 00      UCMD      LDA #0
2150: CE8B 85 AB      STA TEMP
2160: CE8D 20 64 C2    JSR GET12ADR
;
2180: CE90 20 51 C3 L5 JSR RETURN
;
2190: CE93 24 AB      BIT TEMP
2200: CE95 10 0B      BPL U
2210: CE97 A9 21      LDA #HCN ;HIDDEN COMMAND
2220: CE99 20 D2 FF    JSR $FFD2
2230: CE9C 20 23 C3    JSR HEXOUT
2240: CE9F A0 08      LDY #8 ;SPALTE 8
2250: CEA1 2C          ;      .BYTE$2C
;
2270: CEA2 A0 00      U        LDY #0
2280: CEA4 A2 00      LDX #0 ;SPALTE 0
;
2300: CEA6 A1 FB      L4        LDA (PCL,X)
2310: CEAB 20 4F C4    JSR ASCII4 ;ALS BILDSCHIRM-CODE
2320: CEAB D0 F9      BNE L4 ;AUSGEBEN
;
2340: CEAD 20 5D C4    JSR CONTIN
2350: CEB0 90 DE      BCC L5 ;NAECHSTE ZEILE
2360: CEB2 60          RTS
;
;-----
2380: CEB3 20 7E C2 NCMDH     JSR GETADR1
2390: CEB6 A2 00      LDX #0
2400: CEB8 A0 08      LDY #8
2410: CEBB B1 D1      C1        LDA ($D1),Y
2420: CEBD B1 FB      STA (PCL,X) ;IN SPEICHER
2430: CEBE C1 FB      CMP (PCL,X) ;SCHREIBEN

```

```

2440: CEC0 D0 AA      BNE ERR1
2450: CEC2 20 56 C4    JSR ASCII5 ;PC ERHOEHEN
2460: CEC5 90 F3      BCC C1
2470: CEC7 60          RTS ;ZEILE FERTIG
;
;-----
2510: CEC8 20 8D C2 YCMD JSR GETBYT
2520: CECB 29 F0      AND #X11110000
2530: CEDD 85 FF      STA HINIBBLE ;NEUER 4K-BLOCK
2540: CECF 20 3C CF    JSR SETPTR ;ADRESSEN SETZEN
2550: CED2 20 D6 79    JSR $79D6 ;W-BEFEHL
;
2570: CED5 20 3C CF    JSR SETPTR
2580: CED8 A9 14      LDA #14 ;BEREICH OHNE
2590: CEDA 85 FB      STA #FB ;TABELLEN
2600: CEDC A9 02      LDA #2
2610: CEDE 05 FF      ORA HINIBBLE
2620: CEE0 85 FC      STA #FC
2630: CEE2 A9 D1      LDA #<NEWCMDS
2640: CEE4 85 FD      STA #FD
2650: CEE6 A9 0F      LDA #<NEWCMDS>8 & #F
2660: CEE8 05 FF      ORA HINIBBLE
2670: CEEA 85 FE      STA #FE
2680: CEEC 20 68 7A    JSR $7A68 ;V-BEFEHL
;
2700: CEEF A0 00      LDY #0 ;LADE-BEFEHLE
2710: CEF1 A2 0D      LDX #13 ;ANPASSEN
2720: CEF3 8D F2 CF D3 ORA CHANGE,X
2730: CEF6 05 FF      ORA HINIBBLE
2740: CEF8 85 FC      STA PCH ;ADRESSE ALS ZEIGER
2750: CEFA CA          DEX
2760: CEFB 8D F2 CF    LDA CHANGE,X
2770: CEF6 85 FB      STA PCL
2780: CF00 B1 FB      LDA (PCL),Y
2790: CF02 29 0F      AND #X00001111
2800: CF04 05 FF      ORA HINIBBLE
2810: CF06 91 FB      STA (PCL),Y
2820: CF08 CA          DEX
2830: CF09 10 E8      BPL D3
;
2850: CF0B A9 2B      LDA #<CMDS ;ZEIGER AUF
2860: CF0D 85 FB      STA PCL ;BEFEHLSADRESSEN
2870: CF0F A9 00      LDA #<CMDS>8 & #F
2880: CF11 05 FF      ORA HINIBBLE
2890: CF13 85 FC      STA PCH
2910: CF15 A0 35      LDY #53
2920: CF17 B1 FB      LDA (PCL),Y
2930: CF19 29 0F      AND #X00001111
2940: CF1B 05 FF      ORA HINIBBLE ;HIBYTES
2950: CF1D 91 FB      STA (PCL),Y ;ANGLEICHEN
2960: CF1F 88          DEY
2970: CF20 88          DEY
2980: CF21 10 F4      BPL D1
;
3000: CF23 A9 DE      LDA #<NEWADR ;ZEIGER AUF NEUE
3010: CF25 85 FB      STA PCL ;BEFEHLSADRESSEN
3020: CF27 A9 0F      LDA #<NEWADR>8 & #F
3030: CF29 05 FF      ORA HINIBBLE
3040: CF2B 85 FC      STA PCH
;
3060: CF2D A0 13      LDY #TABEND-NEWADR-1
3070: CF2F B1 FB      LDA (PCL),Y
3080: CF31 29 0F      AND #X00001111
3090: CF33 05 FF      ORA HINIBBLE ;HIBYTES
3100: CF35 91 FB      STA (PCL),Y ;ANGLEICHEN
3110: CF37 88          DEY
3120: CF38 88          DEY
3130: CF39 10 F4      BPL D2
3140: CF3B 60          RTS
;
;-----
3170: CF3C A5 FF      SETPTR     LDA HINIBBLE
3180: CF3E 85 A9      STA #A9 ;NEUER START HI
3190: CF40 20 43 CF    JSR GETHI
3200: CF43 68          PLA
3210: CF44 68          PLA
3220: CF45 29 F0      AND #X11110000
3230: CF47 85 A5      STA #A5 ;ALTER START HI
3240: CF49 18          CLC
3250: CF4A 69 10      ADC #10
3260: CF4C 85 A7      STA #A7 ;ALTES ENDE HI
;
3280: CF4E A9 00      LDA #0
3290: CF50 85 A4      STA #A4 ;LO-BYTES LOESCHEN
3300: CF52 85 A6      STA #A6
3310: CF54 85 A8      STA #A8
3320: CF56 60          RTS
;
;-----
3360: CF57 20 7A C2 ECMD JSR GET2ADR
3370: CF5A A9 00      LDA #0
3380: CF5C 4C C7 C9    JMP OCCUPY
;
;-----
3420: CF5F 20 7E C2 QCMD JSR GETADR1
3430: CF62 A0 00      LDY #0
3440: CF64 A9 D0      LDA #D0 ;ZEIGER AUF
3450: CF66 84 FD      STY #FD ;ZEICHENROM
3460: CF68 85 FE      STA #FE
;
3480: CF6A 78          SEI
3490: CF6B A9 03      LDA #X011 ;ROM EINBLENDEN
3500: CF6D 85 01      STA 1
3510: CF6F A2 10      LDX #16 ;4K UEBERTRAGEN
3520: CF71 B1 FD      LDA (#FD),Y
3530: CF73 91 FB      STA (PCL),Y
3540: CF75 C8          INY
3550: CF76 D0 F9      BNE E1
3560: CF78 E4 FC      INC PCH
3570: CF7A E4 FE      INC #FE
3580: CF7C CA          DEX
3590: CF7D D0 F2      BNE E1

```

Listing 2. Assembler-Listing zur SMON-Erweiterung
(Fortsetzung)

```

3600: CF7F A9 27      LDA #27      ; NORMALEINSTELLUNG
3610: CF81 85 01      STA 1
3620: CF83 58        CLI
3630: CF84 60        RTS

; -----
;
3670: CF85 48      LINSTORE PHA      ; BEFEHL MERKEN
3680: CF86 C9 4A      CMP #J"
3690: CF88 D0 10      BNE STORE

;
3710: CF8A A0 27      LDY #39
3720: CF8C B9 00 02 G1 LDA $0200,Y
3730: CF8F 91 D1      STA ($D1),Y      ; ZEILE AUF
3740: CF91 88      DEY      ; BILDSCHIRM
3750: CF92 10 F8      BPL G1      ; SCHREIBEN
3760: CF94 68      PLA
3770: CF95 C6 D6      DEC $D6      ; CURSOR 1 HOCH
3780: CF97 4C D6 C2    JMP EXECUTE

3800: CF9A A0 06      STORE LDY #6
3810: CF9C D9 D7 CF G3 CMP OUTCMDS,Y
3820: CF9F D0 0A      BNE W3

3840: CFA1 A0 27      OK1 LDY #39
3850: CFA3 B1 D1      G2 LDA ($D1),Y
3860: CFA5 99 00 02   STA $0200,Y      ; ZEILE NUR BEI
3870: CFA8 88      DEY      ; H,Z,N,U,K,M,D
3880: CFA9 10 F8      BPL G2      ; SPEICHERN

3900: CFAB 88      W3 DEY
3910: CFAC 10 EE      BPL G3
3920: CFAE 68      STEND PLA      ; BEFEHL ZURUECKHOLEN
3930: CAF4 4C FF C2   JMP CMDSTORE

;
; ++++++++
;
3970: CFB2 A2 0A      MORECMD LDX #NEWADR-NEWCMDS-3
3980: CFB4 D0 D0 CF B1 CMP NEWCMDS-1,X
3990: CFB7 F8 06      BEQ FOUND
4000: CFB9 CA      DEX
4010: CFB4 D0 F8      BNE B1
4020: CFBC 4C D1 C2   JMP ERROR

```

```

4040: CFBF 20 C5 CF FOUND JSR CMDEXEC2
4050: CFC2 4C D6 C2      JMP EXECUTE

;
4070: CFC5 8A      CMDEXEC2 TXA
4080: CFC6 0A      ASL
4090: CFC7 AA      TAX
4100: CFC8 BD DD CF LDA NEWADR-1,X
4110: CFCB 48      PHA
4120: CFCC BD DC CF LDA NEWADR-2,X
4130: CFCF 48      PHA
4140: CFDD 60      RTS

4160: CFD1 28 29 21 NEWCMDS .BYTEHCH,HCZ,HCN
4170: CFD4 45 59 51 .ASC "EYQ"
4180: CFD7 48 5A 4E OUTCMDS .ASC "HZNUDKM"

;
4200: CFDE 52 CE      NEWADR .WORDHCMDSH-1
4210: CFE0 4F CE      .WORDZCMDSH-1
4220: CFE2 B2 CE      .WORDNCMDSH-1
4230: CFE4 56 CF      .WORDECMD-1
4240: CFE6 C7 CE      .WORDYCMD-1
4250: CFE8 5E CF      .WORDQCMD-1
4260: CFEA 08 CE      .WORDHCMDS-1
4270: CFEC 08 CE      .WORDZCMD-1
4280: CFEE 85 CE      .WORDNCMD-1
4290: CFF0 88 CE      .WORDUCMD-1
4300: CFF2      TABEND = *

4320: CFF2 06 00      CHANGE .WORD$0005+1
4330: CFF4 87 03      .WORD$0306+1
4340: CFF6 2D 0C      .WORD$0C2C+1
4350: CFF8 5C 0C      .WORD$0C5B+1
4360: CFFA F5 0C      .WORD$0CF4+1
4370: CFFC A2 0D      .WORD$0DA1+1
4380: CFFE 04 0E      .WORD$0E03+1

```

Listing 2. Assembler-Listing zur SMON-Erweiterung (Schluß)



Gordon Saga

Best.-Nr. MD 240 A

DM 39,-* (Stf. 35,50/65 351,-)

* Inkl. MwSt. Unverbindliche Preisempfehlung.

Das neueste deutsche Grafik-Adventure für Ihren Commodore 64

Suchen Sie die Pforte zu einer anderen Welt!
Beweisen Sie Ihren Spürsinn, denn der richtige Weg ist schwer zu finden, und überall lauern Gefahren!

- hochauflösende Grafik
- ausführliche Spielanweisungen
- riesiger Befehlsvorrat
- Eingabe von ganzen Sätzen möglich
- variabler Spielablauf

**Markt&Technik
BUCHVERLAG**

Hans-Pinsel-Straße 2, 8013 Haar bei München
Schweiz: Markt & Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, ☎ 042/41 56 56
Österreich: Rudolf Lechner & Sohn, Heilwerkstraße 10, A-1232 Wien, ☎ 02 22/67 75 26

Eine neue Dimension der Abenteuerspiele:
Kein Spiel gleicht dem anderen — Sie geraten in Situationen, in denen Sie Ihre Spieltaktik völlig ändern müssen.
Überzeugen Sie sich selbst!

Happy Software gibt's beim Buchhändler, bei Horten, Quelle und im Computershop. Bestellkarten bitte an Ihren Buchhändler oder an eine unserer Depotbuchhandlungen.
Adressenverzeichnis am Ende des Heftes!