

```

6090 SZ=PEEK(I) <103>
6100 PRINT#2,SZ <089>
6110 NEXT I <192>
6120 CLOSE 2:GOTO 40 <248>
7000 REM ***** <124>
7001 REM * LADEN * <187>
7002 REM ***** <126>
7010 PRINT "[CLEAR]":PRINT:PRINT "[SPACE]LADEN
[SPACE]EINS[SPACE]ZEICHENSATZES" <074>
7020 PRINT:PRINT "(INPUT[SPACE]'N', [SPACE]WENN
[SPACE]ZURUECK)" <065>
7030 PRINT:PRINT:PRINT "[SPACE]NAME [SPACE]DER
[SPACE]DATEI?" <182>
7040 INPUT N$ <146>
7050 IF N$="N" THEN 40 <241>
7060 OPEN 2,8,2,N$+ ".S,R" <203>
7070 CLOSE 1:OPEN 1,8,15:INPUT#1,OA
: IF OA<>0 THEN 7010 <141>
7080 FOR I=51200 TO 53248 <225>
7090 INPUT#2,SZ <039>
7100 POKE I,SZ <091>
7110 NEXT I <172>
7120 CLOSE 2:GOTO 40 <228>
8000 REM ***** <226>
8001 REM * DATAS EINGEBEN * <237>
8002 REM ***** <228>
8010 PRINT "[CLEAR]":PRINT:PRINT "[SPACE]DATEN
[SPACE]FUER [SPACE]EINS[SPACE]ZEICHENS[SPACE]
EINGEBEN" <128>
8020 FOR I=1 TO 8:PRINT I". [SPACE]BYTE"
: INPUT B(I):C(I)=B(I) <254>
8030 IF B(I)>255 OR B(I)<0 THEN I=I-1 <103>
8040 NEXT:FL=1:GOSUB 40:GOTO 4010 <107>
10000 REM ***** <094>
10001 REM * MASCHINENROUTINE-EINGABE * <239>
10002 REM ***** <096>
10010 FOR I=0 TO 58:READ A$ <036>
10020 HN=(ASC(LEFT$(A$,1))-48)*16
: IF HN>144 THEN HN=HN-112 <105>
10030 LN=ASC(RIGHT$(A$,1))-48
: IF LN>9 THEN LN=LN-7 <099>
10040 HN=HN+LN:POKE(828+I),HN:NEXT:RETURN <126>
10100 DATA 7B,A9,31,85,01,A9,00,85,64,A9 <240>
10110 DATA D0,85,63,A9,C8,85,65,A2,10,A0 <016>
10120 DATA 00,B1,62,91,64,C8,D0,F9,E6,63 <031>
10130 DATA E6,65,CA,D0,F2,A9,37,85,01,58 <061>
10140 DATA A9,12,8D,18,D0,A9,94,8D,00,DD <083>
10150 DATA A9,C4,8D,8B,02,20,44,E5,60 <153>

```

Listing »Zeichen-Editor«

Super Line — 80 Zeichen für den C 64

Ein kleiner Basic-Lader realisiert, wofür man sonst viel Geld ausgeben muß: 80 Zeichen pro Zeile

Es werden 4 neue Befehle definiert, die das Darstellen von 80 Zeichen möglich machen. Und das, ohne daß man lange programmieren muß.

Beginnen wir mit dem Einfachsten: dem Eingeben. Dies dürfte keine Schwierigkeiten bereiten. Bevor man das Quellprogramm jedoch startet, sollte man es abspeichern, da es sich, vorausgesetzt, man hat keinen Fehler gemacht, selbst löscht. Anschließend kann man es mit RUN laufen lassen und wenn die Prüfsumme stimmt, erscheint nach einigen Sekunden einfach READY. Die neuen Befehle sind nun definiert und können angewendet werden.

Diese Befehle lauten:

- O (für »ON«): dieser Befehl bewirkt ein Einschalten des 80-Zeichen Modus. Dabei wird auf hochauflösende Grafik umgeschaltet.
- C (für »CLEAR«): der 80-Zeichen Bildschirm wird gelöscht.
- W x,y,a\$ (für »WRITE«): der String A\$ wird an Spalte x Zeile y geschrieben.
x geht von 0 bis 79,
y geht von 0 bis 24.
- F (für »OFF«): Abschalten des 80-Zeichen Modus.

So bewirkt zum Beispiel das kurze Programm:

```
10 O
20 C
```

30 W 0,0,"64'ER DAS MAGAZIN FÜR COMPUTER-FANS" daß der in Anführungsstrichen stehende Satz in die linke obere Ecke geschrieben wird. Anschließend rührt sich nichts mehr und man kann durch die 'blinde' Eingabe von 'F' wieder zum normalen Bildschirm zurückkehren.

A\$ läßt sich auch durch einzelne Stringvariable ersetzen, die mit + verknüpft werden, oder man kann auch eine normale numerische Variable verwenden. Allerdings dürfen keine Variable mit dem Namen O,C,W oder F verwendet werden. So ist zum Beispiel O\$,CG,WR\$ oder ähnliches verboten.

Nun einige detaillierte Angaben zum Programm selbst. Das Maschinenprogramm liegt im Bereich von \$ 9000 bis \$ 928F. Wer im Besitz eines Monitors ist, kann es direkt abspeichern und von der Diskette mit ,8,1 laden (danach »NEW« eingeben). Gestartet wird es dann mit SYS 36864. Wer das nicht will, der läßt einfach den Basic-Lader, der das Maschinenprogramm in den Speicher 'POKE'. Zwischen \$ 9000 und \$ 902D wird zunächst der Basic-Vektor umgesetzt und der Anfang des Basic-Speichers hochgelegt. Zwischen \$ 902E und \$ 9044 beginnt nun die Befehlsdecodierung. Bei Erkennen eines Befehls wird verzweigt, ansonsten in die normale Interpreterroutine gesprungen. Bei dem Befehl 'O' wird zunächst in der Unteroutine von \$ 9233 bis 928E der Zeichensatz aus dem verdeckten Bereich \$ D000 in den offenen Bereich \$ C000 übertragen. Der Bereich von \$ 0400 bis \$ 0800 wird mit dem Code für die Hintergrundfarbe gefüllt. Außerdem wird bei \$ 9057 das Register # 648 umgesetzt, damit es auf dem Bildschirm kein farbliches Durcheinander gibt. Weiterhin wird selbstverständlich der hochauflösende Grafik-Modus eingeschaltet. Die Routine für den Befehl 'C' liegt zwischen \$ 9081 und \$ 90A3. Der Bereich der Bit-Map wird einfach mit 00 gefüllt.

Der Befehl 'F' wird zwischen \$ 906C und \$ 907E ausgeführt. Das Register 648 wird zurückgesetzt, der hochauflösende Grafik-Modus ausgeschaltet und der normale Bildschirm gelöscht.

Der Befehl, dessen Routine am längsten ist, ist der Befehl 'W'. Er wird zwischen \$ 90A6 und \$ 9230 bearbeitet. Zunächst werden die beiden Koordinaten x und y geholt und aus ihnen die Adresse der Bit Map berechnet, an der das erste Byte gesetzt wird. Dies geschieht zwischen \$ 90A6 und 9135. Dann werden die einzelnen Zeichen des zu schreibenden Satzes geholt und ihr Code wird so umgerechnet, daß er mit der Stelle übereinstimmt, an der das jeweilige Zeichen in dem nach \$ C000 verschobenen Zeichen ROM steht. Anschließend durchläuft jedes der 8 Bytes, aus denen ein Zeichen definiert ist, die gleiche Prozedur. Das Byte wird geholt, jedes zweite Bit ausgefiltert, und die verbliebenen 4 Bits zusammengescho-ben. Das Zeichen ist jetzt nur noch durch 4 x 8 Punkte definiert. Jetzt müssen die entstandenen Nibbles noch in die Bytes der Bit Map gebracht werden. Dies geschieht mit einer EXOR-Verknüpfung. Dabei steuert ein Flag, das in \$ 9300 steht, ob das Nibble in die linke oder die rechte Hälfte des Bytes geschrieben wird.

(Andreas Zell / rg)

```

1 REM *****
2 REM ****          SUPER LINE          ****
3 REM ****          ANDREAS ZELL        ****
4 REM ****          SEGEBERGERSTR. 27   ****
5 REM ****          8500 NUERNBERG 90   ****
6 REM ****          TEL 0911/31 47 90   ****
7 REM *****
12 FOR T=36864 TO 37518:READ A:POKE T,A:B=B+A:NEXT
13 IF B<>87028 THEN PRINT"FEHLER[SPACE]IN[SPACE]DEN[SPACE]DATA
   [SPACE]ZEILEN":END
14 SYS 36864:END
100 DATA 169, 46, 141, 8, 3, 169, 144, 141, 9, 3
110 DATA 169, 1, 133, 43, 133, 45, 133, 47, 133, 49
120 DATA 169, 68, 133, 44, 133, 46, 133, 48, 133, 50
130 DATA 169, 0, 141, 0, 68, 141, 1, 68, 141, 2
140 DATA 68, 169, 143, 133, 56, 96, 32, 115, 0, 201
150 DATA 79, 240, 15, 201, 70, 240, 14, 201, 87, 240
160 DATA 13, 201, 67, 240, 12, 76, 231, 167, 76, 81
170 DATA 144, 76, 108, 144, 76, 166, 144, 76, 129, 144
180 DATA 234, 32, 51, 146, 234, 234, 234, 169, 64, 141
190 DATA 136, 2, 32, 68, 229, 169, 59, 141, 17, 208
200 DATA 169, 24, 141, 24, 208, 76, 228, 167, 169, 4
210 DATA 141, 136, 2, 169, 27, 141, 17, 208, 169, 21
220 DATA 141, 24, 208, 32, 68, 229, 76, 228, 167, 169
230 DATA 32, 133, 252, 169, 0, 133, 251, 162, 0, 169
240 DATA 0, 129, 251, 24, 165, 251, 105, 1, 133, 251
250 DATA 165, 252, 105, 0, 133, 252, 201, 64, 240, 3
260 DATA 76, 137, 144, 76, 228, 167, 32, 155, 183, 224
270 DATA 80, 144, 3, 76, 72, 178, 138, 74, 133, 251
280 DATA 138, 41, 1, 240, 11, 234, 234, 234, 234, 169
290 DATA 1, 141, 0, 147, 208, 5, 169, 0, 141, 0
300 DATA 147, 32, 253, 174, 24, 32, 158, 183, 134, 252
310 DATA 224, 25, 176, 3, 76, 220, 144, 76, 72, 178
320 DATA 169, 0, 133, 253, 133, 254, 165, 252, 133, 253
330 DATA 162, 0, 6, 253, 38, 254, 232, 224, 5, 208
340 DATA 247, 162, 0, 142, 1, 149, 165, 252, 141, 0
350 DATA 149, 14, 0, 149, 46, 1, 149, 232, 224, 3
360 DATA 208, 245, 24, 165, 253, 109, 0, 149, 133, 253
370 DATA 165, 254, 109, 1, 149, 133, 254, 24, 165, 253
380 DATA 101, 251, 133, 251, 165, 254, 105, 0, 133, 252
390 DATA 24, 162, 0, 6, 251, 38, 252, 232, 224, 3
400 DATA 208, 247, 24, 165, 252, 105, 32, 133, 252, 32
410 DATA 253, 174, 32, 158, 173, 36, 13, 48, 6, 32
420 DATA 221, 189, 32, 135, 180, 32, 166, 182, 134, 253
430 DATA 132, 254, 133, 255, 160, 0, 177, 253, 201, 191
440 DATA 144, 6, 56, 233, 128, 76, 109, 145, 201, 63
450 DATA 144, 6, 56, 233, 64, 76, 109, 145, 234, 234
460 DATA 234, 234, 234, 234, 234, 153, 0, 148, 200, 198
470 DATA 255, 240, 3, 76, 80, 145, 169, 0, 153, 0
480 DATA 148, 234, 234, 234, 234, 234, 160, 0, 185, 0
490 DATA 148, 208, 8, 234, 234, 234, 234, 234, 76, 174
500 DATA 167, 133, 253, 169, 0, 133, 254, 6, 253, 38
510 DATA 254, 6, 253, 38, 254, 6, 253, 38, 254, 24
520 DATA 165, 254, 105, 192, 133, 254, 162, 0, 142, 1
530 DATA 147, 161, 253, 41, 1, 133, 255, 161, 253, 41
540 DATA 4, 74, 69, 255, 133, 255, 161, 253, 41, 16
550 DATA 74, 74, 69, 255, 133, 255, 161, 253, 41, 64
560 DATA 74, 74, 74, 69, 255, 133, 255, 173, 0, 147
570 DATA 201, 1, 240, 8, 6, 255, 6, 255, 6, 255

```

580 DATA	6, 255, 165, 255, 65, 251, 129, 251, 238, 1	<100>
590 DATA	147, 24, 165, 251, 105, 1, 133, 251, 165, 252	<192>
600 DATA	105, 0, 133, 252, 24, 165, 253, 105, 1, 133	<089>
610 DATA	253, 165, 254, 105, 0, 133, 254, 173, 1, 147	<164>
620 DATA	201, 8, 240, 3, 76, 175, 145, 173, 0, 147	<028>
630 DATA	201, 1, 208, 7, 200, 142, 0, 147, 76, 132	<020>
640 DATA	145, 169, 1, 141, 0, 147, 56, 165, 251, 233	<149>
650 DATA	8, 133, 251, 165, 252, 233, 0, 133, 252, 200	<193>
660 DATA	76, 132, 145, 120, 169, 0, 133, 251, 133, 253	<005>
670 DATA	169, 208, 133, 252, 169, 192, 133, 254, 169, 51	<141>
680 DATA	133, 1, 162, 0, 161, 251, 129, 253, 24, 165	<178>
690 DATA	251, 105, 1, 133, 251, 165, 252, 105, 0, 133	<227>
700 DATA	252, 24, 165, 253, 105, 1, 133, 253, 165, 254	<049>
710 DATA	105, 0, 133, 254, 201, 200, 208, 222, 169, 55	<046>
720 DATA	133, 1, 88, 169, 0, 133, 251, 169, 4, 133	<131>
730 DATA	252, 162, 0, 169, 1, 129, 251, 24, 165, 251	<236>
740 DATA	105, 1, 133, 251, 165, 252, 105, 0, 133, 252	<022>
750 DATA	201, 8, 208, 235, 96	<145>

Tastaturpieps

Bei einem Besuch in einer Bank sah ich dort einige größere Computer, die jeden Tastendruck des Bedieners mit einem Piepston quittierten. Dies wollte ich auch beim C 64 nachvollziehen.

Dies sollte unabhängig von einem anderen Programm sein. Daher meine Idee, die Interruptroutine des C 64 zu verändern, da diese 60mal in der Sekunde angesprochen wird, um die Tastatur abzufragen. Dies konnte natürlich nur in Maschinensprache geschehen, da Basic zu langsam wäre. Es ist auch möglich, schon vorhandene Programme damit zu erweitern. Denkbar sind zum Beispiel Textprogramme, Spiele, Programme zum Erlernen des Schreibmaschinenschreibens oder auch nur zur Simulation einer echten Schreibmaschine, die ja auch nicht gerade geräuschlos arbeitet.

Zu Beginn des Programms (\$033C-\$0348) wird der Interruptvektor verbogen. Er zeigt jetzt auf unsere Routine (\$0349-\$039E). Da der Computer alle 1/60-Sekunde die Tastatur abfragt, und dazu einen Interrupt auslöst, wird unsere Routine ebenfalls so häufig angesprochen. In dieser Routine wird zuerst überprüft ob überhaupt eine Taste betätigt wurde. Dazu wird der Wert der Speicherstelle \$CB in den Akkumulator geladen. Enthält diese den Wert \$40, so wurde keine Taste gedrückt und es wird zur normalen Interruptroutine des C 64 gesprungen. Der Wert der Speicherstelle \$CB wird wischenge speichert. Man kann damit bei einem 2. Durchlauf dieser Routine feststellen, ob eine Taste nicht kurz zuvor ($t < 1/60$ sec.) schon einmal betätigt wurde. Dies dient dazu, daß bei Tasten mit Dauerfunktion nicht unaufhörlich der Pieps ertönt. Denn dies ist erstens entnervend und zweitens verzögert dies den

Ablauf der Dauerfunktion so, daß man dabei einschlafen kann. Danach werden die verschiedenen Toneinstellungen vorgenommen. Der Verzögerungsteil (\$037F-\$0387) dient dazu, den Ton höher zu machen. Ansonsten wäre nur ein Knacken zu hören. Sie können also selbst damit experimentieren und einen für Sie angenehmen Ton einstellen. Das Programm läßt sich durch Drücken der Run/Stop- und der Restore-Taste unterbrechen und mit SYS 49152 reaktivieren.

(Wolfgang Roth / rg)

```

1 DATA 120,162,13,160,192,142,20,3,140,21,3,88,
  96,165,203,201,64,208,8,169 <097>
2 DATA 203,141,0,193,76,49,234,165,203,205,0,
  193,208,3,76,49,234,169,15 <227>
3 DATA 141,24,212,169,3,141,5,212,169,242,141,6,
  212,169,26,141,1,212,169 <255>
4 DATA 5,141,0,212,169,33,141,4,212,160,69,162,
  255,202,208,253,136,208,248 <105>
5 DATA 169,0,141,4,212,141,5,212,141,6,212,165,
  203,141,0,193,76,49,234,0 <243>
6 DATA 0,0 <021>
10 FOR A=1 TO 99 <197>
20 READ B : S=S+B : POKE 49151+A,B <137>
30 NEXT A <225>
40 IF S <> 12111 THEN PRINT CHR$(147)
  : PRINT "FEHLER[SPACE]IN[SPACE]DATAS[SPACE]!"
  : END <000>
50 PRINT CHR$(147) : PRINT "OK[SPACE]!"
  : SYS 49152 <046>
60 PRINT "[DOWN]DIES[SPACE]IST[SPACE]DIE[SPACE]
  VERSION,WELCHE[SPACE]AB[SPACE]#C000"
  : PRINT "[DOWN]GESPEICHERT[SPACE]WIRD[SPACE]!";
  <130>
70 PRINT "[SPACE]UNTERBROCHEN[SPACE]WIRD"
  : PRINT "[DOWN]SIE[SPACE]MIT[SPACE]RUN/STOP
  [SPACE]&[SPACE]RESTORE[SPACE]TASTENDRUCK
  [SPACE]!" <248>
80 PRINT "[DOWN]RESTART[SPACE]MIT[SPACE]SYS49152
  [SPACE]!" <114>
90 END <218>

```