

Zeicheneditor für C 64

In der Kürze liegt hier die Würze. Statt seitenlanger Listings brauchen Sie nur eine einzige Bildschirmseite abzutippen, um den Zeichensatz Ihres C 64 ganz nach Belieben ändern zu können.

Das Programm »Character Change« (Listing 1) ist ein Hilfsprogramm, um den Zeichensatz des C 64 beliebig zu verändern und anschließend auf Diskette abzuspeichern. Das ganze Listing wirkt zwar auf den ersten Blick etwas unübersichtlich, aber dafür sind es halt auch bloß 24 Zeilen.

Der Vollständigkeit halber ist auch noch ein Demo-Programm dabei, das es ermöglicht, mit diesem veränderten Zeichensatz in eigenen Programmen zu arbeiten. Einsatzmöglichkeiten gibt es genug, etwa Spiele.

Ein paar Erläuterungen zum Programm:

Nach dem Starten mit »RUN«, fragt der Computer »KEY ?«. Hier kann ein beliebiges Zeichen über die Tastatur eingegeben werden. Allerdings muß es an der Stelle stehen, an der der Cursor blinkt, da aus dieser Speicherzelle des Video-RAMs der Bildschirmcode geholt wird.

Nachdem die Eingabe mit RETURN abgeschlossen wurde, erscheint die Matrix des Zeichens vergrößert auf dem Bildschirm. Dabei steht ein Sternchen für gesetztes, ein Punkt für ein nicht gesetztes Pixel.

Jetzt kann die Zeichenmatrix verändert werden, und zwar zeilenweise. Wieder durch Punkte und Sternchen kennzeichnen, wo ein Punkt gesetzt beziehungsweise nicht gesetzt werden soll. Danach RETURN drücken und die nächste Zeile ändern.

Sind alle acht Zeilen neu eingegeben, so kann man sich für »W« wie »weiter« und »S« wie »save« entscheiden. Drückt man »W«, so geht das Spielchen von vorne los, mit »S« wird der Zeichensatz auf Diskette abgespeichert. Drückt man eine andere Taste, so erfolgt ein Programmabbruch.

Das File des Zeichensatzes nennt sich »character ram«. Das Demo-Programm (Listing 2) lädt einen bereits veränderten Zeichensatz von Diskette und stellt dann alle 256 Zeichen auf dem Bildschirm dar.

So arbeitet das Programm

Noch ein paar Worte zur Arbeitsweise des Programms.

Zuerst wird von einem in DATAs abgelegten Maschinenprogramm mit Hilfe der Blockverschieberoutine des Betriebssystems der Zeichensatz zweimal in verschiedene RAM-Bereiche kopiert. Einmal nach \$E000, einmal nach \$9000. Dies ist erforderlich, um die Zeichenmatrix mit PEEK auslesen zu können, was im unter dem Betriebssystem liegenden RAM bekanntlich nicht geht. Die zweite Kopie ist die, aus der der Computer jetzt die Zeichen ausliest. Danach werden von Basic aus die Zeiger für den Beginn des Video-RAMs, den Zeichengenerator und die damit zusammenhängenden 16 KByte für den Video-Controller neu gesetzt. Das Video-RAM wird also zwangsweise mit verschoben, und zwar nach dezimal 49152.

Aus dem Bildschirmcode des eingegebenen Zeichens wird die Startadresse in der ersten Kopie des Zeichensatzes ermittelt und 8 Bytes ausgelesen. Aus diesen 8 Bytes werden dann die Strings für die Bildschirmdarstellung gebildet. Umgekehrt werden aus den eingegebenen Strings die Bytes für den Zeichensatz gebildet und anschließend in beide Kopien gePOKET.

Ein zweites Maschinenprogramm ruft dann die SAVE-Routine auf und speichert die 4 KByte der ersten Kopie auf Diskette. Eine erste Kopie deshalb, um sie nach dem Laden in andere Programme nochmals lesen oder verändern zu können.

Das Demo-Programm enthält im wesentlichen eine weitere Maschinenspracheroutine, die den Zeichensatz nach \$9000 lädt und dann nach \$E000 verschiebt. Die Umschaltung der Zeiger geschieht auch hier von Basic aus, um dem Anwender zu überlassen, wann er den neuen Zeichensatz verwenden will.

(Marc Ritter/ev)

```

1 POKE 53280,0:POKE 53281,0:FOR I=832 TO 9
11:READ A:POKE I,A:NEXT:SYS 832:POKE 850
,160 <127>
2 POKE 56,144:CLR:SYS 832:POKE 56576,PEEK(
56576)AND 252:POKE 53272,8:POKE 648,192 <014>
3 INPUT" {CLR,YELLOW}KEY";Z$:AD=36864+(PEEK
(53272)AND 2)*1024+B*PEEK(49157) <242>
4 FOR I=0 TO 7:A$(I)="" :Y=PEEK(AD+I):FOR K
=7 TO 0 STEP-1:Z$="." :X=2↑K <119>
5 IF (Y AND X)THEN Z$="*" <023>
6 A$(I)=A$(I)+Z$:NEXT:PRINT" {2SPACE}" +A$(I
):NEXT:PRINT" {HOME}":FOR I=0 TO 7:INPUT"
{LIG.RED}";B$:A=0 <224>
7 FOR K=0 TO 7:IF MID$(B$,8-K,1)=""THEN A
=A +2↑K <137>
9 NEXT:POKE AD+I+20480,A:POKE AD+I,A:NEXT:
PRINT" {RVSON}S {RVOFF}AVE/ {RVSON}W {RVOFF}
EITER":WAIT 203,63:GET C$ <226>
10 IF C$="S"THEN SYS 866:GOTO 3 <117>
11 DATA 120,169,51,133,1,169,,133,95,133,9
0,133,88,169,208,133,96:IF C$="W"THEN 3 <178>
12 END:DATA 169,240,133,89,169,224,133,91,
32,191,163,169,55,133,1,88,96,169,,133 <190>
13 DATA 251,169,144,133,252,162,8,32,186,2
55,169,13,162,130,160,3,32,189,255,169 <239>
14 DATA 251,162,,160,160,32,216,255,96,67,
72,65,82,65,67,84,69,82,32,82,65,77,32 <248>

```

© 64'er

Listing 1. Der Zeicheneditor. Bitte beachten Sie bei der Eingabe den Checksummer.

```

10 REM CHARACTER DEMO <186>
100 FOR I= 828 TO 900 :READ X:POKE I,X:NEX
T <057>
110 DATA 162, 8, 160, 1, 32, 186, 255, 169
, 13, 162, 115, 160, 3, 32 <014>
120 DATA 189, 255, 169, 0, 32, 213, 255, 1
20, 169, 51, 133, 1, 169, 0 <001>
130 DATA 133, 95, 133, 90, 133, 88, 169, 1
44, 133, 96, 169, 240, 133, 89 <017>
140 DATA 169, 160, 133, 91, 32, 191, 163,
169, 55, 133, 1, 88, 96, 67 <163>
150 DATA 72, 65, 82, 65, 67, 84, 69, 82, 3
2, 82, 65, 77, 0, 0 <165>
160 DATA 0, 0, 0 <026>
200 POKE 56,144:CLR:SYS 828:POKE 56,160:CL
R <142>
205 POKE 53272,8:POKE 56576,PEEK(56576)AND
252:POKE 648,192 <189>
210 PRINT" {CLR,8DOWN}":FOR I=0 TO 255:POKE
49152+I,I:POKE 55296+I,3:NEXT <217>

```

© 64'er

Listing 2. Demo-Programm zum Zeicheneditor. .