

Nun, was geschieht da, besonders wo der C 64 derartig hohen Zeilennummern sonst nur mit einem kalten Lächeln (auch Syntax-Error genannt) begegnet?

Es geschieht folgendes:

Die Eingabewarteschleife (\$A480) stellt fest, daß die soeben eingegebene Zeile (»35072120«) mit einer Ziffer beginnt und entscheidet daher, daß es sich um eine in das Programm einzubauende Programmzeile handelt.

Also übergibt sie die Zeile an die Routine zum Einfügen von Programmzeilen (\$A49C).

Die Einfügeroutine versucht nun als erstes (und in diesem Falle auch als letztes), die Zeilennummer ins Low-/High-Byte-Format zu verwandeln. Dazu verzweigt sie nach \$A96B, wo die Routine liegt, die für das ganze »Wunder« verantwortlich ist. Denn: Ist die Zeilennummer (wie unsere) zu groß, so versucht die Routine verzweigt, einen Syntax-Error von sich zu geben (aber die entsprechende Ausgabeschleife ist für einen BRANCH-Befehl zu weit entfernt). Also (da man ja unmöglich drei Byte für einen »JMP \$AF08« opfern konnte) wird nach \$A953 verzweigt, in der Hoffnung, daß der Akku etwas anderes als \$89 enthält. Denn \$A953 liegt mitten in der ON-GOTO-Routine und dort wird, falls der Akku etwas anderes als den GOTO-Code (eben \$89) enthält, zu einem Syntax-Error-Ansprung verzweigt.

Nur lautet die Zeilennummer 35072120, so ist der im Akku enthaltene »Rest« dummerweise gerade \$89, also wird nicht zum Syntax-Error gesprungen, sondern der Prozessor beginnt eine Fahrt ins Ungewisse — vorbei an total unpassenden Befehlen:

- vorbei an einem JSR CHRGET
- weiter bei einem erneuten Versuch, eine Zeilennummer zu holen (leider ist aber die Zeile zu Ende, also \$00)
- dieses \$00 ist auch nicht gleich einem Komma...
also muß der auf dem Stack liegende Code für ON-GOTO geholt werden (PLA) und das Ganze endet mit einem RTS.

Nur... es liegt ja gar kein ON-GOTO-Code auf dem Stack, sondern die Rücksprungadresse aus der Eingabewarteschleife!

Also wirft der ahnungslose Prozessor das Rücksprung-Low-Byte in die Ecke und hält nun das Rücksprung-High-Byte für ein Low-Byte und das nachfolgende Byte für ein High-Byte. Daraus folgt: der Prozessor springt ins Ungewisse: zu einer Adresse mit Low-Byte \$A5. Als High-Byte liegt bei ordnungsgemäß gereinigtem Stack \$79 obenauf (ehemaliger Rücksprung nach \$A67A während der Basic-Initialisierung). Dies führt zu einem Sprung nach \$79A5, wo der Professor hoffentlich auf ein BRK trifft, den sonst... Einen schweigenden, cursorlosen C 64 hat wohl jeder schon einmal gesehen.

Aber meist liegt dort ein BRK (oder ein mit TAX gepflasterter Weg dorthin), was den wild gewordenen Computer wieder auf den rechten Weg bringt: Nämlich die NMI-Routine.

Diese NMI-Routine, die bei unblockierter RUN/STOP-Taste auch über RUN/STOP-RESTORE erreichbar ist, setzt unter anderem den Stop-Vektor (Adresse 808/809!) neu und schon ist das Listing wieder auf den Beinen.

Hier noch einmal zusammengefaßt die Irrfahrt des C 64:

A494 BCC A49C	falls Zeile mit Ziffer am Anfang
A49C JSR A96B	Zeilennummer in Low/High-Format wandeln

A96B	
A97B CMP #19	
A97D BCS A953	falls die Zeilennummer zu groß wird
A953 CMP #89	was ja zutrifft
A955 BNE A8E8	Syntax-Error-Ansprung
A957 DEC \$65	
A959 BNE A95F	\$65 war alles andere nur nicht 1,
	also verzweigen
A95F JSR 0073	CHRGET
A962 JSR A96B	Zeilennummer holen, bei Zeilenende \$00
A965 CMP #", "	\$00 ist aber nicht gleich ", "
A967 BEQ A957	also nicht verzweigen
A969 PLA	lässt das Rücksprung-Low-Byte verschwinden

A96A RTS
Oben auf dem Stack hätte \$9E, \$A4, \$79, \$A6 gelegen, das \$9E wurde jedoch durch das PLA vom Stack geholt. Also erfolgt ein »Rücksprung« nach \$79A5.

Man kann dies ganz einfach ausprobieren, indem man in den (frisch eingeschalteten) Computer, ein POKE 31141,2 (= \$79A5) eingibt. Dies führt dann bei Eingabe von »35072120« zum Absturz des Prozessors. (Befehl \$02 = Prozessorabsturz).

So weit so gut. Nur — hätte es ein POKE 2,0:SYS2 nicht auch getan?

(Frank Sander/tr)

HiRes-Scrolling per Interrupt

Wer hat nicht schon solche Titelbilder wie bei dem Spiel »Boulderdash« gesehen? Man hat den Eindruck, als ob der Hintergrund stufenlos scrollen würde, die Schrift aber stehenbliebe. Solch ein Effekt wäre natürlich auch hervorragend für eigene Spielszenen geeignet. Das folgende Maschinenprogramm läuft völlig interruptgesteuert und simuliert diese Funktion in vielfältiger Weise. Das Maschinenprogramm (Listing 1) wird zuerst mit »SYS 49152« gestartet.

Der erste Befehl schaltet das Pseudo-Scrolling ein. Die Syntax lautet: »SYS 49232,a,b,c«. Nun zu den Parametern: Der Parameter a ist für die Geschwindigkeit zuständig. Sinnvoll

```
programm : pseudo-scroll c000 c15f
-----
c000 : 78 a5 01 48 29 fb 85 01 1f
c008 : a9 d0 85 03 a9 38 85 05 58
c010 : a0 00 84 02 84 04 a2 10 25
c018 : b1 02 91 04 c8 d0 f9 e6 78
c020 : 03 e6 05 ca d0 f2 68 85 82
c028 : 01 ad 18 d0 09 0e 8d 18 87
c030 : d0 58 a9 08 20 d2 ff a2 76
c038 : 28 a0 00 bd 07 c1 99 c0 ce
c040 : 3a e8 c8 c0 08 d0 f4 a9 67
c048 : 05 85 fa a9 00 85 fe 60 ed
c050 : 20 37 c0 20 fd ae 20 9e 53
c058 : b7 8e 05 c1 20 fd ae 20 bd
c060 : 9e b7 e0 00 d0 03 20 Bd d3
```

```
c068 : c0 20 fd ae 20 9e b7 e0 25
c070 : 00 d0 0d 78 a9 aa a0 c0 1f
c078 : 8d 14 03 8c 15 03 58 60 ed
c080 : 78 a9 bf a0 c0 8d 14 03 9f
c088 : 8c 15 03 58 60 a2 00 a9 d9
c090 : 58 9d 00 04 e8 d0 fa 9d 74
c098 : 00 05 e8 d0 fa 9d 00 06 18
c0a0 : e8 d0 fa 9d e8 06 e8 d0 67
c0a8 : fa 60 20 e7 c0 d0 35 e6 0d
c0b0 : fa a6 fa e0 0b d0 05 a9 77
c0b8 : 03 aa 85 fa 4c d1 c0 20 68
c0c0 : e7 c0 d0 20 c6 fa a6 fa 14
c0c8 : e0 ff d0 05 a9 07 aa 85 05
c0d0 : fa a0 00 8a 0a 0a aa da
c0d8 : bd 07 c1 99 c0 3a e8 c8 cf
c0e0 : c0 08 d0 f4 4c 31 ea e6 3f
```

```
c0e8 : fe a5 fe cd 05 c1 f0 03 5a
c0f0 : a9 0a 60 a9 00 85 fe 60 d5
c0f8 : 78 a9 31 a0 ea 8d 14 03 17
c100 : 8c 15 03 58 60 00 00 00 3c 61
c108 : 3c 18 00 00 00 00 00 18 3c 29
c110 : 18 00 00 00 00 00 18 3c 18 0a
c118 : 00 00 00 00 00 00 18 3c 3c 00 6d
c120 : 00 00 00 18 3c 3c 18 00 00 2a
c128 : 00 00 18 3c 3c 18 00 00 00 3b
c130 : 00 18 3c 3c 18 00 00 00 00 55
c138 : 18 3c 3c 18 00 00 00 00 18 b1
c140 : 3c 3c 18 00 00 00 00 00 3c 19
c148 : 3c 18 00 00 00 00 18 3c 69
c150 : 18 00 00 00 00 00 18 3c 18 4a
c158 : 00 00 00 00 00 18 3c 3c 00 ad
```

Listing 1. Das Stenoprogramm zum HiRes-Scrolling. Bitte mit dem MSE eingeben.

sind nur die Werte 1 bis 255, wobei 1 die höchste Geschwindigkeit bewirkt. Der Parameter b kann die Werte 0 oder 1 annehmen. Ist b=0, so wird nach dem Aufruf der gesamte Bildschirm mit dem zu scrollenden Zeichen aufgefüllt. Das ist zu Anfang ein kleiner Kreis, kann aber von Ihnen im Quelltext geändert werden. Ist b=1, so wird der Bildschirm nicht aufgefüllt. Dies wird zum Beispiel in dem Demoprogramm ausgenutzt.

Der Parameter c kann ebenfalls die Werte 0 oder 1 annehmen. Ist c=0, so wird nach oben gescrollt, ist c=1, so wird nach unten gescrollt. Abgeschaltet wird das Ganze mit SYS 49400. Am besten starten Sie zu Anfang das Demoprogramm (Listing 2), um sich mit den Befehlen vertraut zu machen. Zuvor müssen Sie aber das Maschinenprogramm absolut (mit »8,1«) geladen haben.

Zur Funktionsweise:

Zuerst wird der Zeichensatz umkopiert und einige Zeiger initialisiert. Der Trick beim Scrolling besteht darin, daß im Interrupt kontinuierlich das Zeichen »SHIFT-X« (Bildschirmcode=88) umdefiniert wird. Dadurch scheint der Hintergrund gescrollt zu werden. Unterbrechen Sie doch einmal das Demoprogramm mit RUN/STOP, löschen Sie den Bildschirm und läsen das Programm. Sie werden sehen, daß sogar »im Listing« die Zeichen gescrollt werden.

Hinweis: Wenn Sie die neue Version vom »SMON« besitzen (64'er, Ausgabe 10/85), dann können Sie, nachdem Sie den SMON mit »Y8000« verschoben haben, das zu scrollende Zei-

chen beliebig mit dem »Z«-Befehl umdefinieren. Die Zeichendefinitionen liegen ab \$C107.

(Bergerhoff/Nikolas/tr)

1520-Plotter als vollwertiger Drucker

Dieses kleine Programm (Listing 3) schickt alle Druckausgaben, die zur Gerätedresse 4 gesendet werden, automatisch zum 1520-Plotter mit der Gerätedresse 6. Das heißt, daß fertige Programme nicht mehr mühselig nach OPEN-Befehlen durchkämmt werden müssen, um die Druckerausgaben entsprechend umzulenken. Weiterhin wird der Plotter automatisch auf Kleinschrift gestellt, wenn diese am C 64 eingeschaltet ist.

Auch läßt sich die Zeilenlänge am Plotter einstellen: »POKE 713,Z:POKE 754,Z«, wobei folgende Werte für Z zulässig sind:

- 0: 80 Zeichen/Zeile (eingestellter Wert)
- 1: 40 Zeichen/Zeile
- 2: 20 Zeichen/Zeile

(N.J.W. Cuppen/tr)

Kalauer des Monats:

»God Save the Queen Komma acht Komma eins!«

```

0 REM ****
1 REM ** PSEUDO-SCROLL-DEMO **
2 REM ***
3 REM ** (C) 1985 BY **
4 REM ** N. BERGERHOFF **
5 REM ** MEISENWEG 22 **
6 REM ** 5020 FRECHEN 4 **
7 REM ** TEL. 02234/64610 **
8 REM ****
9 :
10 INIT=49152:EIN=49232:AUS=49400
11 SYS INIT
12 :
100 PRINT"CLR,WHITE,5SPACE>XXX<SPACE>XXX
      XXXX <(2SPACE)>X XXX<(2SPACE)>XXXX <038>
110 PRINT"<(5SPACE)>X<(2SPACE)>X <(4SPACE)>X<4S
      PACE>X<(2SPACE)>X <(2SPACE)>X <(2SPACE)>X <137>
120 PRINT"<(5SPACE)>X<(2SPACE)>X <(4SPACE)>X<4S
      PACE>X<(2SPACE)>X <(2SPACE)>X <(2SPACE)>X <147>
130 PRINT"<(5SPACE)>XXX<(3SPACE)>XX<(2SPACE)>XXX
      <(2SPACE)>X<(2SPACE)>X <(2SPACE)>X <(2SPACE)
      >X <200>
140 PRINT"<(5SPACE)>X<(7SPACE)>X <(4SPACE)>X<(2S
      PACE)>X <(2SPACE)>X <(2SPACE)>X <140>
150 PRINT"<(5SPACE)>X<(7SPACE)>X <(4SPACE)>X<(2S
      PACE)>X <(2SPACE)>X <(2SPACE)>X <150>
160 PRINT"<(5SPACE)>X<(4SPACE)>XXX<(2SPACE)>XXXX
      <(2SPACE)>X<(2SPACE)>XXX<(2SPACE)>XXXX <156>
170 PRINT
180 PRINT"<(6SPACE)>XXX<(2SPACE)>XXX XXX<(2SPAC
      E>XXXX <(4SPACE)>X <093>
190 PRINT"<(5SPACE)>X<(4SPACE)>X<(4SPACE)>X<(2SP
      ACE)>X <(2SPACE)>X <(4SPACE)>X <049>
200 PRINT"<(5SPACE)>X<(4SPACE)>X<(4SPACE)>X<(2SP
      ACE)>X <(2SPACE)>X <(4SPACE)>X <059>
210 PRINT"<(6SPACE)>XX<(2SPACE)>X<(4SPACE)>XXX<2
      SPACE>X<(2SPACE)>X <(4SPACE)>X <237>
220 PRINT"<(8SPACE)>X <(4SPACE)>X<(2SPACE)>X <(2
      SPACE)>X <(4SPACE)>X <079>
230 PRINT"<(8SPACE)>X <(4SPACE)>X<(2SPACE)>X <(2
      SPACE)>X <(4SPACE)>X <089>
240 PRINT"<(5SPACE)>XXX<(3SPACE)>XXX <(2SPACE)>
      XXXX XXXX XXXX <169>

```

Listing 2. Ein Demo-Programm zum HiRes-Scrolling.

```

250 SYS EIN,3,1,0 <094>
260 GOSUB 1000:PRINT" (LIG.BLUE,CLR)" <252>
270 SYS EIN,3,0,0 <114>
280 PRINT TAB(11)"<3DOWN,WHITE>PSEUDO-SCRO
      LL-DEMO" <007>
290 PRINT TAB(17)"<4DOWN>HOCH" <125>
300 GOSUB 1000 <002>
310 SYS EIN,3,1,1 <164>
320 PRINT TAB(16)"<UP>RUNTER" <135>
330 GOSUB 1000 <032>
340 SYS EIN,1,1,1 <193>
350 PRINT TAB(16)"<UP>SCHNELL" <132>
360 GOSUB 1000 <062>
370 SYS EIN,10,1,1 <216>
380 PRINT TAB(16)"<UP>LANGSAM" <203>
390 GOSUB 1000 <092>
400 PRINT" (LIG.BLUE,CLR)":SYS EIN,3,0,0 <095>
410 PRINT TAB(12)"<WHITE>TASTE<LIG.BLUE><W
      HITE>DRUECKEN" <096>
420 WAIT 198,255:POKE 198,0:END <126>
430 :
1000 FOR I=1 TO 3000:NEXT:RETURN <132>

```

© 64'er

Listing 2. »Demo« HiRes-Scrolling (Schluß)

```

10 DIM H(75):FOR I=0 TO 9 <139>
20 H(48+I)=I:H(65+I)=I+10:NEXT <067>
30 FOR I=679 TO 764:READ A$ <072>
40 H=ASC(A$):L=ASC(RIGHT$(A$,1)) <132>
50 D=H(H)*16+H(L):POKE I,D <165>
60 NEXT:POKE 794,167:POKE 795,2 <080>
100 DATA A5,B8,B5,F8,A5,BA,C9,04,F0,03,4C,
      4A,F3,A9,03,B5,B9,A9,06,85 <241>
110 DATA BA,A9,7F,B5,B8,20,4A,F3,A6,B8,20,
      C9,FF,A9,00,20,D2,FF,A5,B8 <091>
120 DATA 20,C3,FF,A9,66,C5,B9,F0,15,85,B9,
      A9,7E,B5,B8,AD,18,D0,29,02 <039>
130 DATA 4A,18,69,30,8D,C9,02,4C,C0,02,A9,
      00,85,B9,A9,00,8D,C9,02,A5 <241>
140 DATA FB,B5,BB,4C,4A,F3 <041>

```

© 64'er

Listing 3. Der 1520-Plotter als vollwertiger Drucker.