

Der EPROM-Automat

Was steckt hinter der Datenspeicherung auf EPROMs? Wir zeigen Ihnen, wie man seine eigenen Module aus Basic-Programmen herstellt.

Zur Programmierung von EPROMs sind drei Dinge notwendig: Ein Computer, ein EPROM-Programmiergerät, und das Wissen, in welcher Form Programme auf EPROMs gespeichert sein müssen. Dieser Artikel zeigt Ihnen, wie man eigene Programme so modifiziert, daß sie auf ein EPROM gespeichert und anschließend von dort auch ausgeführt werden können. In den folgenden Zeilen soll schrittweise erarbeitet werden, was dabei zu beachten ist. Sie sollen in die Lage versetzt werden, das Programm Auto-Eprom (Listing 1) zu verstehen und gegebenenfalls an Ihre eigenen Bedürfnisse anzupassen.

Was ist eigentlich ein EPROM? Nun, zunächst ist ein EPROM nichts anderes als ein Datenspeicher, der sich beschreiben, lesen und auch wieder löschen läßt. Wenn man einmal von dem etwas komplizierten Schreib- und Löschvorgang absieht, unterscheidet sich ein EPROM nur in einem für den C 64-Programmierer wichtigen Punkt von anderen Speichern wie beispielsweise den Disketten oder der Datenkassette. Es handelt sich hierbei um die Tatsache, daß ein auf EPROM gespeichertes Programm in der Regel die Anfangsadresse \$8000 (dezimal 32768) hat. Wie kann man aber den Computer überlisten und ein Basic-Programm, das ja normalerweise ab \$0801 (dezimal 2049) liegt, sofort nach dem Einschalten des Computers ohne weitere Befehle startet? Dafür gibt es prinzipiell zwei Lösungen. Die erste Möglichkeit besteht darin, den Basic-Start mit den Befehlen »POKE 44,128:Poke 32768,0:NEW« nach \$8000 zu verschieben und ein Basic-Programm in diesem Speicherbereich laufenzulassen.

Diese Methode hat jedoch den entscheidenden Nachteil, daß dann für Programm und Variablen nur noch magere 8 KByte zur Verfügung stehen. Die zweite wesentlich elegantere Möglichkeit besteht darin, das auf EPROM gebrannte Basic-Programm nach dem Einschalten des Computers in den normalen Speicherbereich für Basic-Programme ab \$0801 (dezimal 2049) zu verschieben. Es wird dann vom C 64 genauso behandelt, als ob es von einem externen Datenspeicher, wie dem Diskettenlaufwerk, geladen worden wäre. Ein automatisches Starten des Programms soll dabei natürlich nicht vergessen werden. Es versteht sich beinahe von selbst, daß diese Anforderungen nur in Maschinensprache zu realisieren sind. Wir wollen nun gemeinsam ein entsprechendes Programm erarbeiten. Dazu ist zunächst die Betriebssystem-Routine zu betrachten, die beim Einschalten des C 64 beziehungsweise nach einem Reset durchlaufen wird (Listing 2). Für uns besonders interessant ist das Unterprogramm, das in der Speicherzelle \$FCE7 beginnt. Dieses Unterprogramm prüft, ob ab Adresse \$8005 die Zeichenfolge »CBM80« steht. Sollte dies der Fall sein, wird in der nächsten Zeile die Adresse angesprungen, die in \$8000 und \$8001 gespeichert ist. Das bedeutet für das Verschiebeprogramm, dafür zu sorgen, daß ab \$8000 die Startadresse des Programms steht und ab \$8005 die Zeichenfolge »CBM80« zu finden ist. Damit haben wir schon einmal das erste Problem gelöst, da nun beim Einschalten das Verschiebeprogramm angesprungen wird. Das nächste Problem ist die Durchführung der Reset-Routine. Das

ist deshalb erforderlich, da sich der Computer nach dem Einschalten in einem völlig undefinierten Zustand befindet. Würde in diesem Zustand die Kontrolle an den Basic-Interpreter übergeben (was wir nach dem Verschieben des Basic-Programms ja vorhaben), wäre ein Systemabsturz die unweigerliche Folge. Das Verschiebeprogramm führt deshalb als nächstes dieselbe Initialisierungs-Routine durch, die das Betriebssystem des C 64 sonst automatisch ausführt. Anschließend gilt es dann, das Basic-Programm wieder an den Basic-Start nach \$0801 zu verschieben. Dazu stellt der Basic-Interpreter bereits eine bestens geeignete Routine zur Verfügung. Diese Routine erwartet die Anfangsadresse des zu verschiebenden Bereichs in den Speicherzellen \$5F/\$60 (dezimal 95/96) in der Form Low-/High-Byte, die Endadresse in den Speicherzellen \$5A/\$5B (dezimal 90/91) und die Endadresse des Bereichs in den verschoben werden soll in \$58/\$59 (dezimal 88/89).

Schließlich gibt es noch zwei Speicherstellen in der Zeropa-ge, die unsere Beachtung verdienen. Es sind die Zellen \$2D/\$2E (dezimal 45/46). Hier schaut der Interpreter später nach, wo das Basic-Programm aufhört, um nicht mit Variablenzuweisungen das Basic-Programm zu überschreiben. Beim Laden von der Diskette oder Kassette werden diese Speicherstellen automatisch auf die richtigen Werte gesetzt, in unserem Fall fällt diese Aufgabe dem Verschiebeprogramm zu. Nachdem wir die Blockverschieberoutine durchgeführt haben, könnte man bereits durch Sprung auf den Basic-Warmstart die Programm-Kontrolle an den Basic-Interpreter übergeben. Das Basic-Programm ist bereits list- und lauffähig. Es muß aber noch manuell mit RUN gestartet werden. Um auch dies dem Computer zu überlassen, brauchen wir vor dem Sprung auf den Basic-Warmstart lediglich noch den ASCII-Code für RUN und RETURN in den Tastaturpuffer schreiben und dem Interpreter mitteilen, daß im Tastaturpuffer ein zu verarbeitender Befehl steht. Der Tastaturpuffer beginnt bei \$0277 (dezimal 631) und die Anzahl der Zeichen, die im Tastaturpuffer stehen, werden dem Computer über die Speicherstelle \$00C6 (dezimal 198) mitgeteilt. Wenn der C 64 nun in den Direktmodus schaltet, verhält er sich genauso, als wenn Sie über die Tastatur »RUN« eingeben und die RETURN-Taste betätigt hätten. Damit ist das Verschiebeprogramm auch schon fertig. Bevor wir uns nun mit dem Problem beschäftigen, wie ein selbstgeschriebenes Basic-Programm zum Brennen auf EPROM in den Speicherbereich ab \$805C (hinter das Verschiebeprogramm) kopiert wird, schauen Sie sich bitte zunächst das dokumentierte Listing des Verschiebeprogramms an (Listing 3). Bei den Werten in den Adressen \$8045/\$8049/\$804D und \$8053 handelt es sich um Platzhalter. Diese Werte werden später entsprechend der Länge des Basic-Programms verändert. Jetzt braucht man das eigentliche Programm nur noch an das Ende des Verschiebeprogramms ab \$805C zu kopieren, die Platzhalter im Verschiebeprogramm durch die richtigen Werte ersetzen und den EPROMmer anwerfen.

Bei eingestecktem EPROM wird dann das Basic-Programm direkt nach dem Einschalten des Computers oder nach einem Reset automatisch gestartet. Nun wäre es aber sehr mühsam, jedes Programm, das gebrannt werden soll, mit einem Monitor nach \$805C zu verschieben. Genau diese Arbeit nimmt Ihnen das Programm »EPROM-Automat« (Listing 1) ab.

In diesem Programm finden Sie auch das eben entwickelte Maschinenprogramm in den DATA-Zeilen 2970 bis 3010 wieder. Die Bedienungsanleitung zu »EPROM-Automat« finden Sie in Bild 1. Betrachten wir uns die Zeilen 2650 bis 5010, die der eigentliche Hauptteil dieses Programms sind, etwas genauer. In diesen Zeilen wird das oben entwickelte Maschinenprogramm in den Speicherbereich ab \$8000 geschrieben und dahinter das zu bearbeitende Basic-Programm kopiert. Nachdem man den Namen des zu bearbeitenden Programms

eingetragen hat, wird in Zeile 2730 ein Lesekanal geöffnet. Danach werden die ersten beiden Byte dieser Programmdatei von der Diskette gelesen. Sie geben in der Form Low-/High-Byte die Startadresse des gespeicherten Programms an. Sollte die Startadresse nicht \$0801 sein, springt das Programm sofort ins Menü zurück. Natürlich ist es theoretisch möglich, auch reine Maschinenprogramme, die an einer beliebigen Adresse beginnen und mit SYS (Startadresse) gestartet werden, mit dem EPROM-Automat zu bearbeiten. Dazu sind folgende Änderungen notwendig: Das Verschiebeprogramm muß so modifiziert werden, daß es nicht mehr mit RUN gestartet wird, sondern die Startadresse direkt mit JMP (Startadresse) angesprungen wird. Die Endadresse darf nicht mehr mit dem festen Offset 30811 (Zeile 2920) berechnet werden, sondern muß in Abhängigkeit von der Anfangsadresse des Programms ermittelt werden. Wenden wir uns der Zeile 2810 zu. In dieser Zeile wird das Maschinenprogramm in den richtigen Speicherbereich verschoben. Im Anschluß daran wird Byte für Byte das Basic-Programm dahintergeschrieben. Dabei wird immer wieder in Abhängigkeit vom Zähler »K« geprüft, ob das Programmende nicht über \$A000 (dezimal 40960) hinausgeht. In diesem Fall käme man nämlich nicht mehr mit einem 8-KByte-EPROM aus. Wollen Sie jedoch auch 16-KByte-EPROMs brennen, müssen Sie außer dem Vergleichswert in Zeile 2850 auch noch das Verschiebeprogramm dahingehend modifizieren, daß es beim Kopieren des Basic-Programms den ROM-Bereich ab \$A000 abschaltet, um den darunterliegenden RAM-Bereich lesen zu können. Weiterhin wäre es dann erforderlich, die PEEK-Funktion des C 64 dahingehend zu erweitern, daß sie in der Lage ist, auch den vom ROM überlagerten RAM-Bereich ab \$A000 zu lesen.

Wollen Sie dagegen 4-KByte-EPROMs benutzen, müssen Sie lediglich den Wert in Zeile 2850 von 40960 in 36864 (\$9000) ändern oder selbst darauf achten, daß das Programm, das Sie bearbeiten wollen, nicht länger als 4 KByte ist. Wenn das Programmende erreicht ist, wird der Programmkanal in Zeile 2890 wieder geschlossen und die Werte für die Speicherstellen \$8046/\$804A und \$804E/\$8054 werden berechnet. Sie erinnern sich, dort stehen bisher nur Platzhalter, die noch durch die richtigen Werte für den zu verschiebenden Ursprungs- und Zielbereich ersetzt werden müssen. Die Berechnung der Endadresse des Ursprungsgebietes ist relativ einfach. Dazu muß man lediglich den Zähler »K«, der ja mit dem Programmende im Speicher übereinstimmt, in Low- und High-Byte zerlegen. Aber auch die Berechnung der Endadresse des Zielbereichs ist nicht viel schwieriger. Dazu braucht man von »K« lediglich die Differenz zwischen der jetzigen Anfangsadresse (\$805C) und der normalen Anfangsadresse (\$0801) subtrahieren. Nun steht Ihr Basic-Programm brennfertig ab \$8000 im Speicher. Nachdem Sie es auf EPROM gebrannt haben, bleibt noch ein kleines Problem. Beim Einschieben einer Modulplatine in den Expansion-Port des C 64 stehen für den Basic-Programmierer nur noch 30719 Byte zur Verfügung. Sollte Ihr Programm mit großen Variablenfeldern arbeiten, werden Sie die restlichen 8 KByte vielleicht auch noch benötigen. Sofern die Garantielzeit bei Ihrem Computer abgelaufen ist, öffnen Sie ihn einfach und bringen einen einfachen Ein-/Aus-Schalter an, der, wie in Bild 2 dargestellt, verdrahtet wird. Mit diesem Schalter können Sie dann bei Bedarf die Leitung EXROM, die bei eingesteckter Modulplatine normalerweise auf Masse liegt, freigeben und haben dann wieder den vollen Speicherbereich zur Verfügung. Noch ein Tip für diejenigen, die keinen EPROMmer besitzen. Auch sie können den EPROM-Automat sehr effektiv einsetzen, da die präparierten Programme nach einem Reset automatisch wieder starten. In Ihrem Basic-Programm muß lediglich die Zeile: POKE 55,0:POKE 56,128:CLR enthalten sein.

(Dirk Gazic/aw)

Bedienungsanleitung für das Programm EPROM-Automat:

1.) Folgende Anforderungen muß das zu bearbeitende Programm erfüllen:

- a) Es muß mindestens eine Basic-Zeile enthalten.
- b) Es darf nicht länger als 8 KByte (32 Blöcke auf der Diskette) sein.

2.) Menü

Nach dem Programmstart meldet sich der EPROM-Automat mit einem Menü. Sie können nun mit Hilfe der Cursorsteuertasten die einzelnen Optionen anwählen. Die jeweils angewählte Option wird dabei hervorgehoben. Durch Druck auf die RETURN-Taste wählen Sie dann den entsprechenden Programmteil aus. Sie haben im Hauptmenü zusätzlich die Möglichkeit, die eingestellten Bildschirmfarben zu verändern (F1 = Rahmen, F2 = Hintergrund, F3 = Schrift).

3.) Programme bearbeiten

Geben Sie bitte den Namen des Programms ein, das Sie bearbeiten wollen. Fehlermeldungen werden gegebenenfalls in der unteren Statuszeile ausgegeben.

4.) Speicherinhalt sichern

Dient zum Abspeichern von bearbeiteten Programmen auf Diskette. Sobald Sie diese Option angewählt haben, wird überprüft, ob Sie bereits ein Programm bearbeitet und dieses noch nicht gespeichert haben. Andernfalls werden Sie aufgefordert, einen Programmnamen anzugeben. Wenn Sie möchten, können Sie auch den vorgegebenen Namen übernehmen. Anschließend werden Sie gebeten, die Startadresse, ab der das Programm später geladen wird, einzugeben. Sollte die Steuersoftware für Ihren Eprommer bei \$8000 liegen, so können Sie zum Beispiel »6« statt »8« eingeben.

5.) Directory zeigen

Dieser Menüpunkt zeigt das Inhaltsverzeichnis der eingelegten Diskette. Dieser Vorgang kann jederzeit mit F1 abgebrochen werden. Die Funktionstaste F3 hält die Auflistung so lange an, bis eine beliebige Taste gedrückt wird.

6.) Programm beenden

Diese Option beendet nach positiver Beantwortung der Sicherheitsabfrage, die Arbeit mit EPROM-Automat. Das zuletzt bearbeitete Programm bleibt dabei im weiterhin geschützten Speicherbereich ab \$8000 erhalten und kann sofort auf ein EPROM gebrannt werden.

Bild 1. Bedienungsanleitung zum Programm »EPROM-Automat«

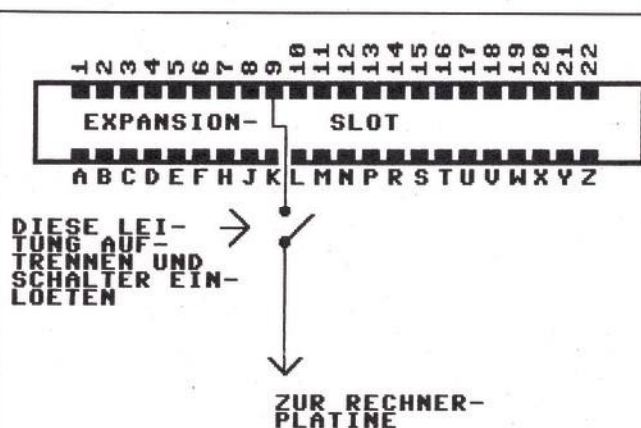


Bild 2. Mit einem kleinen Trick lassen sich auch Module abschalten, das Bild wurde mit HI-EDDI erstellt.


```

1 REM          DIRK GAZIC          <232>
2 REM          AMSTERDAMER STR. 41 <043>
3 REM          5000 KOELN 90       <185>
4 REM          TEL.:02203/32921    <164>
5 :                                <237>
6 :                                <238>
7 :                                <239>
8 :                                <240>
9 :                                <241>
1000 REM INITIALIZIERUNG ----- <000>
1010 :                               <224>
1020 POKE 55,0:POKE 56,128:CLR:DIM ZEX(4), <196>
      TEX(4)                               <058>
1030 HF=0:VF=0:SF=5:REM FARBVORGABE      <181>
1040 PRINT CHR$(14)CHR$(8)CHR$(147)      <204>
1050 RE$(0)="(RVOFF)":RE$(1)="(RVSON)"
1060 FOR I=1 TO 4:READ ZEX(I):READ TEX(I):
      NEXT                                <089>
1070 EI$="IHRE EINGABE BITTE >>>":REM HAEU
      FIGSTE STATUSZEILE                 <017>
1080 DATA 8,"PROGRAMM BEARBEITEN"        <033>
1090 DATA 10,"SPEICHERINHALT SICHERN"    <237>
1100 DATA 12,"DIRECTORY ZEIGEN"          <239>
1110 DATA 14,"PROGRAMM BEENDEN"          <076>
1120 GOTO 2560:REM HAUPTPROGRAMM         <168>
1130 :                                     <090>
1140 :                                     <100>
1150 :                                     <110>
1160 REM INPUT-ROUTINE -----           <005>
1170 :                                     <130>
1180 POKE 204,0:POKE 205,2:WAIT 207,1,1:PO
      KE 204,1                             <050>
1190 L=LEN(WW$):PRINT WW$;"(SPACE,LEFT)":
      IF L>0 THEN FOR ZZ=1 TO L:PRINT"(LEFT
      )":NEXT:L=0                          <015>
1200 POKE 204,0                          <083>
1210 POKE 198,0:WAIT 198,1:GET W$         <162>
1220 POKE 205,2:WAIT 207,1,1:POKE 204,1  <074>
1230 IF W$=CHR$(29)AND L<MX THEN 1300     <050>
1240 IF W$=CHR$(157)AND L>0 THEN L=L-1:PRI
      NT W$:GOTO 1200                     <191>
1250 IF W$=CHR$(13) THEN 1320             <004>
1260 IF W$=CHR$(20)AND L>0 THEN 1350     <241>
1270 IF ASC(W$)<32 OR ASC(W$)=34 THEN 1200 <152>
1280 IF W$="*"OR W$=";"OR W$="," THEN 1200 <209>
1290 IF ASC(W$)>127 AND ASC(W$)<160 THEN 1
      200                                <165>
1300 L=L+1:PRINT W$:IF L<MX THEN 1200    <111>
1310 L=MX-1:PRINT"(LEFT)":GOTO 1200      <225>
1320 IF L>0 THEN FOR ZZ=1 TO L:PRINT"(LEFT
      )":NEXT                          <203>
1330 WW$="":OPEN 3,3:FOR ZZ=1 TO MX:GET#3,
      W$:WW$=WW$+W$:NEXT:CLOSE 3        <143>
1340 RETURN                              <128>
1350 IF L=0 THEN 1200                    <228>
1360 GOSUB 1320                          <158>
1370 FOR ZZ=1 TO MX:PRINT"(LEFT)":NEXT    <191>
1380 IF L=1 THEN WW$=MID$(WW$,2):GOTO 1190 <208>
1390 WW$=LEFT$(WW$,L-1)+MID$(WW$,L+1):PRIN
      T WW$: " "                          <010>
1400 FOR ZZ=1 TO MX-L+1:PRINT"(LEFT)":NEX
      T                                  <219>
1410 L=L-1:GOTO 1200                     <043>
1420 :                                    <126>
1430 :                                    <136>
1440 :                                    <146>
1450 REM LEERSTELLEN ENTFERNEN ----- <203>
1460 :                                    <166>
1470 L1=LEN(WW$):IF L1=0 THEN RETURN      <141>
1480 L2=ASC(RIGHT$(WW$,1)):IF L2<>32 AND L
      2<>160 THEN RETURN                 <094>
1490 WW$=LEFT$(WW$,L1-1):GOTO 1470      <180>
1500 :                                    <206>
1510 :                                    <216>
1520 :                                    <226>
1530 REM UNTERE STATUSZEILE ----- <144>
1540 :                                    <248>
1550 I=INT((39-LEN(AN$))/2)              <198>
1560 AN$=LEFT$(" (36SPACE)",I)+AN$      <091>
1570 POKE 214,23:SYS 58640:PRINT:PRINT"(RV
      SON,39SPACE,RVOFF)":              <135>
1580 POKE 214,23:SYS 58640:PRINT:PRINT"(RV
      SON)":AN$"(RVOFF)":              <127>
1590 POKE 2023,160:POKE 56295,PEEK(646) <081>
1600 RETURN                              <134>

1610 :                                    <062>
1620 :                                    <072>
1630 :                                    <082>
1640 REM OBERE STATUSZEILE ----- <122>
1650 :                                    <102>
1660 PRINT"(HOME,RVSON,40SPACE,RVOFF)"  <039>
1670 I=INT((39-LEN(AN$))/2)              <062>
1680 AN$=LEFT$(" (36SPACE)",I)+AN$      <211>
1690 PRINT"(HOME,RVSON)":AN$;"(RVOFF)"  <247>
1700 RETURN                              <234>
1710 :                                    <162>
1720 :                                    <172>
1730 :                                    <182>
1740 REM FEHLERTON -----              <128>
1750 :                                    <202>
1760 SI=54272                             <021>
1770 FOR I=0 TO 23:POKE I+SI,0:NEXT      <050>
1780 POKE SI+1,7:POKE SI+5,60:POKE SI+24,1
      5:POKE SI+2,50:POKE SI+3,15:POKE SI+4
      ,65                                <180>
1790 RETURN                              <068>
1800 :                                    <254>
1810 :                                    <008>
1820 :                                    <018>
1830 REM DISK-FEHLER -----            <241>
1840 :                                    <038>
1850 INPUT#1,FF$,FE$,TT$,SS$             <177>
1860 IF FE$="OK" THEN RETURN              <217>
1870 AN$=FF$+"",FE$+"",TT$+"",SS$        <189>
1880 GOSUB 1530                           <234>
1890 IF FF$="74" THEN PRINT#1,"UJ":FOR I=1
      TO 3000:NEXT                      <160>
1900 GOSUB 1740:POKE 198,0:WAIT 198,1   <143>
1910 RETURN                              <190>
1920 :                                    <118>
1930 :                                    <128>
1940 :                                    <138>
1950 REM DIRECTORY -----              <094>
1960 :                                    <158>
1970 OPEN 1,8,15,"I":GOSUB 1830:CLOSE 1:IF
      FE$<>"OK" THEN RETURN              <039>
1980 PRINT"(CLR)":RE=0                   <208>
1990 OPEN 1,8,0,"$0"                     <106>
2000 GET#1,A$,B$                          <183>
2010 GET#1,A$,B$                          <193>
2020 GET#1,A$,B$                          <203>
2030 GET Q$:IF Q$="F1" THEN CLOSE 1:RETUR
      N                                  <137>
2040 IF Q$="F3" THEN WAIT 198,1:POKE 198,
      0                                  <172>
2050 PRINT"(RVOFF)":                     <196>
2060 C=0                                   <193>
2070 IF A$<>" " THEN C=ASC(A$)             <162>
2080 IF B$<>" " THEN C=C+ASC(B$)*256      <227>
2090 PRINT MID$(STR$(C),2):POKE 211,4    <041>
2100 IF RE=0 THEN PRINT"(RVSON)":        <118>
2110 GET#1,B$:IF ST<>0 THEN 2220          <150>
2120 IF B$<>CHR$(34) THEN 2110            <026>
2130 PRINT CHR$(34):                     <202>
2140 GET#1,B$:PRINT B$:IF B$<>CHR$(34)THE
      N 2140                             <100>
2150 PRINT"(2SPACE)":                   <118>
2160 GET#1,B$:IF B$=CHR$(32) THEN 2160   <050>
2170 POKE 211,24:C$=""                   <163>
2180 C$=C$+B$:GET#1,B$:IF B$<>" " THEN 2180 <014>
2190 IF LEFT$(C$,1)="*" THEN POKE 211,23 <085>
2200 PRINT C$                             <106>
2210 IF ST=0 THEN RE=1:GOTO 2010         <044>
2220 PRINT"BLOCKS FREE"                  <116>
2230 CLOSE 1                             <209>
2240 AN$="TRISTE" FUER RUECKKEHR INS MENUE
      ":GOSUB 1530                       <136>
2250 POKE 198,0:WAIT 198,1               <190>
2260 RETURN                              <030>
2270 :                                    <214>
2280 :                                    <224>
2290 :                                    <234>
2300 REM MENUESTEUERUNG -----        <219>
2310 :                                    <000>
2320 PRINT"(CLR)":POKE 646,SF            <067>

```

Listing 1. EPROM-Automat — ein komfortables Programm zur Modulherstellung. Bitte beachten Sie die Eingabehinweise auf Seite 54.


```

2330 AN$=NM$:GOSUB 1640 <192>
2340 AN$="HAEHLEN MIT CURSOR UND RETURN":G <001>
      OSUB 1530 <001>
2350 GOSUB 2510:RE=1:GOSUB 2500 <226>
2360 POKE 53280,HF:POKE 53281,VF <250>
2370 POKE 198,0:WAIT 198,1:GET Q$ <040>
2380 IF Q$=CHR$(13) THEN RETURN <019>
2390 IF Q$="UP" AND ZE>1 THEN RE=0:GOSUB <041>
      2500:ZE=ZE-1:RE=1:GOSUB 2500:GOTO 236 <041>
      0 <028>
2400 IF Q$="UP" THEN RE=0:GOSUB 2500:ZE=4 <028>
      :RE=1:GOSUB 2500:GOTO 2360 <028>
2410 IF Q$="DOWN" AND ZE<4 THEN RE=0:GOSU <022>
      B 2500:ZE=ZE+1:RE=1:GOSUB 2500:GOTO 2 <025>
      360 <025>
2420 IF Q$="DOWN" THEN RE=0:GOSUB 2500:ZE <237>
      =1:RE=1:GOSUB 2500:GOTO 2360 <237>
2430 IF Q$="F1" AND HF<255 THEN HF=HF+1:G <240>
      OTO 2360 <240>
2440 IF Q$="F1" THEN HF=0 <041>
2450 IF Q$="F3" AND VF<255 THEN VF=VF+1:G <186>
      OTO 2360 <246>
2460 IF Q$="F3" THEN VF=0 <246>
2470 IF Q$="F5" AND SF<15 THEN SF=SF+1:GO <118>
      TO 2300 <169>
2480 IF Q$="F5" THEN SF=0:GOTO 2300 <136>
2490 GOTO 2360 <136>
2500 POKE 211,8:POKE 214,ZE% (ZE):SYS 58640 <022>
      :PRINT RE$(RE) TE$(ZE):RETURN <097>
2510 FOR I=1 TO 4:POKE 211,8:POKE 214,ZE%( <036>
      I):SYS 58640:PRINT TE$(I):NEXT <220>
2520 RETURN <230>
2530 : <240>
2540 : <240>
2550 : <102>
2560 REM HAUPTPROGRAMM ----- <006>
2570 : <006>
2580 NM$=" AUTO.EPROM V1.2(6SPACE)(C) BY J <076>
      IRK GAZIC" <067>
2590 ZE=1:GOSUB 2300 <032>
2600 ON ZE GOSUB 2650,3050,1950,3270 <034>
2610 GOTO 2560 <056>
2620 : <066>
2630 : <076>
2640 : <065>
2650 REM PROGRAMM BEARBEITEN ----- <096>
2660 : <096>
2670 AN$="MODUS: PROGRAMM BEARBEITEN":GOSU <253>
      B 1640 <089>
2680 AN$=EI$:GOSUB 1530:WW$="" <089>
2690 PRINT "HOME,7DOWN,7RIGHT"&***** <158>
      *****S" <120>
2700 PRINT "RIGHT"&FILENAME: (17SPACE) <120>
2710 PRINT "RIGHT"&***** <064>
      *****X" <064>
2720 PRINT "2UP,17RIGHT":;MX=16:GOSUB 116 <066>
      0:GOSUB 1450:NA$=WW$ <195>
2730 OPEN 1,8,15:OPEN 2,8,0,NA$ <195>
2740 GOSUB 1830:IF FE$<>"OK" THEN CLOSE 1:C <170>
      LOSE 2:RETURN <131>
2750 GET#2,LB$:GET#2,HB$ <131>
2760 IF ASC(LB$+CHR$(0))=1 AND ASC(HB$+CHR <167>
      $(0))=8 THEN 2790 <167>
2770 AN$="PROGRAMMSTART IST NICHT $0801":C <140>
      LOSE 1:CLOSE 2:GOSUB 1740:GOSUB 1530 <140>
2780 POKE 198,0:WAIT 198,1:RETURN <127>
2790 RESTORE <044>
2800 AN$="BITTE WARTEN":GOSUB 1530 <017>
2810 FOR I=1 TO 4:READ A:READ A$:NEXT <089>
2820 FOR I=32768 TO 32859:READ A:POKE I,A: <177>
      NEXT <255>
2830 GET#2,A$:A$=A$+CHR$(0) <144>
2840 POKE I,ASC(A$) <046>
2850 I=I+1:IF I<40960 OR ST=64 THEN 2880 <046>
2860 AN$="PROGRAMM IST ZU LANG":CLOSE 1:CL <209>
      OSE 2:GOSUB 1740:GOSUB 1530 <219>
2870 POKE 198,0:WAIT 198,1:RETURN <165>
2880 IF ST<>64 THEN 2830 <174>
2890 CLOSE 1:CLOSE 2 <024>
2900 HB=INT(I/256):LB=I-HB*256 <024>
2910 POKE 32838,HB:POKE 32842,HB <080>
2920 I1=I-30811:I2=I <234>
2930 HB=INT(I1/256):LB=I1-HB*256 <245>
2940 POKE 32846,HB:POKE 32852,HB <175>
2950 RETURN <214>
2960 : <142>
2970 DATA 9,128,9,128,195,194,205,56,48,12 <227>
      0,142,22,208,32,163,253,32,80,253,32 <227>
2980 DATA 21,253,32,91,255,88,32,83,228,32 <031>
      ,191,227,32,61,128,162,251,154,169 <031>
2990 DATA 82,141,119,2,169,213,141,120,2,1 <238>
      69,13,141,121,2,169,3,141,198,0,76 <238>
3000 DATA 134,227,169,92,133,95,169,128,13 <033>
      3,96,169,110,133,90,169,128,133,91 <033>
3010 DATA 169,20,133,88,133,45,169,8,133,8 <024>
      9,133,46,76,191,163 <024>
3020 : <202>
3030 : <212>
3040 : <222>
3050 REM SPEICHERINHALT SICHERN ----- <092>
3060 : <242>
3070 IF I2=0 THEN GOSUB 1740:RETURN <061>
3080 AN$="MODUS: SPEICHERINHALT SICHERN":G <229>
      OSUB 1640 <074>
3090 AN$=EI$:GOSUB 1530 <074>
3100 PRINT "HOME,9DOWN,7RIGHT"&***** <231>
      *****S" <022>
3110 PRINT "RIGHT"&FILENAME: (17SPACE) <022>
3120 PRINT "RIGHT"&***** <011>
      *****X":WW$="MODUL." +NA$ <011>
3130 WW$=LEFT$(WW$,16):PRINT "2UP,17RIGHT" <028>
      ":MX=16:GOSUB 1160:GOSUB 1450 <075>
3140 OPEN 1,8,15:OPEN 2,8,1,WW$ <075>
3150 GOSUB 1830:IF FE$<>"OK" THEN CLOSE 1:C <072>
      LOSE 2:RETURN <072>
3160 PRINT "HOME,10DOWN,7RIGHT"&STARTADRES <077>
      SE: (3SPACE)* 4096(4SPACE)" <077>
3170 WW$=MID$(STR$(8),2):PRINT "UP,22RIGHT <051>
      ":;MX=1:GOSUB 1160 <051>
3180 AD=VAL(WW$):IF AD<1 OR AD>8 THEN GOSU <105>
      B 1740:GOTO 3160 <105>
3190 AD=AD*4096:HB=INT(AD/256):LB=AD-HB*25 <034>
      6 <028>
3200 PRINT#2,CHR$(LB)CHR$(HB); <028>
3210 AN$="BITTE WARTEN":GOSUB 1530 <175>
3220 FOR I=32768 TO I2:PRINT#2,CHR$(PEEK(I <146>
      )));NEXT <146>
3230 GOSUB 1830:CLOSE 1:CLOSE 2:I2=0:RETUR <185>
      N <168>
3240 : <168>
3250 : <178>
3260 : <188>
3270 REM PROGRAMM BEENDEN ----- <203>
3280 AN$="MODUS: PROGRAMM BEENDEN":GOSUB 1 <171>
      640 <171>
3290 AN$="MOLLEN SIE WIRKLICH BEENDEN [J/N <181>
      ]":GOSUB 1530:POKE 198,0:WAIT 198,1:G <166>
      ET Q$ <014>
3300 IF Q$="N" THEN RETURN <067>
3310 IF Q$<>"J" THEN 3290 <067>
3320 PRINT "CLR":IF I1=0 THEN 3340 <067>
3330 PRINT "BEARBEITETES PROGRAMM STEHT AB <095>
      $8000(4SPACE)EPROMFERTIG IM SPEICHER" <176>
3340 NEW <176>

```

© 64'er

Listing 1. EPROM-Automat — ein komfortables Programm zur Modulherstellung (Schluß).

```

FCE2 LDX &FF
FCE4 SEI
FCE5 TXS
FCE6 CLD
FCE7 JSR FD02 prüft auf ROM in $8000
FCEA BNE FCEF Sprung wenn Kein Autostart-Modul
FCEC JMP (8000) Sprung auf Modul-Start
FCEF STX D016 Ab hier folgt der eigentliche Reset
FCF2 JSR FDA3
FCF5 JSR FD50
FCF8 JSR FD15
FCFB JSR FF5B
FCFE CLI
FCFF JMP (A000) Sprung zum BASIC Kaltstart

```

Listing 2. Diese Betriebssystem-Routine prüft, ob ein Modul eingesteckt ist.

Fortsetzung auf Seite 99

Fortsetzung von Seite 94

```

8000 .byte 09      Startadresse des Verschiebeprogramms (Lowbyte)
8001 .byte 80      Startadresse des Verschiebeprogramms (Highbyte)
8002 .byte 09      Neuer NMI-Vektor (Lowbyte)
8003 .byte 80      Neuer NMI-Vektor (Highbyte)
8004 .byte C3      'C'
8005 .byte C2      'B'
8006 .byte CD      'M'
8007 .byte 38      '8'
8008 .byte 30      '0'
8009 SEI
800A STX D016
800D JSR FDA3      Interrupt vorbereiten
8010 JSR FD50      Arbeitsspeicher initialisieren
8013 JSR FD15      Hardware und I/O Vektoren setzen
8016 JSR FF5B      Video-Reset
8019 CLI
801A JSR E453      Basic-Vektoren laden
801D JSR E3BF      RAM für Basic initialisieren
8020 JSR 803D      Basic-Programm verschieben
8023 LDX #FB
8025 TXS

8026 LDA #52        Code für »I«
8028 STA 0277
802B LDA #D5        Code für »U«
802D STA 0278

```

```

8030 LDA #0D        Code für RETURN
8032 STA 0279
8035 LDA #03        3 Zeichen im Tastaturpuffer
8037 STA 00C6
803A JMP E386       Sprung zum Basic-Warmstart

803D LDA #5C        Anfangsadresse des zu verschiebenden Bereiches (Lowbyte)
803F STA 005F
8041 LDA #80        Anfangsadresse des zu verschiebenden Bereiches (Highbyte)
8043 STA 0060
8045 LDA #6E        Endadresse des zu verschiebenden Bereiches (Lowbyte)
8047 STA 005A
8049 LDA #80        Endadresse des zu verschiebenden Bereiches (Highbyte)
804B STA 005B
804D LDA #14        Endadresse des Zielbereiches (Lowbyte)
804F STA 0058
8051 STA 002D        Basic-Ende setzen (Lowbyte)
8053 LDA #08        Endadresse des Zielbereiches (Highbyte)
8055 STA 0059
8057 STA 002E        Basic-Ende setzen (Highbyte)
8059 JMP A3BF       Sprung zur Blockverschieberoutine

```

Listing 3. Dieses Programm muß vor jedem Basic-Programm im EPROM stehen

Tipp-Utility

In »Tipp-Utility« sind vier wichtige Funktionen wie Tastaturpieps und OLD-Funktion integriert, die das Programmieren in Basic erleichtern.

Das Programm besteht aus einer hilfreichen LIST-Funktion, einer akustischen Tasten- und Fehlerüberwachung, sowie aus einer OLD-Routine. Nach dem Abtippen des Programms (Listing) sollte man es unbedingt speichern, da eventuelle Fehler den C 64 »aufhängen« können. Gestartet wird das Programm mit »SYS49152«. Um das Programm nach einem Reset oder RUN/STOP-RESTORE wieder zu aktivieren, genügt die Eingabe von »SYS 49152«.

Die vier Funktionen

Pieps: Die erste Funktion, die das Programm beinhaltet, ist die der Tastenüberwachung, die bei jedem Tastendruck einen hellen Ton erzeugt. Diese Funktion läßt sich mit F3 ausschalten und mit F1 wieder aktivieren.

Fehlerton: Sollten sie irgendwann einmal einen dunklen Ton hören, so haben sie irgendeinen Fehler gemacht, denn der C 64 gibt nun auch mit jeder Fehlermeldung einen Warnton aus.

Listunterstützung: Vorteilhaft ist auch die LIST-Routine. Nach der Eingabe von LIST kann mit allen SHIFT-Tasten, also auch mit der SHIFT-LOCK-Taste zeilenweise gelistet werden. Der C 64 listet nur, wenn Sie die SHIFT-Taste gedrückt halten. Drücken sie die RUN/STOP-Taste, so verläßt er die Routine.

OLD: Die letzte Funktion ist die mit »SYS 49444« zu starten- de OLD-Routine. Sie bewirkt, daß nach einem NEW oder RESET das Basic-Programm zurückgeholt wird.

Informationen zum Programm: Mit POKE 1,55 läßt sich das Maschinenprogramm abschalten. Durch POKE 1,53 oder SYS 49152 wird es wieder aktiviert. Grundsätzlich können alle Parameter der Töne geändert werden. Hier nur die wichtigsten:

Das Low-Byte ist für alle Töne (gemeint sind Tipp- und Error-Ton) immer gleich. Es kann der entsprechende Wert nach dem Start in die Adresse 54272 gePOKEt werden.

Für das High-Byte gibt es allerdings zwei verschiedene Werte. Das High-Byte für den Tippton wird in die Adressen 49336 und 54273 gePOKEt. Für den Error-Ton gilt die Adresse

49336. Der Wert für die Lautstärke gilt wieder für alle Töne und gehört wie immer in die Adresse 54296.

(Schentke/og)

PROGRAMM : TIPP-UTILITY C000 C161

```

C000 : A2 32 A9 A0 A0 00 84 22 9A
C008 : 85 23 B1 22 91 22 C8 D0 BE
C010 : F9 E6 23 CA D0 F4 A2 32 42
C018 : A9 E0 A0 00 84 22 85 23 0F
C020 : B1 22 91 22 C8 D0 F9 E6 54
C028 : 23 CA D0 F4 A9 35 8D 01 FF
C030 : 00 A9 20 8D E7 E5 A9 7A 08
C038 : 8D E8 E5 A9 C0 8D E9 E5 D4
C040 : A9 B7 8D 4F A4 A9 C0 8D C8
C048 : 50 A4 A9 0F 8D 18 D4 A9 77
C050 : 19 8D 05 D4 A9 F0 8D 06 70
C058 : D4 A9 50 8D 01 D4 A9 45 AE
C060 : 8D 00 D4 A9 E1 8D A5 A6 C6
C068 : A9 C0 8D A6 A6 A9 08 8D 9C
C070 : 15 A7 A9 C1 8D 16 A7 60 E4
C078 : 00 00 AD CB 00 C9 05 D0 61
C080 : 08 A9 B1 8D 93 C0 4C 97 1A
C088 : C0 C9 04 D0 05 A9 97 8D 5F
C090 : 93 C0 4C 97 C0 00 00 A9 E9
C098 : 21 8D 04 D4 A0 19 A2 FF 79
C0A0 : CA F0 03 4C A0 C0 88 F0 41
C0A8 : 03 4C 9E C0 A9 00 8D 04 6A
C0B0 : D4 20 B4 E5 60 00 00 A9 D7
C0B8 : 09 8D 01 D4 A9 21 8D 04 45
C0C0 : D4 A0 59 A2 FF CA F0 03 AF
C0C8 : 4C C5 C0 88 F0 03 4C C3 18
C0D0 : C0 A9 00 8D 04 D4 A9 50 45
C0D8 : 8D 01 D4 20 D7 AA 60 00 73
C0E0 : 00 A9 F1 8D 86 E8 A9 C0 BB
C0E8 : 8D 87 E8 20 68 A9 60 00 FC
C0F0 : 00 AD 8D 02 C9 01 D0 04 5A
C0F8 : 20 EA E8 60 AD CB 00 C9 A0
C100 : 3F D0 EE 4C 08 C1 00 00 7B
C108 : A9 EA 8D 86 E8 A9 E8 8D F5
C110 : 87 E8 4C 86 E3 60 00 00 30
C118 : 00 00 00 00 00 00 00 19
C120 : 00 00 00 00 A5 2B A4 2C BF
C128 : 85 22 84 23 A0 03 C8 B1 EC
C130 : 22 D0 FB C8 98 18 65 22 F6
C138 : A0 00 91 2B A5 23 69 00 BB
C140 : C8 91 2B 88 A2 03 E6 22 CF
C148 : D0 02 E6 23 B1 22 D0 F4 91
C150 : CA D0 F3 A5 22 69 02 85 B4
C158 : 2D A5 23 69 00 85 2E 60 F4
C160 : 00 FF 00 FF 00 FF 00 FF 60

```

Listing. »Tipp-Utility«. Beachten Sie bitte die Eingabehinweise auf Seite 54