

# Automatische Zeilennumerierung

Die lästige Arbeit, vor jeder Zeile die entsprechende Nummer einzutippen, nimmt Ihnen diese kleine Routine ab.

Die Maschinen-Routine — ganze 141 Bytes — wird vom Basic-Lader über DATAs in den Speicherbereich ab Adresse \$C000 beziehungsweise 49152 Dezimal geschützt abgelegt. Dabei führt das Basic-Programm eine Checksummenprüfung durch, da sich der Basic-Lader nach Ablauf von selbst löst.

Vom Programm aus wird eine Schrittweite von 10 eingestellt. Diese kann aber durch Verändern des Wertes in Adresse 49296 — im Bereich von 1 bis 255 — eingestellt werden. Der Wert der ersten Zeilennummer wird in die Adressen: 49294 (High-Byte) und 49295 (Low-Byte) eingepoket, wenn Sie mit einem anderen Wert als 10 beginnen soll.

Beispiel: POKE 49296,S:POKE 49294,INT(Z/256):POKE 49295,Z-(INT(Z/256)\*256)

S = Schrittweite ; Z = Wert der ersten Zeilennummer

Durch Eingabe von Pfeil-links/Return kann nun eine Zeilennummer angefordert werden. Die Vorgabe einer Zeilennummer wird akustisch untermauert. Nun kann eine Basic-Zeile wie gewohnt eingegeben und mit der Return-Taste abgeschlossen werden. Die nächste Ausgabe erfolgt wieder auf Anforderung, mit der eingestellten Schrittweite.

(Thomas Schulz/rg)

```

10 REM      *** AUTONUM 64 ***          <040>
11 REM                                           <154>
12 REM      AUTOMATISCHE                <034>
13 REM      ZEILENNUMMERNVORGABE       <139>
14 REM                                           <157>
15 REM                                           <158>
16 REM      1. ZEILENNR. LOW -BYTE IN 49294 <137>
17 REM      1. ZEILENNR. HIGH-BYTE IN 49295 <185>
18 REM      ZEILENABSTAND (1-255) IN 49296 <085>
19 REM                                           <162>
20 REM                                           <163>
21 REM      EIN UTILITY VON:            <225>
22 REM                                           <165>
23 REM      TH. SCHULZ                  <073>
24 REM      POSTFACH 602542             <050>
25 REM      2000 HAMBURG 60             <214>
26 REM      TEL. 040/6316055           <233>
27 REM                                           <170>
28 REM                                           <171>
29 POKE 49294,0:POKE 49295,0:POKE 49296,10 <194>
30 POKE 55,0:POKE 56,195:AD=49152:Z=0 <110>
31 POKE 53280,0:POKE 53281,0:PRINT CHR$(30); <205>
                                           <077>
32 PRINT CHR$(147);                       <136>
33 READ X:IF X=-1 THEN 51                 <184>
34 POKE AD,X:AD=AD+1:Z=Z+X:GOTO 33        <094>
35 DATA 32,115,0,201,95,240,6,32,121,0,76 <237>
36 DATA 231,167,32,115,0,24,173,144,192 <157>
37 DATA 109,142,192,141,142,192,144,3,238 <007>
38 DATA 143,192,162,23,160,0,24,32,10,229 <247>
39 DATA 173,143,192,174,142,192,32,205 <120>
40 DATA 189,169,152,133,251,169,7,133,252 <027>
41 DATA 169,119,133,253,169,2,133,254,160 <019>
42 DATA 0,177,251,201,32,240,6,145,253 <108>
43 DATA 200,76,66,192,132,198,160,0,162,23 <062>
44 DATA 24,32,10,229,169,15,141,24,212 <111>
45 DATA 169,207,141,7,212,169,34,141,8 <128>

```

```

46 DATA 212,169,240,141,13,212,169,17,141 <009>
47 DATA 11,212,162,64,160,255,136,208,253 <013>
48 DATA 202,208,248,169,0,141,11,212,76 <169>
49 DATA 128,164,169,0,141,8,3,169,192 <087>
50 DATA 141,9,3,96,-1 <052>
51 IF Z<>18097 THEN 55 <150>
52 PRINT CHR$(147)CHR$(17); <060>
53 PRINT"AUTONUM[SPACE]64[SPACE2]ZEILENNR.VORG. <122>
   :[SPACE2]_";
54 PRINT CHR$(18)"RETURN":SYS 49283:NEW <050>
55 PRINT CHR$(17)"FEHLER[SPACE]INI[SPACE]DATAS! <077>
56 PRINT CHR$(17)"CHECKSUMME[SPACE3]"CHR$(18); <054>
57 PRINT"[SPACE]IST:"CHR$(146)Z CHR$(18); <009>
58 PRINT TAB(15)"SOLL:"CHR$(146)18097:END <051>

```

## Worktool — eine Programmierhilfe

Drei nützliche Optionen zur Programmeingabe werden durch dieses Programm realisiert. Farbwahl, aktuelle Zeitangabe und Warnton bei mehr als 80 Zeichen Eingabe auf Tastendruck.

Wie der Name schon sagt, hilft das »Worktool« dem Programmierer bei der Eingabe seiner eigenen Programme. Mehrere Optionen, die ansonsten nur bei erheblich teureren Geräten oder Hardware—Erweiterungen zu finden sind, werden ohne Basicspeicherverlust vom Toolkit übernommen.

Mittels Funktionstasten werden folgende Erweiterungen angesprochen:

— F1: Rahmen, Innenteil und Cursor erscheinen in einer vorher von Ihnen festgelegten Farbkombination.

Beim Schreiben von Programmen oder deren Test wird die Farbkombination, die Sie zur Eingabe verwenden, oft durch das Programm geändert. Haben Sie die Worktoolerweiterung eingegeben, können Sie sich jetzt, nach dem Probelauf 20 Tastendrucke ersparen um Farbkombinationen, wie zum Beispiel braun-rot-violett in Ihre Lieblingskombination zu verwechseln.

— F3: Ausgabe der aktuellen Uhrzeit, oder einer Stoppuhr. Wollen Sie wissen wieviel Uhr es ist, oder wie lange Sie schon mit dem Toolkit arbeiten, so genügt ein Druck auf F3. Wie sonst nur bei Steckmodulen der oberen Preisklasse wird vom Computer in der Mitte der 1. Zeile die Zeit in reverser Darstellung ausgegeben. Jedoch nicht permanent, sondern nur solange die Taste gedrückt ist. Das hat den Vorteil, daß keine Bildschirmzeile verloren geht.

— F5 und F7: Falls der eine oder andere von Ihnen schon einmal am CBM 4032 gearbeitet hat, wird er bestimmt bemerkt haben, daß dieser eine praktische Eingabehilfe besitzt. Er gibt, wenn der Cursor eine bestimmte Spalte erreicht, einen Piepser aus. Diese Option wurde beim Worktool noch erheblich verbessert. Im Direktmodus gibt er jetzt immer einen hellen Ton (ähnlich Tastaturklick) aus, sobald der Cursor die Spalte 35 oder 75 erreicht. Im Programmeingabemodus dagegen wird bei 35 der Klick, bei 75 ein Sägezahnnton, und bei 80 ein auf-

und abschwelliger Ton erzeugt. Jeder kennt das ungeheuer langwierige und nervenzermürende Editieren einer falsch eingegebenen PRINT-Zeile. Gerade im Programmierausch schreibt man oft über die maximale Zeilenlänge hinaus, und muß später zeitaufwendig ausbessern. Ist mit F5 die Tonooption eingeschaltet, gibt der Computer rechtzeitig die oben genannten »Erinnerungstöne« aus. Mit F7 läßt sich diese Funktion ausschalten (hebt das Klima beim Mittagsschlaf Ihrer Frau).

Im Programm selbst wurde weitgehend auf Steuerzeichen verzichtet, und stattdessen der CHR\$ und POKE-Befehl verwendet. Die DATAs sind der Übersicht und der Arbeitersparnis halber in hexadezimaler Form einzugeben und werden von einer Routine automatisch umgerechnet. Das Programm gibt bei Fehleingabe einen Checksumfehler mit der Angabe des betreffenden Datenblocks aus. Die Funktionen einzelner Abschnitte sind in REM-Zeilen kurz umrissen. Das Basic-Programm benötigt 3k Speicher. Nachdem Sie Ihre Eingaben gemacht haben, löscht sich der Basic-Teil selbst und nur noch zirka 250 Bytes im abgeteilten Maschinensprachespeicher bei \$C000 sind nötig. Es ist also der gesamte Basic-RAM frei verfügbar.

(Ulrich Grothaus / rg)

10-110	Ausgabe von Kopf und Menü
120-140	Eingabe der gewünschten Farbkombinationen
220	Farbwerte für Maschinenprogramm merken
230	Eingabe der aktuellen Uhrzeit beziehungsweise wenn nur RETURN der Nullstellung der Stoppuhr
240	diese Funktion rechnet den Dezimalwert in den TOD-BCD-Wert um
250-280	Auswerten des Zeitstrings, Einlesen der Werte in die CIA's und initialisieren der TOD Register, sowie starten der Uhr
260-410	Einlesen der Vorroutine und des Hauptteils sowie Start der Vorroutine. Die GO-SUBS springen zum Umrechnen und Einpocketeil
490	Mit dem POKE 792, 193 wird das Low-byte des NMI-Interruptvektors auf einen RTI gelenkt. Dadurch ist die RESTORE-Taste ausgeschaltet, was ein Zurücksetzen der unteren Pages verhindert. Im Bedarfsfall könnten Sie dies durch Löschen der Zeile oder nachträglich mit POKE 792,71 rückgängig machen.
500-590	Kurze Erklärungen und Abfrage, ob sich das Basic-Programm löschen kann. Geben Sie »N« ein, beginnt es von vorne.
660-690	Ausgabe des Kopfes
700-710	Sobald eine Taste gedrückt wird, löscht sich der Bildschirm, der Basic-Teil, und das Maschinenprogramm wird gestartet. Ferner wird der Bildschirm gelöscht, so daß der gesamte Bildschirm zum Programmieren frei ist und kein Ausgabekopf stört.
710	Die Schleife dient auf recht unkonventionelle Weise dazu, das Basic-Programm zu löschen, aber trotzdem noch den SYS abzuarbeiten.
720-760	Einleseroutine für den 1. Maschinenspracheteil. Zeile 730 und 740 rechnen die hexadezimalen DATAs in Dezimalzahlen von 0 bis 255 um.

770-790	Diese DATA-Zeilen enthalten den 1. Teil des Maschinenprogramms.
800-840	Einleseroutine für den 2. Maschinenspracheteil.
850-1180	Diese DATA-Zeilen enthalten das Interruptprogramm.

**Programmbeschreibung**

\$ C000-C00D	Der Interruptvektor wird auf das Worktool zusätzlich zur normalen Interruptroutine des Betriebssystems jede sechzigstel Sekunde abgearbeitet wird.
\$ C00E-C03A	Hier wird entschieden ob aufgrund einer gedrückten Funktionstaste oder eines Registerwertes in ein Unterprogramm verzweigt werden soll. Am Ende dieses Teils beziehungsweise einer aufgerufenen Unteroutine wird die normale Interruptroutine angesprungen.
\$C040-C052	Falls F1 gedrückt ist, setzt diese Routine die Farbregister auf den eingestellten Wert, das heißt der Bildschirm erscheint in den von Ihnen gewünschten Farben.
\$ C057-C08E	Wenn F3 gedrückt ist, werden die TOD-Register ausgewertet und die Zeit aktualisiert beziehungsweise ausgegeben. Dies hat gegenüber ähnlichen Programmen den Vorteil daß Sie die Zeit bei ihrer Programmeingabe überschreiben können, was bei der Ausgabe der Zeit im Interrupt nicht möglich gewesen wäre. Sie könnten ansonsten die 1. Bildschirmzeilen nicht benutzen, weil das einen Syntax Error hervorrufen würde.
\$ C091-C0A6	Die derzeitige Cursorposition wird abgefragt und bei Bedarf, sofern Sie diese Option aktiviert haben, in eine Tonausgaberroutine gesprungen.
\$ COAD-C109	In diesem Teil werden die verschiedenen Töne ausgegeben.

**Aufbau der Maschinenroutine**

```

0 REMWORKTOOL <116>
1 REMULRICH GROTHAUS <040>
2 REMFALLWEG 18 <096>
3 REMB450 AMBERG <117>
4 : <062>
5 : <063>
10 REM ----- <057>
20 REM ABFRAGE DER OPTIONSWERTE <025>
30 REM ----- <077>
40 : <098>
50 : <108>
60 RESTORE:POKE 53280,0:POKE 53281,0 <039>
70 PRINT CHR$(147):POKE 646,5
:PRINT "*****[WHITE]WORKTOOL [GREEN]
*****" <096>
80 POKE 646,7:PRINT TAB(8)"VON[SPACE]ULRICH
[SPACE]IGROTHAUS" <044>
90 POKE 646,15:PRINT:PRINT:PRINT"OPTIONS
:":POKE 646,10 <126>
100 PRINT "F1=FARBE [SPACE]SETZEN"
:PRINT, "F3=ZEIT [SPACE]AUSGEBEN"
:PRINT, "F5=WARNTON [SPACE]EIN" <173>
110 PRINT, "F7=WARNTON [SPACE]AUS":PRINT:PRINT
:POKE 646,14 <235>
120 INPUT "FARBCODE [SPACE]RAHMEN";R <191>
130 INPUT "FARBCODE [SPACE]GRUND";G <131>
140 INPUT "FARBCODE [SPACE]CURSOR";C <231>
150 : <208>
160 : <218>
    
```

```

170 REM ----- <172>
180 REM WERTUEBERGABE MASCHINENPROGRAMM <002>
190 REM ----- <192>
200 : <002>
210 : <012>
220 POKE 49425,R:POKE 49426,G:POKE 49427,C <144>
230 PRINT:PRINT TAB(7);"HMMSS":PRINT CHR$(145);
:INPUT"ZEIT[SPACE]";Z$ <092>
240 DEF FN F(X)=INT(X/10)*16+(X-INT(X/10)*10)
<194>
250 H=VAL(LEFT$(Z$,2)):M=VAL(MID$(Z$,3,2))
:S=VAL(RIGHT$(Z$,2)) <239>
260 POKE 56335,PEEK(56335)AND 127 <048>
270 POKE 56329,FN F(S):POKE 56330,FN F(M)
:POKE 56331,FN F(H):POKE 56328,0 <169>
280 POKE 56334,PEEK(56334)OR 128 <068>
290 : <093>
300 : <103>
310 REM ----- <057>
320 REM EINLESEN DER DATEN <106>
330 REM ----- <077>
340 : <143>
350 : <153>
360 GOSUB 720 <143>
370 FOR T=1 TO 4:SYS 36500:NEXT <149>
380 FOR T=1 TO 40:PRINT:NEXT <204>
390 POKE 646,13:PRINT"DAS[SPACE2]EINLESEN
[SPACE2]DES[SPACE]MASCHINENPROGRAMMES[SPACE]
DAUERT[SPACE]15[SPACE]SEKUNDEN" <135>
400 GOSUB 800 <182>
410 FOR T=1 TO 4:SYS 36500:NEXT <189>
420 : <223>
430 : <233>
440 REM ----- <142>
450 REM BENUTZERHINWEISE <029>
460 REM ----- <162>
470 : <017>
480 : <027>
490 POKE 792,193:PRINT CHR$(147):POKE 646,7
<072>

500 PRINT"AUS[SPACE]PROGRAMMTECHNISCHEN[SPACE]
GRUENDEN[SPACE]IST[SPACE]DER[SPACE,WHITE]
NMI-INTERRUPT";:POKE 646,7 <097>
510 PRINT"[SPACE]AUSGESCHALTET (ZEILE[SPACE]
490)." <062>
520 PRINT"DAS[SPACE]WORKTOOL[SPACE]LIEGT[SPACE]
BEI[SPACE]49152." <184>
530 PRINT"ES[SPACE]SIND[SPACE]38911[SPACE]BYTES
[SPACE]BASIC-RAM[SPACE]FREE." <003>
540 POKE 53280,0:POKE 646,3:PRINT"[DOWN14]KANN
[SPACE]ICH[SPACE]MICH[SPACE]LOESCHEN[SPACE]
(J/N) ?" <077>
550 FOR T=1 TO 11:GET N$:NEXT <239>
560 GET A$:IF A$=""THEN 560 <154>
570 IF A$="N"THEN GOTO 60 <006>
580 : <128>
590 : <138>
600 REM ----- <092>
610 REM LOESCHEN DES BASICTEILS <003>
620 REM START DER TOOL-IRQ ROUTINE <227>
630 REM ----- <122>
640 : <188>
650 : <198>
660 POKE 53280,PEEK(49425):POKE 53281,
PEEK(49426):POKE 646,PEEK(49427) <076>
670 PRINT CHR$(147);"*****@WORKTOOL@
*****" <123>
680 PRINT"[SPACE7]----38634[SPACE]BYTES[SPACE]
FREE----":PRINT <160>
690 PRINT"*****1984[SPACE]BY[SPACE]
GROTO*****" <251>
700 GET A$:IF A$=""THEN 700 <034>
710 PRINT CHR$(147):FOR T=2048 TO 2060:POKE T,0
:NEXT:POKE 45,3:POKE 46,8:SYS 49152:END <028>
720 PS=0:FOR AD= 36500 TO 36519 :READ DA$ <093>
730 L=ASC(LEFT$(DA$,1))-48:R=ASC(RIGHT$(DA$,
1))-48 <179>
740 PO=(L+(L>9)*7)*16+R+(R>9)*7:POKE AD,PO
:PS=PS+PO:NEXT <105>
750 IF PS<> 2456 THEN PRINT"[CLEAR]
PRUEFSUMMENFEHLER[SPACE]IM[SPACE]1.[SPACE]
DATABLOCK.":END <217>
760 RETURN <136>
770 DATA 00,00,A2,00,8E,20,D0,E8 <048>
780 DATA F0,03,4C,98,8E,C8,F0,03 <088>
790 DATA 4C,96,8E,60 <233>

```

```

800 PS=0:FOR AD= 49152 TO 49420 :READ DA$ <144>
810 L=ASC(LEFT$(DA$,1))-48:R=ASC(RIGHT$(DA$,
1))-48 <004>
820 PO=(L+(L>9)*7)*16+R+(R>9)*7:POKE AD,PO
:PS=PS+PO:NEXT <186>
830 IF PS<> 32697 THEN PRINT"[CLEAR]
PRUEFSUMMENFEHLER[SPACE]IM[SPACE]2.[SPACE]
DATABLOCK.":END <101>
840 RETURN <217>
850 DATA EA,78,A9,0E,8D,14,03,A9 <174>
860 DATA C0,8D,15,03,58,60,8D,04 <129>
870 DATA D4,EA,A9,04,C5,C5,F0,28 <204>
880 DATA A9,05,C5,C5,F0,39,A9,06 <187>
890 DATA C5,C5,D0,03,8D,10,C1,A9 <198>
900 DATA 03,C5,C5,D0,03,8D,10,C1 <185>
910 DATA AD,10,C1,C9,06,F0,5A,4C <231>
920 DATA 31,EA,60,00,EA,EA,EA,EA <026>
930 DATA AD,11,C1,8D,20,D0,AD,12 <241>
940 DATA C1,8D,21,D0,AD,13,C1,8D <006>
950 DATA 86,02,4C,18,C0,EA,EA,AD <036>
960 DATA 0B,DC,29,7F,A2,10,20,78 <000>
970 DATA C0,AD,0A,DC,A2,13,20,78 <028>
980 DATA C0,AD,09,DC,A2,16,20,78 <033>
990 DATA C0,AD,08,DC,4C,31,EA,00 <064>
1000 DATA 48,29,F0,4A,4A,4A,4A,18 <044>
1010 DATA 69,80,9D,00,04,68,29,0F <029>
1020 DATA 18,69,80,9D,01,04,60,EA <046>
1030 DATA EA,A5,D3,C9,23,F0,11,A5 <099>
1040 DATA D3,C9,46,F0,2F,A5,D3,C9 <127>
1050 DATA 4F,F0,4D,4C,31,EA,60,00 <100>
1060 DATA EA,EA,EA,A9,06,8D,00,D4 <165>
1070 DATA A9,78,8D,01,D4,A9,0A,8D <142>
1080 DATA 18,D4,A9,33,8D,05,D4,A9 <135>
1090 DATA 00,8D,06,D4,A9,11,8D,04 <115>
1100 DATA D4,4C,31,EA,EA,EA,EA,A9 <236>
1110 DATA 06,8D,00,D4,A9,32,8D,01 <135>
1120 DATA D4,A9,0A,8D,18,D4,A9,63 <190>
1130 DATA 8D,05,D4,A9,00,8D,06,D4 <178>
1140 DATA A9,21,8D,04,D4,4C,31,EA <197>
1150 DATA A9,01,8D,03,D4,A9,0F,8D <215>
1160 DATA 18,D4,A9,FF,8D,06,D4,A9 <254>
1170 DATA 00,8D,05,D4,A9,41,8D,04 <197>
1180 DATA D4,4C,31,EA,00 <012>

```

