

Nichts ist ewig

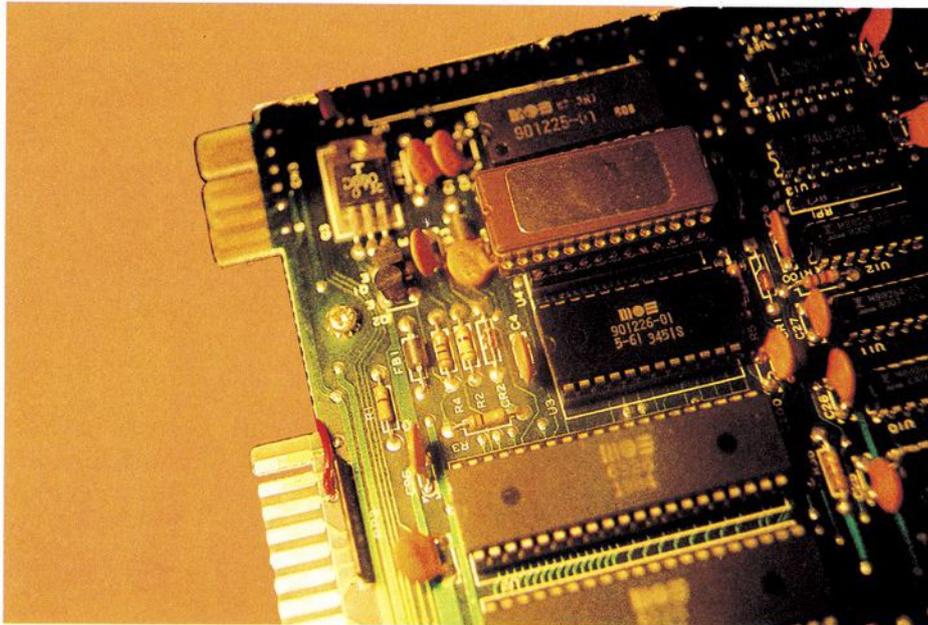


Bild 4. Das EPROM 2754 zweifach »gesockelt«

Zugegeben, der Commodore 64 hat einige Nachteile. Aber warum sollte man sich damit abfinden? Alles kann mit dem ROM-Change-Programm verändert werden.

Jedem Computer, auch dem Commodore 64 wird ab Werk eine bestimmte Ausstattung an Software mit auf den Weg gegeben. Damit ist nicht die Demo-Diskette gemeint, sondern die fest im Computer eingebaute, auf PROMs gespeicherte Firmware. Sie sorgt dafür, daß der Commodore überhaupt auf Eingaben reagiert oder einen Basic-Befehl ausführt.

Der gesamte C 64 (und jeder andere Computer) ist eigentlich nichts anderes als eine Anzahl miteinander verdrahteter Baugruppen, die allein zu nichts fähig sind. Leben eingehaucht wird dem ganzen erst durch ein sehr wichtiges Programm, das sogenannte Betriebssystem. Dieses Programm initialisiert und verwaltet die gesamte Hardware. Beim Commodore 64 ist es genau 8 KByte lang und liegt im Bereich von

\$E000 bis \$FFFF. Der zweite wichtige Festwertspeicher ist der Basic-Interpreter. Er ist ein Übersetzungsprogramm, das eine Anweisung in ein maschinengerechtes Signal umwandelt. Auch der Interpreter benötigt 8 KByte und liegt im Bereich von \$A000 bis \$C000. Jetzt fehlt nur noch der Charakter-Set von 4 KByte, der ebenfalls in einem eigenen ROM untergebracht ist.

Diese drei Programme sind für das äußere Erscheinungsbild und die Funktionalität des Commodore 64 verantwortlich. Hier eröffnet sich ein extrem interessanter Bereich der Programmierung. Dazu bieten sich zwei Wege an: Der erste Weg beruht auf der glücklichen Tatsache, daß es beim Commodore möglich ist, den RAM-Bereich unter einem ROM zu nutzen. In der Praxis sieht das folgendermaßen aus: Die Speicherinhalte des ROMs werden zum Bearbeiten in den darunter liegenden RAM-Bereich kopiert. Ob nun das RAM oder das ROM aktiv ist, entscheidet das 6510 Datenrichtungsregister (Speicherstelle 1). Vom Basic aus ist dieses Register allerdings nur dann zu verändern, wenn sowohl der Basic-Interpreter, als auch das Betriebssystem in das RAM kopiert wurden. Das geschieht

entweder mit einer POKE-Schleife oder mit dem ROM-Change Programm. (Betriebssystem kopieren: FOR A=57344 TO 65535: POKE A,PEEK (A) : NEXT) (Basic kopieren: FOR A=40960 TO 49152: POKE A, PEEK (A) : NEXT). Der Normalwert dieses Registers ist 55 (probieren Sie es aus). Soll das RAM (für Basic) selektiert aktiv sein, muß hier der Wert 54 eingeschrieben werden. Für Basic und Betriebssystem zusammen beträgt der Wert 53. Alle Veränderungen des Basic und Betriebssystems sind dann aktiv. Bekannte Programme wie Quickcopy und viele Basic-Erweiterungen funktionieren nach diesem Prinzip.

Der zweite und wesentlich reizvollere Weg die Firmware zu beeinflussen ist, das Betriebssystem dauerhaft zu verändern. Dazu ist aber ein Eingriff im Computer notwendig, denn die oben beschriebenen Bausteine müssen gegen andere ausgetauscht werden. Wer also noch Garantie auf seinen Commodore hat, sollte besonders vorsichtig sein. Im ersten Teil dieses Artikels wollen wir, zum Einüben sozusagen, die Funktionstasten des Commodore 64 mit bestimmten, oft gebrauchten Werten belegen. Im zweiten Teil wird das

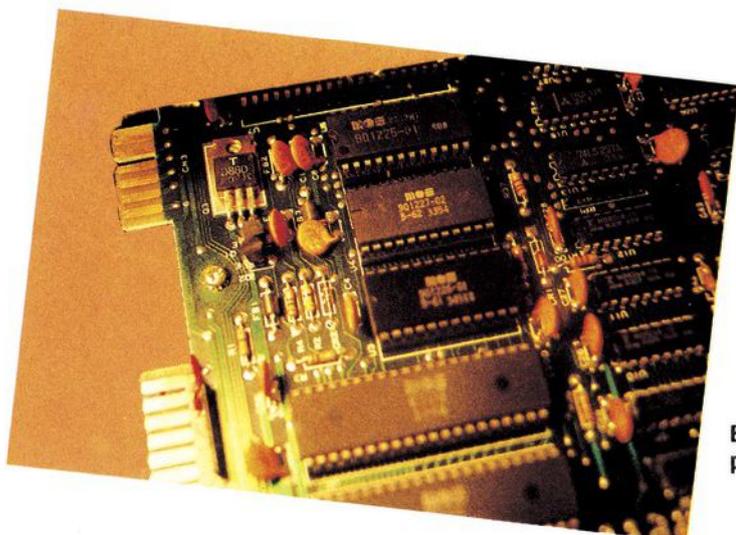


Bild 1. Die Steckplätze U3 bis U5

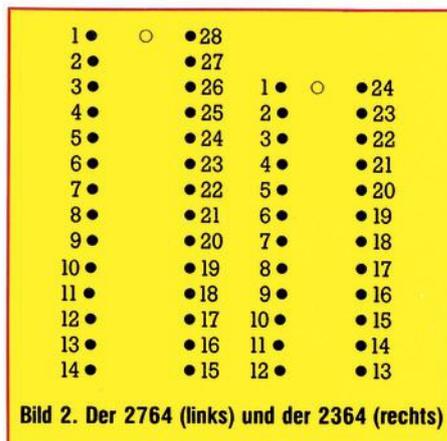


Bild 2. Der 2764 (links) und der 2364 (rechts)

Bild 3. So wird der 2764 mit den Pins des 2364-Sockels verbunden

2764		2364
1,28,27,26	—	24
2	—	21
3	—	1
4	—	2
5	—	3
6	—	4
7	—	5
8	—	6
9	—	7
10	—	8
11	—	9
12	—	10
13	—	11
14	—	12,20
15	—	13
16	—	14
17	—	15
18	—	16
19	—	17
20,14	—	nicht angeschl.
21	—	19
22	—	20
23	—	18
24	—	22
25	—	23
1,28,27,26	—	24

triebssystems erklärt werden. Nach dem Öffnen des Computers finden wir auf der linken hinteren Seite drei kleine Bausteine, hinter denen auf der Platine die Bezeichnungen U3 bis U5 stehen. U3 ist der Basic-Interpreter, U4 das Betriebssystem, U5 das Charakter-ROM (Bild 1). Heute interessieren wir uns aber nur für den U4-Steckplatz. Wer Glück hat, findet dort einen Stecksockel. Wer Pech hat, muß seinen Händler oder einen LötKolbenfachmann bitten, ihm hier einen Stecksockel einzulöten. Anstelle des dort befindlichen ROMs kann aber, und das ist die wesentlichste Veränderung, auch ein EPROM stecken. Am besten eignen sich hierzu die 2564-Typen, denn sie sind pinkompatibel zu den Commodore-ICs. Leider sind sie sehr schwer zu beschaffen. Im Normalfall wird aber wahrscheinlich ein 2764-EPROM Verwendung finden. Der Nachteil dieses EPROMs besteht in einer anderen Belegung der Anschlußpins. Hier hilft allerdings ein einfacher Adaptersockel. Dazu braucht man einen 24- und einen 28-Pin-Stecksockel. Diese beiden Sockel werden miteinander verdrahtet (Bild 2 und 3). Bild 2 zeigt die beiden Sockel mit den Beinen nach unten stehend. Der kleine Sockel steckt später im U4-Steckplatz, der größere Sockel trägt das geänderte Betriebssystem und steckt auf dem kleineren Sockel. Vor dem Einbau ist es aber ratsam, alle Kontakte auf richtigen Anschluß und Leitfähigkeit zu überprüfen. Schon ein einziger falscher Anschluß führt zum »Absturz« des gesamten Systems. Bild 4 zeigt, wie der neue Sockel mit der EPROM-Kerbe zur Gehäuse-Rückseite auf der Platine steht.

Nun aber zur Praxis, dem ROM-Change-Programm. Es ermöglicht das gefahrlose Ändern und Auspro-

bieren aller Umprogrammierungen. Dazu wird nach dem Start der selektierte ROM-Bereich (Betriebssystem oder Basic) ab Adresse \$6000 ins RAM kopiert. Das geschieht mit einem kurzen Maschinenprogramm, das im Kassettenpuffer steht. Das Programm ist für den Betrieb mit einem Diskettenlaufwerk gedacht, läuft aber auch im Kassettenbetrieb, wenn ein eigener Monitor zum Laden und Abspeichern Verwendung findet. Nach dem Kopieren erscheint das Hauptmenü, von dem aus alle wichtigen Funktionen erreichbar sind. Die erste dient dem Einlesen von fest im Programm eingebauten DATA-Zeilen. In unserem Beispiel sind ab Zeilennummer 8000 die DATAs für die Funktionstasten einprogrammiert. Hier können natürlich auch eigene Werte stehen. Die erste Zahl gibt dabei die reelle Adresse an, ab der die DATAs geschrieben werden sollen (zum Beispiel 57612 = \$E10C). Der Computer errechnet dann die entsprechende Stelle im RAM. Die zweite Zahl gibt an, wieviel Bytes übertragen werden sollen. Die dritte Zahl ist die Prüfsumme. Danach folgen die Programm-DATAs. Bei einem Prüfsummenfehler zeigt der Computer die falsche Prüfsumme an. Eigene Daten werden einfach an die vorhandenen DATAs angehängt. Die ersten drei Bytes müssen natürlich auch die obige Bedeutung haben. Als letzte Zeile muß stehen: DATA 0, da es sonst zu einem OUT OF DATA ERROR kommt. Der zweite Menüpunkt liest Maschinenprogramme direkt an die vorgesehene Stelle. Damit kann man beispielsweise ein Programmfile, das von einem Assembler erzeugt wurde, direkt einlesen. Wichtig ist, daß die Startadresse des Programms im ROM-Adreßbereich beziehungsweise im Bereich \$6000 liegt. Der mit Punkt 3 wählbare Minimonitor hilft beim schnellen Ändern einzelner Bytes

Hypra-Load aus der Ausgabe 10/84 im Betriebssystem verankert. Der Nachteil dieser Änderungen soll nicht verschwiegen werden. Da alle neuen Funktionen natürlich Speicherplatz benötigen, müssen andere Teile des Betriebssystems entfallen. Wir haben uns entschlossen, die Kassettenroutinen ab \$F800 herauszunehmen und zu überschreiben. Das Laden von Kassette wird dadurch unmöglich, es sei denn, das alte Betriebssystem wird parallelgeschaltet.

Bevor wir nun auf das Hilfsprogramm für diese Veränderungen, das ROM-Change-Programm, eingehen, sollen die Hardware-Voraussetzungen für die Änderung des Be-

im hexadezimalen Zahlensystem. Die Startadresse entspricht dabei der Adresse im ROM. Der Minimonitor wird durch Eingabe einer Zahl größer \$FF oder des Buchstabens X verlassen. Will man keine Speicherstelle ändern, genügt RETURN für die Übernahme des alten Wertes. Für größere Änderungen reicht dieser Minimonitor natürlich nicht mehr aus. Dazu wird mit Punkt 4 ein eigener Monitor aktiviert. Dieser muß allerdings vor dem Start des ROM-Change-Programms bereits geladen sein. Der Monitor darf im Bereich von \$8000 bis \$9FFF oder von \$C000 bis \$CFFF stehen. Zum Starten genügt das Eingeben der Startadresse des Monitors (hexadezimal!). Mit den Menüpunkten 5 und 6 wird der Speicherbereich von \$6000 bis \$7FFF geladen beziehungsweise abgespeichert. Das Laden eines kompletten Betriebssystems dauert allerdings mehrere Minuten. Einer der wichtigsten Menüpunkte ist aber der siebte. Er startet das gerade veränderte oder geladene neue Betriebssystem (es wird natürlich zuerst in seinen richtigen Speicherbereich verschoben). Vor dem Ausprobieren dieser Funktion muß in jedem Fall der Punkt 5 ausgewählt

werden, wenn die Änderungen nicht verlorengehen sollen. Es kann vorkommen, daß das Betriebssystem beim Starten »abstürzt«. Dann genügt ein RESET und die erneute Aktivierung des geänderten ROMs durch POKE 1,53.

Sind alle Tests im RAM positiv verlaufen, können wir uns an die Speicherung des neuen Betriebssystems auf EPROMs heranwagen. Das neue Betriebssystem wird von Diskette in den RAM-Bereich ab \$6000 absolut, das heißt mit LOAD "Ihr Betriebssystem",8,1 geladen. Ab dieser Adresse beginnt auch die Programmierung des EPROMs. Sie endet bei \$8000 (8 KByte). Das neue EPROM wird anschließend in den beschriebenen Stecksockel eingesetzt und in Steckplatz U4 befestigt. Fertig ist das Betriebssystem.

Was noch zu klären bleibt, ist die im ROM-Change-Programm bereits eingebaute Funktionstastenbelegung. Die Abfrage der Tastatur geschieht im Betriebssystem über den Interrupt. Wird eine Taste gedrückt, hält der Prozessor das laufende Programm an und springt in die Interrupt-Unterroutine bei \$EA31. Dort wird festgestellt, welche Taste gedrückt wurde. Der ASCII-Code der

Taste wird im Tastaturpuffer gespeichert. Das Funktionstastenprogramm greift an dieser Stelle ein, indem es den Wert der gedrückten Tasten mit den ASCII-Codes für die Funktionstasten (133-140) vergleicht. Stimmt der Wert nicht überein, so wird das Programm normal weitergeführt. Ansonsten vergleicht das Programm den Tastenwert mit einer Tabelle, in der die Belegungen für die Funktionstasten stehen. Hat das Programm die zur Funktionstaste gehörige Belegung gefunden, wird diese in den Tastaturpuffer geschrieben. Damit auch Programme, die auf einer Abfrage der Funktionstaste aufbauen, funktionieren, wurde eine Autoabschalt-Unterroutine mitprogrammiert. Kommt es dennoch einmal zu Schwierigkeiten, werden die Funktionstasten mit POKE 2,1 abgeschaltet und mit POKE 2,0 wieder aktiviert.

Abschließend sei nochmals darauf hingewiesen, daß alle Arbeiten an der Hardware des C 64 mit einem nicht geringen Zerstörungsrisiko verbunden sind. Wer also im Umgang mit Lötkolben und EPROMs nicht geübt ist, sollte sich an einen Fachmann wenden.

(Arnd Wängler/Ernst Schöberl/aa)

Beim Abtippen des Programms sind die Werte in Klammern am Ende einer Zeile nicht mit einzugeben. Sie sind erst für eine spätere Ausgabe von Bedeutung. Unterstrichene Buchstaben sind mit der Shift-Taste, überstrichene mit der Commodore-Taste einzugeben. Bei Ausdrücken in eckigen Klammern ist die jeweilige Taste zu drücken.

```

100 REM SPEICHERPLATZ FUER ROM VOR BASIC SCHUET
ZEN <003>
110 POKE 55,0:POKE 56,96:CLR <120>
120 POKE 53272,23:POKE 53280,2 <212>
130 GOSUB 1130 <212>
140 IF PEEK(53247)<>33 THEN 160 <055>
150 IF PEEK(53246)=0 THEN GOSUB 3600:GOTO 500
<111>
155 GOSUB 3700:GOTO 500 <074>
160 PRINT " [DOWN2] ":PRINT " [SPACE] BRUECKEN [SPACE]
SIE ":PRINT <059>
170 PRINT:PRINT TAB(7); "1 [SPACE2]UM [SPACE]
ERNAL [SPACE] ($E000-$FFFF) " <219>
180 PRINT:PRINT TAB(7); "2 [SPACE2]UM [SPACE] BASIC
[SPACE] ($8000-$BFFF) " <131>
190 PRINT:PRINT:PRINT TAB(11); "IN [SPACE] ROM
[SPACE] AB [SPACE] $6000 [SPACE] ZU [SPACE]
KOPIEREN. " <072>
200 GET A$:IF A$="" THEN 200 <039>
210 IF A$="1" THEN POKE 32767,255:GOSUB 3600
:GOTO 240 <225>
220 IF A$<"2" THEN 200 <223>
230 GOSUB 3700 <061>
235 REM AN=ANFANGSADRESSE ROM
EN=ENDADRESSE <231>
240 GOSUB 1600 <068>
300 POKE K+8,INT(AN/256) <252>
310 POKE K+29,INT(EN/256) <061>
340 SYS K:REM SCHIEBEROUTINE <032>
500 POKE 53280,5:GOSUB 1130:PRINT:POKE 53247,0
<128>
505 PRINT:PRINT:PRINT " [SPACE6] ERWEITERUNG [SPACE]
DES [SPACE] ROMS [SPACE] DURCH ":PRINT <052>
506 PRINT TAB(8); "1: [SPACE] DATA [SPACE] ZEILEN
[SPACE] EINLESEN " <182>
510 PRINT TAB(8); "2: [SPACE] PROGRAMM [SPACE] VON
[SPACE] DISK " <010>
520 PRINT TAB(8); "3: [SPACE] HEXMONITOR " <032>
530 PRINT TAB(8); "4: [SPACE] EIGENEN [SPACE]
MONITOR " <193>
540 PRINT TAB(8); "5: [SPACE] ABSPEICHERN [SPACE]
DES [SPACE] NEUEN [SPACE] ROMS " <101>
545 PRINT TAB(8); "6: [SPACE] LADEN [SPACE] VON
[SPACE] GEAENDERTEM [SPACE] ROM " <021>
546 PRINT TAB(8); "7: [SPACE] STARTEN [SPACE] DES
[SPACE] NEUEN [SPACE] ROMS " <106>
547 PRINT TAB(8); "8: [SPACE] PROGRAMM [SPACE]
BEENDEN " <137>
550 PRINT:PRINT:PRINT TAB(10); "BITTE [SPACE]
WAEHLLEN [SPACE] SIE! " <017>
560 GET A$:IF A$="" THEN 560 <154>
570 I=VAL(A$):IF I<1 OR I>8 THEN 560 <200>
575 POKE 53247,33:POKE 53280,2 <160>
580 ON I GOTO 1000,2000,3000,4000,5000,6000,
2500,3500 <001>
999 REM DATA ZEILEN EINLESEN <173>
1000 PRINT " [CLEAR] ":PRINT:PRINT TAB(10); "LESEN
[SPACE] DER [SPACE] DATA-ZEILEN ":PRINT:PRINT
<051>
1010 GOSUB 1600 <073>
1020 READ A:IF A=0 THEN 500 <235>
1030 READ B:REM ANZAHL DER BYTES <188>
1032 READ P1:REM PRUEFSUMME <230>
1033 P2=0 <113>
1035 D=A:GOSUB 1300:PRINT "$";A$; "-";:D=A+B-1
:GOSUB 1300:PRINT "$";A$; <109>
1040 FOR I=A-OF TO A-OF-1+B <254>
1050 READ D:POKE I,D <115>
1051 P2=P2+D:IF P2>65535 THEN P2=P2-65536 <212>
1053 NEXT I <236>
1055 IF P2<>P1 THEN 1070 <132>
1057 PRINT " [SPACE3] OK " <156>
1060 GOTO 1020 <116>
1070 PRINT "PRUEFSUMME [SPACE] FALSCH: [SPACE]";P2
<192>
1080 GET A$:IF A$="" THEN 1080 <210>
1090 GOTO 1020 <146>

```

Listing »ROM-Change«

```

1100 STOP <224>
1130 PRINT "[CLEAR]":PRINT TAB(8);"PROGRAMM
[SPACE]ZUM[SPACE]ENDERN[SPACE]DES":PRINT
<065>
1140 PRINT TAB(12)"BETRIEBSSYSTEMS":PRINT:PRINT
<050>
1150 PRINT "[SPACE]GESCHRIEBEN[SPACE]VON[SPACE]
ERNST[SPACE]SCHOEBERL[SPACE](1984)" <187>
1160 RETURN <026>
1199 REM HEX IN DEZ UMWANDLUNG <163>
1200 D=0:FOR J1=0 TO LEN(A$)-1 <235>
1210 B=ASC(RIGHT$(A$,1)):A$=LEFT$(A$,LEN(A$)-1)
<245>
1220 B=B-48:IF B>9 THEN B=B-7 <197>
1230 IF B<0 OR B>16 THEN 1280 <254>
1240 D=D+B*16↑J1 <082>
1250 NEXT J1 <227>
1260 RETURN <126>
1280 D=-1:RETURN <159>
1299 REM DEZ IN HEX <006>
1300 J2=INT(LOG(D)/LOG(16)):A$="" <084>
1310 FOR J1=J2 TO 0 STEP-1 <166>
1320 K1=INT(D/16↑J1):D=D-K1*16↑J1 <185>
1330 IF K1>9 THEN K1=K1+7 <090>
1340 K1=K1+48:A$=A$+CHR$(K1) <245>
1350 NEXT J1 <072>
1360 RETURN <227>
1600 RESTORE:K=828:FOR I=K TO K+32:READ A
:POKE I,A:NEXT I <056>
1610 DATA 120,160,0,132,251,132,253,169,224,133,
252,169,96,133,254,177,251,145 <241>
1620 DATA 253,200,208,249,230,252,230,254,165,
252,201,0,208,239,96 <167>
1630 RETURN <242>
1999 REM PROGRAMM VON DISK IN ROM EINFUEGEN
<003>
2000 GOSUB 1130:PRINT <252>
2010 PRINT TAB(2);"[EINFUEGEN[SPACE]EINES[SPACE]
PROGRAMMS[SPACE]VON[SPACE]DISK" <087>
2020 PRINT:PRINT:INPUT"EILENAME:":A$ <200>
2030 OPEN 1,8,0,A$ <022>
2033 GOSUB 6100:I=ASC(A$):GOSUB 6100
:I=I+256*ASC(A$):IF ST>0 THEN 6200 <052>
2035 PRINT:PRINT"FOUND[SPACE]":A$ <198>
2036 IF I<AN THEN 2110 <124>
2040 GET#1,A$:A$=A$+CHR$(0) <045>
2050 IF ST>0 THEN 2100 <135>
2060 POKE I-OF,ASC(A$):I=I+1:GOTO 2040 <190>
2100 CLOSE 1:GOTO 500 <101>
2109 REM PROGRAMM NICHT IN ROMBEREICH
<038>
2110 PRINT"ROM[SPACE]VON[SPACE]#":D=AN
:GOSUB 1300:PRINT A$:"-":D=AN+8191
:GOSUB 1300:PRINT A$ <075>
2120 PRINT"STARTADRESSE[SPACE]DES[SPACE]
PROGRAMMS[SPACE]BEI[SPACE]#":D=I:GOSUB 1300
:PRINT A$ <193>
2130 INPUT"NEUE[SPACE]STARTADRESSE[SPACE](MUSS
[SPACE]IM[SPACE]ROM-BEREICH[SPACE]LIEGEN
:":A$ <134>
2140 GOSUB 1200:IF D=-1 OR D<AN THEN 2100 <235>
2150 I=D:GOTO 2040 <054>
2499 REM ROM STARTEN <106>
2500 GOSUB 1600 <033>
2510 POKE K+8,160:POKE K+29,0 <235>
2515 POKE K+12,160 <142>
2520 SYS K <202>
2525 K=828 <133>
2530 POKE K+8,96 <074>
2540 POKE K+29,128 <179>
2550 POKE K+12,INT(AN/256) <249>
2560 SYS K <243>
2565 PRINT "[CLEAR]":PRINT">NEUES[SPACE]ROM
[SPACE]AKTIVIERT":PRINT <200>
2570 POKE 1,53:INPUT"RESET[SPACE](J/N)":A$
:IF A$="J"THEN SYS 64738 <233>
2580 GOTO 500 <060>
2999 REM MONITOR FUER EINZELNE SPEICHERSTELLEN
<111>
3000 PRINT "[CLEAR]":PRINT"***[SPACE]MONITOR
[SPACE]***":PRINT:PRINT <148>
3010 INPUT"STARTADRESSE:":A$:GOSUB 1200 <015>
3020 I=D-OF:IF I=-1 THEN 3010 <093>
3030 D=I+OF:GOSUB 1300:PRINT A$;"[SPACE]":
:D=PEEK(I):GOSUB 1300:PRINT A$;"[SPACE4]":
<229>
3040 INPUT":":A$ <142>
3050 IF A$=CHR$(13)THEN 3070 <132>
3060 GOSUB 1200:IF D=-1 OR D>255 THEN 3090 <154>
3065 POKE I,D <084>
3070 I=I+1:GOTO 3030 <177>
3090 PRINT"ENDE[SPACE](J/N)" <175>
3095 GET A$:IF A$="N"THEN 3010 <002>
3096 IF A$<>"J"THEN 3095 <126>
3100 GOTO 500 <070>
3500 POKE 53247,0:END <107>
3599 REM WERTE FUER KERNAL RAM <002>
3600 AN=14*4096:EN=0:OF=8*4096:POKE 53246,0
:RETURN <035>
3699 REM WERTE FUER BASIC ROM <025>
3700 AN=10*4096:EN=12*4096:OF=4*4096
:POKE 53246,1:RETURN <050>
3999 REM EIGENER MONITOR <100>
4000 PRINT "[CLEAR]":PRINT:PRINT"[SPACE7]EIGENER
[SPACE]MONITOR" <048>
4010 PRINT:PRINT:INPUT"STARTADRESSE[SPACE](HEX)
:":A$ <205>
4020 IF LEN(A$)>4 THEN 4000 <023>
4030 GOSUB 1200:IF D=-1 THEN 4000 <031>
4040 SYS D <185>
4050 GOTO 500 <255>
4999 REM ABSPEICHERN DES ROMS <106>
5000 GOSUB 1130:PRINT <192>
5010 PRINT TAB(7);"ABSPEICHERN[SPACE]DES[SPACE]
NEUEN[SPACE]ROMS" <124>
5020 PRINT:PRINT:INPUT"EILENAME:":A$ <140>
5030 OPEN 1,8,1,A$ <219>
5035 PRINT:PRINT"SAVING[SPACE]":A$ <214>
5040 PRINT#1,CHR$(0);CHR$(96); <253>
5050 FOR I=6*4096 TO 32767 <171>
5060 PRINT#1,CHR$(PEEK(I));:NEXT I <128>
5070 PRINT#1,CHR$(AN-INT(AN/256)*256); <139>
5080 PRINT#1,CHR$(AN/256) <209>
5090 CLOSE 1:GOTO 500 <030>
5999 REM GEAEENDERTES ROM LADEN <142>
6000 GOSUB 1130:PRINT <172>
6010 PRINT TAB(10);"LADEN[SPACE]EINES[SPACE]
NEUEN[SPACE]ROMS" <106>
6020 PRINT:PRINT:INPUT"EILENAME:":A$ <120>
6030 OPEN 1,8,0,A$ <198>
6040 GOSUB 6100:GOSUB 6100 <145>
6042 IF ST>0 THEN 6200 <051>
6043 PRINT:PRINT"LOADING[SPACE]":A$ <000>
6050 FOR I=6*4096 TO 32767 <151>
6060 GET#1,A$:A$=A$+CHR$(0) <241>
6070 POKE I,ASC(A$):NEXT I <090>
6080 GOSUB 6100:AN=ASC(A$):GOSUB 6100
:AN=AN+256*ASC(A$) <041>
6090 OF=AN-6*4096:EN=AN+8192:CLOSE 1 <174>
6095 IF EN>65535 THEN EN=0 <217>
6096 IF AN<14*4096 THEN POKE 53246,1:GOTO 500
<143>
6097 POKE 53246,0:GOTO 500 <055>
6100 GET#1,A$:A$=A$+CHR$(0):RETURN <225>
6200 CLOSE 1:PRINT <244>
6210 OPEN 1,8,15 <032>
6220 IF ST=64 THEN 6250 <038>
6230 GET#1,A$:PRINT A$;:GOTO 6220 <244>
6250 CLOSE 1 <083>
6260 GET A$:IF A$=""THEN 6260 <039>
6270 GOTO 500 <180>
7999 REM AB HIER DATAS <252>
8000 DATA 64226,126,14014 <209>
8001 DATA 232,134,198,201,133,144,4,201,141,144,
3,76,66,235,157,119,2,72,152 <150>
8002 DATA 72,160,0,196,2,208,13,185,41,251,221,
119,2,240,11,200,192,176,208 <087>
8003 DATA 243,104,168,104,76,66,235,200,185,41,
251,201,133,144,4,201,141,144 <140>
8004 DATA 238,236,137,2,176,233,157,119,2,232,
134,198,76,15,251,133,76,207 <080>
8005 DATA 34,36,34,44,56,13,137,76,79,65,68,134,
76,73,83,84,13,138,83,65,86 <160>
8006 DATA 69,135,82,85,78,13,139,83,89,83,136,
63,70,82,69,40,48,41,13,140,83 <202>
8007 DATA 89,83,54,52,55,51,56,13,133,136 <241>
8010 DATA 60223,3,552 <016>
8011 DATA 76,226,250 <227>
10000 DATA 0 <234>

```

Listing »ROM-Change« (Schluß)