

Tips & Tricks für Anfänger und Profis

In der Rubrik Tips & Tricks können wir Ihnen diesmal einige Routinen anbieten, die sich besonders zum Einbau in eigene Programme eignen.

Screen-Dump

Eine Bildschirmmaske kann auf viele verschiedene Arten auf den Bildschirm gebracht werden. Die einfachste ist wohl:
FOR I = 0 TO 999 : POKE 1024 + I, PEEK (STADR + I) : NEXT I

wobei STADR die Startadresse der Maske ist. Doch das ist zu langsam und eintönig. Das gleiche Programm in Assembler ist zirka 200mal schneller, aber immer noch eintönig.

Das Programm »Screen-Dump« (siehe Listing 1) hilft dem ab. Auf über 100 verschiedene Arten kann hiermit das oben genannte Programm ablaufen (das heißt 100 verschiedene Reihenfolgen), von denen einige verblüffen, fast alle aber besser »aussehen« als dieses Programm. Auch die verstellbare Geschwindigkeit ermöglicht interessante Veränderungen.

Die Handhabung ist denkbar einfach:

SYS 680, Art, Stadr., Geschwindigkeit

Art: Eine Zahl zwischen 0 und 127

Stadr: Die Startadresse der Maske geteilt durch 256 (High-byte).

Geschwindigkeit: zwischen 1 und 255 verstellbar.

Also SYS 680, 20, 64, 8 kopiert den Bereich von 16384 bis 17383 auf sehr originelle Weise in den Bildschirm.

Andere »schöne Versionen« sind:

Art	Geschw.	Art	Geschw.
2	1	6	4
9	4	13	8
19	2	20	8
28	8	33	4
36	10	44	4
46	20	61	20
64	4	65	10
66	4	84	4
88	2	98	8
104	4	106	2
120	20	125	20

```
10 FOR I=659 TO 763:READ J:POKE I,J:NEXT J:P
RINT" {WHITE,CLR,DOWN}SYS 680,ART,BLOCK,
GESCHWINDIGKEIT {LIG.BLUE} <004>
20 NEW <158>
1000 DATA 164,6,162,32,202,208,253,136,208
,248,96,0,0,0,0,0,17,1,0,32,155 <153>
1001 DATA 183,134,2,32,155,183,134,5,32,15
5,183,134,6,165,2,10,9,1,133,97,169 <047>
1002 DATA 4,162,0,134,2,133,3,134,4,160,0,
177,4,145,2,165,2,24,101,97,133,2 <179>
1003 DATA 133,4,144,4,230,3,230,5,165,3,20
1,8,208,11,169,4,133,3,165,5,56,233 <102>
1004 DATA 4,133,5,32,147,2,165,2,208,211,1
65,3,201,4,208,205,96 <195>
```

Listing 1. »Screen-Dump«, mit dem Checksummer eingeben.

Funktionsweise

Wie funktioniert Screen-Dump, wo es doch nur 100 Byte lang ist? Am besten ist das an folgendem Beispiel zu erkennen: Wir nehmen eine Gruppe von zwölf Elementen. Wie greift man nun nacheinander auf alle zwölf Elemente zu, so daß keine Reihenfolge erkennbar wird? Ganz einfach:

Wir nehmen eine Zahl, die mit 12 keinen gemeinsamen Teiler hat, zum Beispiel 5. Jetzt beginnen wir bei 0 und zählen 5 hinzu. Diese Zahl nehmen wir heraus. Dann zählen wir wieder 5 hinzu. Ergebnis 10. Unser zweites Element. Dann wieder + 5 = 15. Element 15 ist nicht vorhanden, also ziehen wir die Gesamtanzahl der Elemente ab. 15—12=3 und so weiter. Es ergibt sich die Reihenfolge: 5 10 3 8 1 6 11 4 9 2 7 12. Augenscheinlich völlig wahllos. Es funktioniert immer, wenn folgende Voraussetzungen erfüllt sind:

1. Die Additionszahl (hier: 5) muß kleiner sein als die Gesamtzahl der Menge.
2. Die Additionszahl darf mit der Gesamtzahl keinen gemeinsamen Teiler haben.

Im Screen-Dump wird dieses System mit einer Menge von 1 024 Elementen angewandt. Die möglichen Additionszahlen sind somit alle ungeraden Zahlen zwischen 0 und 255.

(Michael Schmidt/tr)

Pseudo-Interrupt

Diese Befehlserweiterung erlaubt es, ein Basic-Programm zu jedem beliebigen Zeitpunkt per Tastendruck durch die F1-Taste unterbrechen zu lassen. Es kann dann in eine vorher definierte Basic-Routine gesprungen werden. Diese könnte

programm : pseudo-irq 9e01 9f40

```
9e01 : ff 89 00 20 73 00 c9 21 6a
9e09 : d0 03 4c 6d 9e a5 cb c9 f5
9e11 : 40 d0 06 20 79 00 4c e7 d8
9e19 : a7 c9 04 f0 06 20 79 00 0b
9e21 : 4c e7 a7 a5 cb c9 40 d0 ad
9e29 : fa ad 01 9e f0 06 20 79 c1
9e31 : 00 4c e7 a7 18 a9 03 20 61
9e39 : fb a3 a5 7a 48 a5 7b 48 ef
9e41 : a5 39 48 a5 3a 48 a9 75 c1
9e49 : 48 8d 01 9e ad 02 9e 85 dc
9e51 : 14 ad 03 9e 85 15 20 a3 99
9e59 : a8 4c ae a7 a9 04 8d 08 c9
9e61 : 03 a9 9e 8d 09 03 a9 75 cc
9e69 : 8d 01 9e 60 20 73 00 c9 5c
9e71 : 46 f0 07 c9 4a f0 41 4c f4
9e79 : f1 9e 20 73 00 c9 31 d0 e5
9e81 : 34 20 73 00 c9 4a d0 2d 2f
9e89 : 20 73 00 c9 55 d0 26 20 51
9e91 : 73 00 c9 4d d0 1f 20 73 8e
9e99 : 00 c9 50 d0 18 20 73 00 fc
9ea1 : 20 6b a9 a5 14 8d 02 9e 89
9ea9 : a5 15 8d 03 9e a9 00 8d ef
9eb1 : 01 9e 4c ae a7 4c 08 af 47
9eb9 : 20 73 00 c9 42 d0 f6 20 93
9ec1 : 73 00 c9 41 d0 ef 20 73 c3
9ec9 : 00 c9 43 d0 e8 20 73 00 f6
9ed1 : c9 4b d0 e1 68 c9 75 d0 fc
9ed9 : 59 68 85 3a 68 85 39 68 77
9ee1 : 85 7b 68 85 7a a9 00 8d ff
9ee9 : 01 9e 20 79 00 4c e7 a7 c2
9ef1 : c9 53 f0 07 c9 43 f0 1c 34
9ef9 : 4c 08 af 20 73 00 c9 46 24
9f01 : d0 b3 20 73 00 c9 31 d0 d6
9f09 : ac a9 75 8d 01 9e 20 73 05
9f11 : 00 4c ae a7 20 73 00 c9 09
9f19 : 46 d0 9a 20 73 00 c9 31 33
9f21 : d0 93 a9 00 8d 01 9e 20 c1
9f29 : 73 00 4c ae a7 4a 42 41 de
9f31 : 43 cb a9 2e 85 22 a9 9f d9
9f39 : 85 23 4c 47 a4 01 00 96 cb
```

Listing 2. »Pseudo-IRQ«. Bitte mit dem MSE eingeben.

zum Beispiel den noch freien Speicherplatz oder die Uhrzeit anzeigen.

Das Programm (Listing 2) bitte mit dem MSE eingeben. Es liegt dann im Speicherbereich von 40499 bis 40768. Geladen wird es absolut mit LOAD "PSEUDO-IRQ",8,1. Da das Programm im Bereich für die Basic-Variablen steht, muß es durch POKE 56,158:CLR vor Überschreiben geschützt werden. Nach dem Start durch »SYS 40541« stehen die neuen Befehle zur Verfügung:

!F1JUMP <Zeilennummer> legt fest, in welche Zeile im Falle einer Unterbrechung durch die F1-Taste gesprungen werden soll. Tritt der Befehl mehrmals auf, so gilt die zuletzt angegebene Zeilennummer.

!JBACK bewirkt die Fortsetzung des Basic-Programms ab der Stelle, an der unterbrochen wurde.

!SF1 verhindert Unterbrechungen. Dies kann zum Beispiel beim Aufbau einer Grafik oder bei Arbeiten mit der Diskettenstation wichtig sein.

!CF1 läßt gesperrte Unterbrechungen wieder zu.

(Guido Schuhmacher/tr)

List-Schutz für Basic-Programme

Dieser List-Schutz ist für Nichteingeweihte sehr verblüffend. Die Grundidee dazu stammt aus dem Bericht »Disketten-Manipulationen« aus 64'er, Ausgabe 6/85. Er wurde jedoch etwas ausgebaut, so daß hier beim Listen alle SteuerCodes aktiv werden. Dies wird dadurch erreicht, daß man in eine Speicherstelle vor den Codes die Zahl 141 schreibt. Um nun ein Programm zu schützen, läßt man es und gibt folgende zwei Zeilen ein:

```
1 POKE2067,73:GOTO 10          <134>
2 REM"A{CLR,3DOWN,YELLOW}IT IS NOT ALLOWED
  TO LIST THIS PROGRAM{BLUE}AA  <123>
```

Danach gibt man im Direktmodus POKE 2067,71 : POKE 2073,141 : POKE 2118,0 : POKE 2119,0 ein und speichert das Programm.

Listet man nun das Programm, so wird der Bildschirm gelöscht und der Text in der REM-Zeile ausgegeben. Durch das künstlich erzeugte Basic-Programm-Ende-Zeichen (drei Nullen) wird das Listen abgebrochen. Wird im Programm dann auch noch durch POKE 788,52 die RUN-STOP-, und durch POKE 792,193 die RESTORE-Taste ausgeschaltet, kann keiner mehr an das Programm. Aufheben läßt sich dieser List-Schutz nur mit einem Monitor und mit der Kenntnis der Funktionsweise des Schutzes.

(Thomas Uttendorfer/tr)

Sichere INPUT-Routine in Basic

Ich habe bisher immer eine leistungsfähige INPUT-Routine in Basic vermißt. Das ging anscheinend nicht nur mir so, denn in einigen abgedruckten Programmen (zum Beispiel Ligatab, Listing des Monats, 3/85) fehlt eine solche Routine, was zu chaotischen Ergebnissen führen kann. (Wenn man zum Beispiel aus Versehen an Stelle der Shift-Taste die Crsr-Down-Taste gedrückt hat, so nützt es dem Anwender wenig, wenn er mit der Crsr-Up-Taste wieder in das Eingabefeld zurückkehrt, weil jetzt der INPUT-Befehl die ganze Zeile als Eingabe ansieht.) Deshalb hier eine INPUT-Routine, die folgende Vorteile hat:

- a) Da die Routine (Listing 3) in Basic geschrieben ist, ist sie für jeden Programmierer leicht verständlich und abänderbar.
- b) Der Programmierer kann beim Aufruf der Routine gleich Ort, mini- und maximale Länge und die zulässigen Zeichen der Eingabe festlegen.
- c) Jedes Zeichen wird gleich bei der Eingabe auf seine »Richtigkeit« überprüft.
- d) Es wird nur die tatsächliche Eingabe als Eingabe übernommen (und nicht noch eventuell Teile der Maske).
- e) Es funktionieren noch sämtliche Sonderfunktionstasten (inst, del, home, clr und so weiter).
- f) Der Anwender kann das Eingabefeld nicht verlassen.
- g) Die Routine ist mit 1,5 KByte relativ kurz.

Benutzung der Routine:

- a) Setzen des Parameterstrings
 - b) Aufruf mit gosub 60000
 - c) Übergabe von in\$ an gewünschte Variable
- Der Parameterstring:

Er setzt sich zusammen aus:

Stelle 1,2	: Zeile der Eingabe
Stelle 3,4	: Spalte der Eingabe
Stelle 5,6	: minimale Länge der Eingabe
Stelle 7,8	: maximale Länge der Eingabe
Stelle 9	: Code (siehe weiter unten)

Es ist darauf zu achten, daß einstellige Angaben (zum Beispiel: 3. Zeile) mit 03 angegeben werden, da sonst keine klare Trennung der einzelnen Parameter vorgenommen werden kann.

Die Routine sieht Standardwerte für die minimale und maximale Länge und für den Code der Eingabe vor (Zeile 60200). Diese können natürlich frei nach eigenem Bedarf eingerichtet werden. Entspricht die geforderte Eingabe den Standardwerten, so braucht der Programmierer beim Setzen des Parameterstrings diese Werte nicht anzugeben, sondern nur noch Zeile und Spalte. Dies führt natürlich zu einer weiteren Vereinfachung (Beispiel: pa\$ = "0510", das heißt Eingabe in Zeile 5 ab Spalte 10 mit den Standardwerten).

Code (c)

Der Code dient dazu, die eingegebenen Zeichen sofort zu überprüfen. Selbstverständlich kann auch er frei programmiert werden. In Zeile 60490 wird entsprechend dem Code in die Unterprogramme verzweigt. Dort wird die gedrückte Taste überprüft. Entspricht sie nicht dem Code, wird f1 auf 1 gesetzt.

In der vorliegenden Routine sind Code 1 bis 4 schon programmiert, wobei

- 1 bedeutet, daß nur Zahlen
 - 2 bedeutet, daß nur Buchstaben
 - 3 bedeutet, daß alles
 - 4 bedeutet, daß nur Kleinbuchstaben
- zulässig sind.

Diese Codes oder auch neue Codes können spielend leicht geändert oder neu programmiert werden.

Funktionen der Sondertasten:

CRSR UP,CRSR DOWN	sind abgeschaltet
INST,DEL	funktionieren innerhalb des Eingabefensters wie gewohnt (Rest der Maske wird nicht berührt)
HOME	springt an Anfang des Eingabefensters
CLR	löscht Eingabefenster und springt an den Anfang desselben
CRSR LEFT,RIGHT	funktionieren wie gewohnt, nur Eingabefeld kann nicht verlassen werden.

(Karlheinz Boss/tr)

Programmbeschreibung

Variablenliste:

pa-	
\$	: Parameterstring; dient der Übergabe der Parameter
bl\$	: String, der nur aus Spaces besteht
in\$	: Eingabe
n\$	: gedrückte Taste
ze	: Zeile für die Eingabe
sp	: Spalte, ab der die Eingabe erfolgen soll
mi	: minimale Länge der Eingabe
ma	: maximale Länge der Eingabe
c	: Code, welchem die Eingabe entsprechen muß
ss	: aktuelle Cursorspalte
fl	: Flag, ob zuletzt gedrückte Taste Code entsprach

```

0 REM      INPUT (DEMO)                                <061>
1 REM INPUT-ROUTINE FÜR VC 64                            <067>
2 REM                                                    <064>
3 REM COPYRIGHT BY                                       <193>
4 REM KARLHEINZ BOSS                                     <157>
5 REM SYLVESTER-JORDAN-STR. 11                          <080>
6 REM 3550 MARBURG                                       <045>
7 REM 06421/13509                                       <003>
8 POKE 53281,09:POKE 53280,09:POKE 646,07             <090>
9 PRINT CHR$(14)                                         <246>
10 REM INPUT-ROUTINE                                    <090>
20 REM                                                    <082>
30 REM PA$ IST AUFGEBAUT WIE FOLGT:                     <170>
40 REM PA$ = ZESPIMAC WOBEI                             <124>
50 REM      ZE = ZEILE                                   <107>
60 REM      SP = SPALTE                                  <016>
70 REM      MI = MINDESTLÄNGE                           <086>
80 REM      MA = MAXIMALE LÄNGE                         <022>
90 REM      C = CODE WOBEI                               <161>
100 REM      1 = NUR ZAHLEN                              <193>
110 REM      2 = NUR BUCHSTABEN                          <163>
120 REM      3 = ALLES                                    <110>
125 REM      4 = NUR KLEINBUCHST.                       <138>
130 REM      5-6 = FREI PROGRAMMIERBAR                  <120>
132 REM      ( ZEILE 60490 )                             <189>
140 REM PA$ MUSS DIE ZEILE UND SPALTE                   <209>
150 REM BEIHALTEN, DER REST IST HIN-                    <023>
160 REM REICHEND. ES WERDEN DANN DIE                    <235>
170 REM STANDARDPARAMETER IN ZEILE                     <143>
180 REM 60200 ANGENOMMEN                                <219>
190 REM Z.B. PA$="0510"                                 <104>
200 :                                                    <176>
1000 PRINT "CLR";                                       <176>
1010 PRINT " (RVSON,2SPACE) INPUT-ROUTINE (3SP        <021>
      ACE) COPYRIGHT BY J. BOSS (3SPACE,RV OFF)
      "
1020 PRINT "UEBER GET WIRD EINE INPUT-ROUTINE SIMU-    <191>
      -"
1030 PRINT "LIERT, WOBEI SAEMTLICHE SONDER            <205>
      FUNKTIONS-";
1040 PRINT "TASTEN (INST,DEL,HOME,CLR,CR,C            <185>
      L) NOCH "
1050 PRINT "FUNKTIONIEREN, MAN ABER DAS EINGABEFELD    <072>
      "
1060 PRINT "NICHT VERLASSEN KANN. AUSSERDE            <077>
      M KANN MAN";
1070 PRINT "BEIM AUFRUF DER ROUTINE GLEICH            <104>
      ORT, MIN.";
1080 PRINT "UND MAX. LÄNGE UND DIE ZULÄS              <070>
      SIGEN ZEILE-";
1090 PRINT "CHEN DER EINGABE ANGEBEN."                <076>
1100 PRINT " (RVSON,GREY 2,SPACE) BEISP.: (32         <167>
      SPACE)";
1105 PRINT "{40SPACE}";                                <089>
1110 PRINT "DATUM (TIMMJJ) : (2SPACE,6RIGHT          <036>
      ,16SPACE)";

```

```

1120 PRINT "{40SPACE}";                                <104>
1130 PRINT "NAME : (2SPACE,20RIGHT,12SPACE)          <157>
      ";
1140 PRINT "{40SPACE}";                                <124>
1150 PRINT "WUERZEL (MIND 2 KL. BUCHST.) :           <200>
      (2SPACE,3RIGHT,5SPACE)";
1160 PRINT "{40SPACE}";                                <144>
1170 PRINT "COMPUTER : (2SPACE,15RIGHT,13SPACE)      <168>
      ACE)";
1180 PRINT "{40SPACE,RV OFF,YELLOW}";                 <060>
1900 REM HIER NUN DIE 4 AUFRUFE AUS                   <252>
1910 REM VORANGEGANGEN BEISPIEL                      <149>
1920 REM                                                <204>
2000 PA$="121806061":GOSUB 60000:A$=IN$              <085>
2010 PA$="140800202":GOSUB 60000:B$=IN$              <065>
2020 PA$="163202034":GOSUB 60000:C$=IN$              <222>
2030 PA$="1812":GOSUB 60000:D$=IN$                   <058>
2040 REM                                                <068>
2100 REM      DAS WAR'S                                <132>
2110 REM                                                <140>
3000 PRINT:PRINT "DIE EINGABEN WAREN : ";A           <136>
      $
3010 PRINT B$                                          <146>
3020 PRINT C$                                          <164>
3030 PRINT D$                                          <182>
3040 PRINT "(12SPACE,RVSON,SPACE) TASTE DRU          <003>
      ECKEN (SPACE,RV OFF)";
3050 GET NJ$: IF NJ$="" THEN 3050                     <188>
3995 PRINT "CLR";                                     <173>
4000 POKE 211,0:POKE 214,24:SYS 58640                <050>
4010 PRINT "LIST 1900-2140 (10SPACE)"                 <000>
4015 PRINT " (HOME) "                                <185>
4020 LIST 30-190                                      <105>
29999 END                                             <027>
30000 :                                                <002>
31000 :                                                <242>
32000 :                                                <226>
33000 :                                                <209>
34000 :                                                <193>
60000 BL$="{40SPACE}"                                <209>
60010 IN$=""                                           <007>
60099 :                                                <128>
60100 ZE=VAL (MID$(PA$,1,2))                          <243>
60110 SP=VAL (MID$(PA$,3,2))                          <121>
60120 IF LEN(PA$)=4 THEN 60200                       <191>
60130 MI=VAL (MID$(PA$,5,2))                          <125>
60140 MA=VAL (MID$(PA$,7,2))                          <135>
60150 CO=VAL (MID$(PA$,9,1))                          <153>
60160 GOTO 60300                                       <059>
60190 REM*****                                       <043>
60200 MI=0:MA=15:CO=3                                 <096>
60210 REM*****                                       <063>
60220 :                                                <251>
60300 GOSUB 60950                                       <217>
60310 POKE 204,0                                       <008>
60320 GET N$: IF N$="" THEN 60320                     <154>
60330 SS=PEEK (211)                                    <069>
60340 IF ASC(N$)=13 AND LEN(IN$)>=MI THEN              <135>
      POKE 204,1 : GOSUB 63890:RETURN
60350 IF ASC(N$)=147 THEN GOSUB 61000 : GO            <132>
      TO 60320
60360 IF ASC(N$)=19 THEN GOSUB 63890 : GOS            <108>
      UB 60950:GOTO 60320
60370 IF ASC(N$)=20 AND PEEK (211)>SP THEN              <165>
      GOSUB 61410 : GOTO 60320
60380 IF ASC(N$)=157 THEN GOSUB 61310:GOTO            <171>
      60320
60390 IF ASC(N$)=148 AND LEN(IN$)<MA THEN              <051>
      GOSUB 61505:GOTO 60320
60400 IF ASC(N$)=29 THEN GOSUB 61210:GOTO              <144>
      60320
60410 IF ASC(N$)=17 OR ASC(N$)=145 THEN                <222>
      60320
60420 IF ASC(N$)=148 OR ASC(N$)=20 OR ASC(N$)=13      <124>
      THEN GOTO 60320
60430 :                                                <207>
60485 FL=0                                              <202>

```

Fortsetzung auf Seite 95

Listing 3. Die bedienungssichere Input-Routine mit eingebautem Demo.

Fortsetzung von Seite 90

```

60490 ON CO GOSUB 63900,63910,63920,63930
        \,63940,63950 <021>
60495 IF FL=1 GOTO 60320 <168>
60500 : <021>
60505 IF LEN(IN$)=SS-SP THEN IN$=IN$+N$:GO
        TO 60515 <096>
60510 IN$=LEFT$(IN$,SS-SP)+N$+MID$(IN$,SS-
        SP+2,LEN(IN$)-SS+SP-1) <084>
60515 PRINT N$; <189>
60520 IF SS=SP+MA-1 THEN PRINT CHR$(157); <150>
60530 GOTO 60320 <047>
60901 : <168>
60902 : <169>
60903 : <170>
60950 POKE 211,SP:POKE 214,ZE:SYS 58640:RE
        TURN <244>
60999 : <012>
61000 GOSUB 60950 <155>
61010 PRINT LEFT$(BL$,MA); <145>
61020 GOSUB 60950 <175>
61030 IN$="" <011>
61040 RETURN <135>
61099 : <112>
61210 IF SS<SP+LEN(IN$)AND SS<SP+MA-1 THEN
        SS=SS+1 <019>
61220 GOSUB 63890 <138>
61230 POKE 211,SS:SYS 58640 <065>
61240 RETURN <081>
61250 : <009>
61310 IF SS>SP THEN SS=SS-1 <220>
61320 GOSUB 63890 <238>
61330 POKE 211,SS : SYS 58640 <165>
61340 RETURN <181>
61350 : <109>
61410 IN$=LEFT$(IN$,SS-SP-1)+MID$(IN$,SS-S
        P+1,LEN(IN$)-SS+SP) <118>
61420 GOSUB 63890 <082>
61425 IF LEN(IN$)<MA-1 THEN PRINT " "; <034>
61430 POKE 211,SS-1 : SYS 58640 <060>
61440 RETURN <027>
61490 : <251>
61505 IF SS=SP+LEN(IN$) THEN GOTO 61540 <188>
61510 IN$=LEFT$(IN$,SS-SP)+""+MID$(IN$,SS
        -SP+1,LEN(IN$)-SS+SP) <055>
61520 GOSUB 63890 <184>
61530 POKE 211,SS : SYS 58640 <111>
61540 RETURN <127>
61590 : <095>
63890 GOSUB 60950 <251>
63892 PRINT IN$;:IF LEN(IN$)<MA THEN PRINT
        " "; <126>
63898 RETURN <199>
63899 : <118>
63900 IF ASC(N$)<48 OR ASC(N$)>57 THEN FL=
        1 <105>
63909 RETURN <210>
63910 IF ASC(N$)=32 THEN RETURN <234>
63911 IF (ASC(N$)<65 OR ASC(N$)>90)AND (ASC(
        N$)<193 OR ASC(N$)>218) THEN FL=1 <230>
63919 RETURN <220>
63920 REM <225>
63929 RETURN <230>
63930 IF ASC(N$)=32 THEN RETURN <254>
63931 IF (ASC(N$)<65 OR ASC(N$)>90) THEN FL
        =1 <006>
63939 RETURN <240>
63940 REM <245>
63949 RETURN <250>
63950 REM <255>
63959 RETURN <004>

```

© 64'er

Listing Schluß

Synthetische Melodien

Das wahrscheinlich kürzeste und erstaunlichste Musik-Synthesizer-Sequencer-Programm, das je veröffentlicht wurde. Es handelt sich um ein 47 Byte langes Assemblerprogramm, das

im Listing 4, in Form eines DATA-Laders vorliegt. Die Melodie und die Klangfarbe kann man über die Speicherstellen 1022 und 1023 einstellen. Dabei kommt es allerdings nur auf die Differenz der beiden Byte an. Es empfiehlt sich also, die Speicherstelle 1022 mit 0 zu belegen und für 1023 alle Werte von 0 bis 128 auszuprobieren.

Melodische Tonfolgen erreicht man mit POKE 1023,7+8*n, wobei n eine ganze Zahl zwischen 0 und 16 sein darf. Effektvolle Klänge erreicht man zum Beispiel über folgende Werte (POKE 1023,...): 11, 28, 62, 96, 130.

Funktionsweise

Zuerst werden Kurven- und Wellenform festgelegt. Die beiden Speicherinhalte von 1022 und 1023 werden erhöht beziehungsweise erniedrigt. Dann werden sie miteinander durch die AND-Funktion verknüpft. Das Ergebnis kommt ins High-Byte des Tonhöhenregisters. Eine kleine Warteschleife macht die Komposition hörbar. (Wolfgang Horak/tr)

```

10 POKE 1022,0 <012>
20 INPUT"PARAMETER";A:POKE 1032,A <151>
30 FOR T=0 TO 46:READ Q:POKE T+832,Q:NEXT <078>
40 SYS 832 <220>
50 DATA 169,15,141,24,212,141,5,212,169,11
   2,141,6,212,169,17,141,4,212,238 <190>
60 DATA 255,3,206,254,3,162,0,16,0,200,208
   ,253,232,224,7,208,246,173,255 <222>
70 DATA 3,45,254,3,141,1,212,176,227 <042>

```

© 64'er

Listing 4 »Synthetische Melodien«. Bitte mit dem Checksummer eingeben.

Die Super-POKEs

POKE 792,226: POKE 793,252

Sobald die RESTORE-Taste gedrückt wird, wird ein Reset ausgelöst.

POKE 774,226: POKE 775,252

Sobald der LIST-Befehl eingetippt wird, wird ein Reset ausgelöst.

POKE 818,226: POKE 819,252

Sobald der SAVE-Befehl eingetippt wird, wird ein RESET ausgelöst. (tr)

Tips & Tricks — Mischmasch

Und hier noch ein paar Kleinigkeiten für die Tips & Tricks-Sammlung:

— Neustart eines Basic-Programms nach Drücken von RUN-STOP-RESTORE:

1 DATA 169, 49, 141, 20, 3, 76, 113, 168

2 FOR I=700 TO 707: READ A: POKE I,A: NEXT

3 POKE 770,188: POKE 771,2

Einzige Bedingung ist, daß Ihr Programm eine Zeile 0 enthält, da der Interpreter einen RUN 0-Befehl ausführt.

— Nachladen von Datasette mit Autostart:

POKE 56335,2

Dem Betriebssystem wird vorgegaukelt, daß SHIFT-RUN-STOP gedrückt wäre.

— Nützliche SYS-Adressen:

43121: entspricht dem Befehl RUN 0

42039: Ausgabe von Fehlermeldungen. Im X-Register des Prozessors muß die Fehlernummer stehen (Speicherstelle Nummer 781).

64763: RESET, ohne Initialisierung der Vektoren (IRQ etc.). (tr)