

Tips & Tricks für Anfänger und Fortge- schrittene

Hier wieder einige interessante Hinweise zum Programmieren des C 64.

Kopieren von Bildschirmzeilen

Der Kern des Einzelers (Listing 1) besteht aus dem Aufruf einer Betriebssystemroutine, die das Kopieren von Bildschirmzeilen ermöglicht. Mit dieser Routine wird ein Scrollen des Bildschirms erreicht, indem die erste Zeile in die letzte Bildschirmzeile kopiert und anschließend durch den Print-Befehl ein Zeilenvorschub ausgelöst wird. Somit erscheint die am oberen Bildschirmrand verdrängte Zeile wieder unten. Um den Einzelers komfortabler zu gestalten, wird das zum Starten nötige »RUN« zuerst gelöscht, um ein Mitscrollen zu verhindern; mit Druck auf eine beliebige Taste wird der Scroll-Vorgang angehalten und nach Loslassen wieder fortgesetzt. Nach dem Eingeben des Einzelers sollte der Bildschirm gelöscht werden und anschließend der zu scrollende Text oder die Grafik erstellt werden. Es ist lediglich darauf zu achten, daß die 39. Bildschirmzeile nicht benutzt wird. Das Starten erfolgt am Anfang der letzten Bildschirmzeile mit »RUN«, gefolgt von einem Doppelpunkt (um eventuell folgende Zeichen auszuschalten). Die Routine eignet sich besonders gut für Spiele, die mit bewegtem Hintergrund arbeiten, da sie relativ schnell ist.

Dank dieser Routine ist es zum Beispiel möglich gewesen, ein grafisches Froggerspiel (Listing 2) in nur 23 Basic-Zeilen zu erstellen. (Stefan Moll/tr)

```
1 PRINT" {UP,4SPACE}":FOR T=0 TO 1:POKE 172
,PEEK(60656):POKE 780,PEEK(216):SYS 5984
8:PRINT:WAIT 203.64:T=0:NEXT <157>
```

**Listing 1. Der Einzeiler zum Zeilencopy.
Bitte alle Befehle abkürzen
und die Zwischenräume weglassen.**

[illegible]

**Listing 2. »Frogger« in 23 Basic-Zeilen unter
Zuhilfenahme der Routine aus Listing 1.**

```

4 FOR T=0 TO 126:POKE 832+T,PEEK(1024+T):N
EXT:POKE V+41,13:POKE V+42,6:POKE V+27,1
2 <061>
5 POKE 650,128:POKE 649,2:POKE V+32,0:POKE
V+33,5:POKE V+23,8:POKE V+29,8:S=300:G=
0 <168>
6 POKE V+16,8:POKE V+6,30:POKE V+7,120:Y=1
00:X=40:POKE V+4,X:POKE V+5,Y:POKE V+21,
12 <082>
7 POKE V+30,PEEK(V+30)AND 251:POKE V+31,PE
EK(V+31)AND 243 <049>
8 X$="{RVSON,SPACE,DOWN,LEFT,SPACE,DOWN,LE
FT,SPACE,4DOWN,LEFT,SPACE,DOWN,LEFT,SPAC
E,DOWN,LEFT,SPACE,4DOWN,LEFT,SPACE,DOWN,
LEFT,SPACE,DOWN,LEFT,SPACE,4DOWN,LEFT,SP
ACE,DOWN,LEFT,SPACE,DOWN,LEFT,SPACE}" <206>
9 PRINT "{CLR,RED}";TAB(10);X$:PRINT "{HOME,
BLACK,2DOWN}";TAB(17);X$:PRINT "{HOME,WHI
TE}";TAB(24);X$:PRINT "{HOME,GREY 2,2DOWN
}";TAB(31);X$ <002>
10 POKE 56322,224:J=PEEK(56321):IF(J AND 1
)=0 THEN Y=Y-15:GOTO 17 <159>
11 IF PEEK(V+31)AND 4=4 THEN PRINT "{BLACK,
CLR}TOT";G;"FROESCHE GERETTET":WAIT 145
,16,16:GOTO 5 <216>
12 IF PEEK(V+30)=12 THEN PRINT "{CLR,BLACK}
GERETTET !":S=S-50:G=G+1:WAIT 145,16,16
:GOTO 6 <043>
13 IF(J AND 2)=0 THEN Y=Y+15:GOTO 17 <018>
14 IF(J AND 4)=0 THEN X=X-15:GOTO 17 <172>
15 IF(J AND 8)=0 THEN X=X+15:GOTO 17 <110>
16 POKE 172,PEEK(60656):POKE 780,PEEK(216)
:SYS 59848:PRINT:FOR L=1 TO S:NEXT:GOTO
10 <023>
17 IF Y>230 THEN Y=230 <059>
18 IF Y< 40 THEN Y= 40 <185>
19 IF X>63 AND(PEEK(V+16)AND 4)=4 THEN X=6
3 <009>
20 IF X<25 AND(PEEK(V+16)AND 4)<>4 THEN X=
25 <145>
21 IF X<0 AND(PEEK(V+16)AND 4)=4 THEN POKE
V+16,8:X=255+X <210>
22 IF X>255 THEN POKE V+16,PEEK(V+16)OR 4:
X=X-255 <073>
23 POKE V+4,X:POKE V+5,Y:GOTO 10 <155>

```

Listing 2. »Frogger« (Schluß)

Scrolling nach unten in Basic

Die Idee ist folgende: Wenn man auf dem Bildschirm eine Zeile schreibt, die länger als 40 Zeichen ist, so wird alles, was unterhalb dieser Zeile steht, nach unten wegscrollt. Der folgende Einzeiler bewirkt, daß nachfolgende PRINT-Anweisungen auf dieselbe Weise den Bildschirm nach unten wegschieben:

1 PRINT CHR\$(19)CHR\$(17)CHR\$(157)CHR\$(148):POKE
218,152 (Thomas Dreier/tr)

Disketten von verstellten Laufwerken lesen

```
10 OPEN1,8,15,"M-W"+CHR$(105)+CHR$(0)+  
CHR$(X):CLOSE1
```

Dieses »Programm« schreibt in die Speicherzelle \$69 des Laufwerks-Speichers den Wert X. Diese Speicherzelle enthält die Konstante, die bestimmt, wieviele Leseversuche gemacht werden, bevor eine Fehlermeldung ausgegeben wird. Wird diese Konstante (der Standardwert ist 5) größer gewählt, so kann man auch Disketten lesen, bei denen das Laufwerk vorher Probleme hatte. Die Zugriffszeit erhöht sich natürlich entsprechend der Konstanten und der Anzahl der auftretenden Leseschwierigkeiten. Man hat aber somit die Möglichkeit, Disketten von schlecht justierten Laufwerken zu lesen.

(Alex Haderer/tr)

Reset ohne Datenverlust

Wirkung: Folgender Einzeiler wirkt (fast) wie ein Reset; das Programm wird jedoch ohne Datenverlust fortgeführt:
1 POKE 648,4:SYS 64789:SYS 58451:SYS 58784

Anwendungsbeispiele:

Sie wollen in einem Basic-Programm sämtliche Erweiterungen (Toolkits etc.) abschalten sowie den regulären Zeichensatz darstellen.

Ein Programm mit hochauflösender Grafik, verändertem Zeichensatz und/oder Schutz-POKEs (gegen Listen, Run/Stop, Save etc.) ist beendet (Fehlermeldung, Abbruch). Wenn Sie Eingaben machen, erscheinen diese nicht auf dem Bildschirm, nur die Farbe ändert sich. In einem solchen Fall geben Sie den Einzeiler ohne Zeilennummer blind ein.

(Florian Müller/tr)

Bildschirmcode in ASCII-Code umwandeln

1 X=X+(X>127)*128:W=X-(X>-1ANDX<32ORX>95)
*64-(X>63ANDX<96)*32

In der Variablen X wird der Bildschirmcode (PEEK(...)) abgelegt. In der Variablen W befindet sich dann der zugehörige ASCII-Code. Es muß allerdings berücksichtigt werden, daß nach Einsetzen eines Codes für inverse Zeichen der Inhalt der Variablen W nachher dem Code für das »normale Zeichen« entspricht. Will man diese Routine aber auch dazu benutzen, inverse Zeichen zu drucken, so muß man vorher überprüfen, ob der Bildschirminhalt größer als 127 ist, denn dann handelt es sich ja um ein inverses Zeichen. Wäre das der Fall, könnte man zum Beispiel durch Setzen des Reversflags (Adresse 199) auf 1 einen inversen Ausdruck bewirken.

(Knut Smoczyk/tr)

Langsame Bildschirmausgaben

Dieses kleine Maschinenprogramm (Listing 3) verlangsamt sämtliche Bildschirmausgaben des C 64 um ein Vielfaches. Es läßt sich ohne Probleme in eigene Programme einbinden.

Abschalten läßt sich die kleine Erweiterung mit RUN/STOP-RESTORE oder mit »POKE 806,202:POKE 807,241«. Mit POKE 53232,Zahl kann man die Ausgabegeschwindigkeit regulieren (1=normal).

Nach dem Starten verschiebt sich das Maschinenprogramm, das zuerst in den Bildschirmspeicher geschrieben wird, selbst nach 53228. Dann werden sämtliche Ausgabevektoren auf diese Adresse gerichtet, die nichts weiter als zwei ineinandergeschachtelte Verzögerungsschleifen beinhaltet.

(Dominik Irion/tr)

```
10 PRINT "{CLR}":FOR I=0 TO 43:READ X:POKE
1024+I,X:NEXT:SYS 1024 <028>
11 DATA 160,0,185,24,4,153,236,207,200,192
,20,208,245,160,236,140,38,3,160 <222>
12 DATA 207,140,39,3,96 <179>
13 DATA 72,152,72,160,100,169,255,233,1,20
8,252,136,208,247,104,168,104,76 <027>
14 DATA 202,241 <176>
```

Listing 3. Die verlangsamte Bildschirmausgabe.

Direkter Programmmodus

Jedem Floppy-Besitzer ist das schon oft passiert: die rote Leuchtdiode blinkt und zeigt einen Fehler an. Was tun?

- Ein Programm zum Auslesen des Fehlerkanals eingeben oder einladen ist nicht möglich, da das im Computer befindliche Programm nicht gelöscht oder verändert werden soll.
- Ein Programm im Direktmodus eingeben? Der Computer

reagiert auf diesen Versuch mit einem »ILLEGAL DIRECT ERROR«.

Die Lösung:

Wir simulieren mit POKE 58,1 einen Programmablauf-Modus und verhindern so die oben genannte Fehlermeldung:
OPEN1,8,15:POKE58,1:FORI=1TO50:GET #1,A\$:
PRINTA\$;:CLOSE-(ST=64):IFST < > 64THENNEXT

(Jürgen Atrott/tr)

Auto-Befehl in Basic

Mit diesem Mini-Programm lassen sich Zeilennummern automatisch vorgeben. Gestartet wird das Programm mit A=Anfangszeilennummer: GOTO 1. Nun wird der Bildschirm gelöscht und die in A definierte Zeilennummer ausgegeben. Mit POKE 19,1 wird die Tastatur zum aktuellen Eingabegerät. Dadurch wird bei der nachfolgenden INPUT-Anweisung das Fragezeichen nicht mit ausgegeben. Nach der Eingabe der Zeile wird darunter ausgegeben: »A = nächste Zeilennummer: GOTO 1«. In den Tastaturpuffer wird »Cursor Home« und »RETURN« geschrieben, so daß die Zeile nach dem END-Befehl automatisch ins Programm übernommen wird.

```
1 PRINT "{CLR}"A;:POKE 19,1:INPUT A$:POKE 19,0:PRINT
2 PRINT "A="A+10":GOTO1":POKE 198,3:POKE
631,19:POKE 632,13:POKE 633,13:END
```

Das Programm wird mit RUN/STOP-RESTORE wieder verlassen.

(Herbert Kunz/tr)

Einzeiler zur Eingabe von Einzeilern

Es lassen sich Basic-Zeilen bis zu 88 Zeichen Länge eingeben. Man startet also den Einzeiler: Es wird dabei der Bildschirm gelöscht, der Cursor auf die dritte Bildschirmzeile gebracht, und in der vierten Zeile, Spalte 8, ein »I« ausgedruckt. Nun gibt man eine Zeile, beginnend bei der Cursorposition, ein, bis der Pfeil auf das letzte Zeichen der Eingabezeile zeigt. Auf dem Bildschirm ist nun also eine zweizeilige Basic-Zeile zu sehen und in der dritten Basic-Zeile noch acht weitere Zeichen. Das macht zusammen $2 \times 40 + 8 = 88$ Zeichen, welche auch nach einem RETURN angenommen werden. Warum steht der Cursor aber erst in der dritten Bildschirmzeile? Ganz einfach: Denn nun kann man in der ersten Bildschirmzeile ein LIST (Zeilennummer) eingeben. Die gelistete Basic-Zeile geht dann genau in den 88-Zeichenbereich. Aber am besten, man probiert den Einzeiler einfach selber mal aus.

Nun zum Programm selbst:

SYS 59749 erzeugt die wichtige Fortsetzungszeile, welche der C 64 selbständig nach 40 Eingabezeilen ausdrückt (Scrolen nach unten, nächste Zeile löschen). SYS 42112 ein Basic-Ende ohne »READY«. Der Vektor hierfür steht in den Speicherstellen 770 und 771.

```
1 PRINT "{CLR,2DOWN}";:SYS 59749:SYS 59749:
PRINT "{HOME,2DOWN}":PRINT TAB(7)"I":PRINT
{HOME,DOWN}":SYS 42112
```

(Jörg Peschel/tr)

Einzeiliger Taschenrechner

```
1 PRINT "{LIG.BLUE,CLR}"A$="A:INPUTA$.:PRINT
"{HOME,BLUE}A="A$":GOTO1":POKE631,19:POKE632,13:
POKE198,2
```

Mit diesem Einzeiler kann jede beliebige Rechenoperation innerhalb eines Programms durchgeführt werden.

Zum Programm: Beim ersten Start erscheint in der oberen Ecke ein »=0«, das ignoriert werden kann. In den INPUT-Befehl wird nun eine beliebige Rechenoperation eingegeben. Jetzt folgt der eigentliche Trick: Zuerst wird die Variable A, dahinter die Rechenoperation, die im String vorhanden ist und schließlich der Befehl »GOTO 1« (zum späteren Wiederstarten

des Programms) in die erste Bildschirmzeile geschrieben. Um zu verhindern, daß es ein kurzes Auftauchen der Variablenzuweisung gibt, ist der PRINT-Befehl der Bildschirmfarbe angepaßt. Nun wird zuerst ein »HOME«, gefolgt von einem »RETURN« in den Tastaturpuffer geschrieben und dieser auf die Länge von zwei Eingaben gebracht. Ist das Programm beendet, nimmt der Computer die Variablenzuweisung auf, startet das Programm erneut und gibt das Ergebnis aus.

(Thorsten Wanschura/tr)

Auto-DATA-Routine

Das Maschinenprogramm (Listing 4) muß mit dem MSE eingegeben werden. Es belegt keinen Basic-Speicher, sondern liegt im Speicherbereich von \$C000-\$C0CF. Nach dem Laden mit LOAD »Name«, Gerätenummer, 1 muß »NEW« eingegeben werden. Die Erweiterung wird mit SYS 49152 initialisiert und stellt dann folgende neue Befehle zur Verfügung:

@A Startzeilennummer, Schrittweite und

@D Startzeilennummer, Schrittweite.

Dabei darf die Startzeilennummer Werte von 0 bis 63999 annehmen und die Schrittweite darf 255 nicht überschreiten. Wird die Zeilennummer 63999 überschritten, so hat dies einen »ILLEGAL QUANTITY ERROR« zur Folge.

Der @A-Befehl gibt bei der Eingabe eines Basic-Programms die jeweils folgende Zeilennummer aus. Allerdings ist dieser Befehl wesentlich komfortabler als zum Beispiel der Befehl, der in der 64'er-Ausgabe 12/84 veröffentlicht wurde.

Es ist möglich, die zuletzt eingegebene Zeile zu korrigieren und danach das Abtippen fortzusetzen, ohne den Auto-Befehl zu verlassen. Die Zeilennummer wird dabei nicht stur weiter nach oben gezählt und die Zeile wird korrekt angenommen. Wer einmal mit der Routine von Frank Siedel oder auch mit der Auto-Routine von Hypra-Ass gearbeitet hat, wird diesen Vorteil sehr zu schätzen wissen.

Der @A-Befehl kann zum Beispiel durch Drücken der Return-Taste verlassen werden.

Der @D-Befehl gibt zusätzlich zu der Zeilennummer am Anfang jeder Zeile noch ein »DATA« aus. Er kann verlassen werden, indem man vor dem Drücken der Return-Taste das »DATA« mit der INST/DEL-Taste löscht oder die Cursor-down-Taste betätigt. In Verbindung mit dem Checksummer stellt diese Erweiterung eine große Hilfe beim Abtippen von Programmen dar.

(Florian Gallwitz/tr)

```

programm : auto.obj          c000 c0d0
c000 : a9 20 a0 c0 8d 02 03 8c 08
c008 : 03 03 a9 9a a0 c0 8d 08 a1
c010 : 03 8c 09 03 a9 00 85 02 b1
c018 : 60 a9 00 85 02 4c a8 b2 07
c020 : a6 02 f0 3b a5 fb a4 fc 32
c028 : c0 fa b0 ed 85 63 84 62 99
c030 : a2 90 38 20 49 bc 20 df e7
c038 : bd a2 00 bd 00 01 f0 06 d6
c040 : 9d 77 02 e8 d0 f5 a9 20 da
c048 : 9d 77 02 a4 02 88 f0 0c f6
c050 : e8 a9 44 9d 77 02 e8 a9 50
c058 : c1 9d 77 02 e8 86 c6 20 24
c060 : 60 a5 86 7a 84 7b 20 73 0f
c068 : 00 aa f0 29 a2 ff 86 3a d7
c070 : 90 06 e8 86 02 4c 96 a4 34
c078 : 20 6b a9 a5 14 a4 15 18 58
c080 : 65 fd 85 fb 90 01 c8 84 02
c088 : fc a6 7a bd 00 02 d0 02 85
c090 : 85 02 4c 9f a4 85 02 4c 34
c098 : 5f c0 20 73 00 08 c9 40 b6
c0a0 : f0 04 28 4c e7 a7 20 73 49
c0a8 : 00 c9 41 f0 07 c9 44 f0 ad
c0b0 : 07 4c a2 c0 a2 01 d0 02 17
c0b8 : a2 02 86 02 20 73 00 20 1b
c0c0 : eb b7 86 fd a5 14 a4 15 a0
c0c8 : 85 fb 84 fc 4c 20 c0 00 d4

```

Listing 4. Die Auto-DATA-Routine

Zahlenratespiel

Die Zahlenobergrenze ist veränderbar, indem man die Zahl 10^8 (=100.000.000) variiert. Durch Verändern der ersten FOR...NEXT-Schleife kann man die Anzahl der Rateversuche erhöhen beziehungsweise erniedrigen. Das eigentliche Herz des Programms ist die SGN-Funktion. Durch sie wird die Variable D definiert. Nach jedem Rateversuch wird der derzeitige Wert dieser Variable am linken Bildschirmrand ausgegeben: »-1« bedeutet, daß die zu erratende Zahl kleiner ist als die eben eingetippte. »1« bedeutet, daß die Zahl größer ist. Bei »0« hat man die gesuchte Zahl erraten, und der Computer denkt sich automatisch eine neue aus. Die bisherige Anzahl der Versuche wird immer rechts des D-Wertes ausgegeben.

7PRINT"NEUE ZAHL":A=INT(RND(1)*1018):FORB=1TO99:
INPUTC:D=SGN(A-C):PRINTD,B:OND+1GOTO7:NEXT
Alle Befehle müssen abgekürzt werden! (Felix Vogler/tr)

Der Hypra-POKE

POKE 57818,8. Nachdem man Hypra-Load geladen und gestartet hat, braucht man nach Eingabe dieses POKes bei den Befehlen LOAD und SAVE nie wieder das »8« am Schluß anhängen. Die Defaultnummer (das ist die Nummer, die genommen wird, wenn keine angegeben wird) für die Geräteadresse wird von vormals 1 (Datasette) auf 8 (Diskette) gestellt.

(Ernst Hofler/tr)

Tips zum Apfelmännchen

Aufgrund der großen Resonanz und vieler Anfragen, bringen wir hier die Daten der in Ausgabe 11/85 auf Seite 169 veröffentlichten Bilder.

Bild 1: -0.7; 2.1; -1; 1; 30

Bild 2: 1.67; 1.86; -.75; 0.75; 40

Bild 3: 0.1429; 0.1802; 1.02; 1.0477; 100

Bild 4: 0.5665; 0.5737; 0.5602; 0.5665; 85

Bild 5: 0.7425; 0.74825; 0.09621; 0.10067; 150

Bild 6: -0.103; 0.379; 0.618; 0.929; 50

Doch damit nicht genug, wir haben noch ein paar mehr:

0.7654; 0.76722; 0.10021; 0.10151; 200

-0.0171; -9.44e-3; 0.6534; 0.658; 150

0.22223; 0.2545; 0.7089; 0.7421; 150

0.76461; 0.76498; 0.10056; 0.10082; 254

(Gerhard Pehland/og)

Anpassung des Apfelmännchen an Datasette

Folgende Änderungen sind zur Anpassung mit Turbo-Tape notwendig:

300 ?"down]"tab(10)"(5){2space} Effekt"

310 entfällt

315 entfällt

340 get a\$:if a\$<"1" or a\$>"5"then 340

360 on val (a\$)gosub 1000,2000,3000,4000,70003

4020-s nb\$

4030 return

Die Zeilen 4040 bis 5720 entfallen. Will man ein Bild sofort nach dem Berechnen speichern, so antwortet man im Programm mit »j«, und drückt, nachdem der Computer zu zeichnen begonnen hat, Record und Play.

(Dirk Trossen/og)

Literaturhinweise zu Apfelmännchen:

In den Zeitschriften »Spektrum der Wissenschaft«, Ausgabe 10/85 und »geo« sind ausführliche Berichte zur Mandelbrotmenge (Apfelmännchen) enthalten. (Dietmar Horst/og)