

Tips & Tricks für Profis

Endlich: Mit unserem Ascompiler aus der Ausgabe 1/86 und einem kleinen Trick ist es möglich, Interrupt-Routinen in Basic zu schreiben! Erwähnenswert ist auch das Windowing für den C 64!

Kannten Sie den schon? Wenn Sie ein Programm mit »SAVE CHR\$(160) + CHR\$(20) + "Name" + CHR\$(34),8« auf Diskette speichern, so ist es nur noch mit genau derselben Zeichenfolge wieder in den Computer zu bekommen.

Ein »LOAD "??Name*" ,8« weist die Floppy mit einem »file not found error« zurück. Für Nicht-Eingeweihte eine harte Nuß! (tr)

C 64-Programme auf den CBM-Computern

Vorgehensweise:

Laden des C 64-Programms (zum Beispiel von Kassette)

sY 1024 eintippen (ruft den Monitor auf)

.m 0400 0401 eintippen (genau ein Leerzeichen nach »m« ist wichtig!)

Mit dem Cursor in die angezeigte Zeile hochgehen und ändern:

.. 0400 00 01 08 00 00 00 ... Rest egal;

mit <RETURN> verlassen!

.x

0

Fertig!!!

Zur Funktionsweise:

Mit dem Monitor wird eine leere Basic-Zeile 0 erzeugt. Der Zeiger auf die nächste Zeile wird auf den Anfang des C 64-Programms, nämlich \$0801 gesetzt.

.x bewirkt einen Rücksprung in das normale Basic

0 (RETURN) löscht diese Zeile 0 und verschiebt gleichzeitig das Programm an den im CBM üblichen Platz.

Diese Methode funktioniert auch dann noch, wenn das C 64-Programm bereits die Zeile 0 enthält. (Curt Bauer/tr)

Hypra-Platos und Star SG 10

Durch Eingabe einer neuen MSE-Zeile (mit der MSE-Funktion »Control-N«) im Programm »2. PRG« kann man Hypra-Platos auf den inzwischen weit verbreiteten Star-Drucker SG 10 anpassen. Dies betrifft die Adresse 39289 (\$9979). In dieser Speicherstelle wird die Größe des Zeilenvorschubs festgelegt. Er muß beim Star SG 10 den Wert \$0E anstatt \$16 (Epson-Drucker) besitzen.

name : 2. prg 9100 9cba

9978 : a9 0e 20 d2 ff 20 33 f3 40

(Klaus Lilienthal/tr)

Y und Z dauerhaft getauscht

Wer ärgert sich nicht über die vertauschte Anordnung der Buchstaben »Y« und »Z« auf dem C 64? Besonders diejenigen, die einen Schreibmaschinenkurs erfolgreich absolviert haben, sind gezwungen sich nun wieder umzustellen. Dabei müßte die Forderung heißen: Der Computer soll umgestellt werden!

Die meisten Textverarbeitungsprogramme, die auch deutsche Umlaute liefern, haben gleichzeitig die beiden oben genannten Zeichen DIN-Norm-gerecht vertauscht. Wenn man ein solches Textprogramm wieder ausschaltet, ist alles beim alten, das heißt die beiden Buchstaben liegen wieder an der falschen Stelle. Es gibt zwar einen Trick, wie man mit einem kleinen Programm die Tastatur softwaremäßig verändert, dieses Programm versagt aber bei dem weitverbreiteten Textprogramm »VIZAWRITE«, bei dem die Autoren beim Einbau der deutschen Umlaute leider nicht auch den Tausch von »Y« und »Z« vorgenommen haben. Daher wurde eine Lösung gesucht, wie man die Änderung dauerhaft vornehmen könnte: Man brennt sich einfach ein neues Kernel. Dazu verschiebt man es mit Hilfe der Transfer-Funktion eines Monitors (zum Beispiel mit dem Befehl »T E000 FFFF 1000«) nach \$1000.

Jetzt geht man mit dem Monitor an die Speicherstellen ab \$1B8D (enthält die Tastenbelegung des C 64) und findet auf dem Bildschirm eine Anordnung von Speicherstellen vor, die entsprechend dem Bild 1 aussehen, wobei diese Hardcopy schon die Änderungen enthält, so daß man beim Vergleich genau erkennen kann, welche Bytes vertauscht wurden. Die jeweiligen zu ändernden Zahlen stellen den Code des Zeichens in hexadezimaler Schreibweise dar. (Stefan Zellin/tr)

\$1b88 entspricht Speicherstelle \$eb88

.. 1b88 11 33 57 41 34 52 53 45
.. 1b90 01 35 52 44 36 43 46 54
.. 1b98 58 37 5a 47 38 42 48 55

.. 1bc8 8b 91 23 d7 c1 24 d9 d3
.. 1bd0 c5 01 25 d2 c4 26 c3 c6
.. 1bd8 d4 d8 27 da c7 28 c2 c8

.. 1c08 8a 8b 91 96 b3 b0 97 b7
.. 1c10 ae b1 01 98 b2 ac 99 bc
.. 1c18 bb a3 bd 9a ad a5 9b bf

Bild 1.
Die unterstrichenen
Adressen müssen
geändert werden

Schreiberling und Citizen 120 D

Bei einer Computeranlage mit Drucker »Citizen 120 D« und Interface »Wiesemann 92000/G« müssen folgende Zeilen geändert werden:

Zeile 620 OPEN 1,4,1 : Z=PEKK(BZ)

Zeile 800 POKE 768,61:OPEN 1,4,1:PRINT #1,L\$;:CLOSE
1:POKE 768,139

Mit dieser kleinen Änderung läuft der Schreiberling V2 (Ausgabe 2/86) nun fehlerfrei. Wir haben bei der ersten Version (Heft 10/85) alle OPEN-Befehle auf »1,4,1« geändert und auch diese Version läuft fehlerfrei.

Wir hoffen, daß wir mit diesem kleinen Tip manchem Freak helfen können. (Angelika Bachhuber/tr)

Versteckspiel mit dem Directory

Ein weiterer Trick beim Diskettenformatieren besteht in der Eingabe der Zeichen "N:" + CHR\$(20) + CHR\$(20) + CHR\$(20) + CHR\$(0) + CHR\$(0) + CHR\$(0) + "Name,ID" beim Formatieren.

So wird die normalerweise vor dem Disknamen stehende »Zeilennummer« 0 und der Anführungsstrich gleich nach dem Ausdrucken auf dem Bildschirm von den drei CHR\$(20) (entsprechen dreimaligem Drücken der <DELETE>-Taste) gelöscht. Danach wird durch drei Nullbytes das Programmende angenommen und das Directory-Listen abgebrochen. Dennoch lassen sich gespeicherte Programme wieder laden!

(Henning Zipf/tr)

Einzelstschrittmodus für Basic-Programme

Folgendes Listing (siehe weiter unten) ist wohl das kürzeste Single-Step-Programm der Welt.

Dieses Programm wird mit SYS 49152 aufgerufen. Der erste Teil schreibt in die Adresse \$0308 und \$0309 (Zeiger auf eine Routine, die CHRGET aufruft) die Adresse \$C00B. Wenn nun der Interpreter ein neues Zeichen aus dem Basic-Programm holt, läuft er so lange durch eine Prüfschleife, bis Sie <SHIFT> gedrückt haben. Nun macht er normal weiter, bis er erneut ein Zeichen holt. Um den Single-Step-Modus auszu-schalten, müssen Sie das Maschinenprogramm ab \$C01C aufrufen. Es schreibt in \$0308 und \$0309 wieder die alten Adressen. Nun aber zum Assembler-Listing:

```
C000 LDA #$0B      ;Low-Byte von $C00B ...
C002 STA $0308     ;auf $0308 ...
C005 LDA #$C0      High-Byte von $C00B ...
C007 STA $0309     ;auf $0309 ...
C00A RTS          ;Rücksprung ins Basic.
C00B LDA $028E     ;<SHIFT>-Muster von der Tastatur
C00E BNE $C013     ;bei <SHIFT> nach $C013 ...
C010 JMP $C00B     ;sonst nach $C00B zurück.
C013 JSR $0073     ;Zeichen aus dem Basic-Programm;
C016 JSR $A7ED     ;macht bei Commodore-Routine weiter;
C019 JMP $A7AE     ;macht bei Commodore-Routine weiter;
C01C LDA #$E4      ;Low-Byte von $A7E4 ...
C01E STA $0308     ;auf $0308 ...
C021 LDA #$A7      ;High-Byte von $A7E4 ...
C023 STA $0309     ;auf $0309.
C026 RTS          ;Rücksprung ins Basic.
```

(Gabriel Range/tr)

Abhilfe bei Resets durch Störimpulse

Es hat sich bei mir mehrfach herausgestellt, daß der C 64 auf Störimpulse aus dem 220V-Netz mit einem Reset reagiert. Die »Überempfindlichkeit« des Computers kann sogar dazu führen, daß das Einschalten induktiver Lasten (Neonröhren in Nebenräumen) einen Reset des Computers bewirkt. Damit wird ein längeres Arbeiten mit dem Computer fast unmöglich. Erstaunlicherweise ist die Peripherie des C 64 dafür verantwortlich (was erklärt, warum das Problem oft erst längere Zeit nach dem Kauf des C 64 auftritt):

Es stellte sich nämlich heraus, daß die »Überempfindlichkeit« verschwand, wenn die Peripheriegeräte (Floppy, Drucker und Plotter) vom seriellen Bus des Computers entfernt wurden. Der Verdacht auf ein zu langes Buskabel erwies sich aber als falsch. Mit einem Oszilloskop wurde nun insbesondere die Resetleitung des seriellen Busses mit, beziehungsweise ohne angeschlossene Peripheriegeräte beobachtet. Es zeigte sich dabei, daß die Geräte die Resetleitung (als sogenannter »activ low«-Eingang muß sie einen 5V-Pegel im Normalzustand haben) zusammen so belasteten, daß die Spannung schließlich bis auf fast den halben Wert absank.

Traten zusätzlich Störimpulse auf, so überlagerten sie sich auf die Resetspannung in Form hochfrequenter Nadelimpulse. Sie führten aber nur deswegen zu einem Reset, weil außerdem die Resetspannung durch die Belastung bereits zu niedrig war!

Die Abhilfe gegen die Störimpulse ist nun sehr einfach: Es muß nur erreicht werden, daß die Resetspannung auch bei Belastung nicht zu stark absinkt. Man schaltet dazu einfach einen Widerstand von der Resetleitung gegen +5V und eventuell zur Unterdrückung der Störimpulse parallel dazu noch zusätzlich einen Kondensator. Geeignete Werte hierfür sind zum Beispiel 1 Mikrofara und 2,2 K Ω .

Auf diese Weise lassen sich die offenbar verbreiteten Probleme mit Störimpulsen häufig beheben; dabei ist dieser Trick nicht so teuer wie der Versuch, durch einen Entstörfilter Abhilfe zu schaffen.

(Klaus Kuhl/tr)

Windowing auf dem C 64

Mit folgendem Einzeiler ist es möglich, Bildschirmfenster zu definieren, so daß sich sämtliche Bildschirmen- und -ausgaben nur noch in einem bestimmten Bereich abspielen:

```
20 POKE59639, (AN-1)AND255:POKE59652, EN:POKE59522, EN+1:
POKE58719, EN:PRINT "(HOME) "
```

Die Variablen AN und EN geben die Zeilen an, von wo bis wo das Bildschirmfenster reichen soll. AN enthält die oberste Zeile des Bildschirmfensters, EN die unterste. Ein Beispiel: Das Bildschirmfenster soll von der zweiten bis zur zwölften Zeile reichen:

```
20 AN=2:EN=12:POKE59639, (AN-1)AND255:POKE59652, EN:POKE
59522, EN+1:POKE58719, EN:PRINT "(HOME) "
```

Hierbei muß man die im Handbuch aufgeführten Abkürzungen benutzen, da die Zeile sonst länger als die erlaubten achtzig Zeichen wird.

Die beiden Einzeiler setzen voraus, daß das Betriebssystem ins RAM kopiert wurde. Wenn man für diesen Zweck nicht auf Maschinensprache zurückgreifen will, kann man folgende Zeile benutzen.

```
10 A=16384:FORI=40960TO49151:POKEI, PEEK(I):POKEI+A, PEEK
(I+A):NEXT:POKEI, 53
```

Diese Art des Erzeugens von Bildschirmfenstern hat einen Nachteil. Die Taste <CLR> löscht auch den Bildschirm über dem Bildschirmfenster, die Taste <CLR/HOME> setzt den Cursor immer noch nach links oben in den Bildschirm und man kann das Bildschirmfenster mit Cursor-Up nach oben verlassen. Außerdem kann es unter Umständen geschehen, daß es zu Fehlfunktionen kommt, wenn der Cursor in der letzten Zeile des Bildschirmfensters steht und man nun über den rechten Rand schreibt. Diese Nachteile sind jedoch sehr leicht zu umgehen, da diese Funktionen selten nötig sind.

Die Bildschirmfenster können auch in Maschinensprache verwendet werden, wenn Ausgaben über die Kernelroutine »BSOUT« mit Einsprung \$FFD2 erfolgen.

Um sich von den Möglichkeiten zu überzeugen, kann man die folgenden Zeilen eingeben:

```
30 POKE1, 55:FORI=1TO999:PRINT "A ";:NEXT:PRINT "{ HOME,
4 DOWN} ":POKE1, 53
40 PRINTSPC(RND(1)*25) "HALLO":GOTO40
```

Eine mögliche Anwendung wäre zum Beispiel bei Adventures die Generierung eines Eingabe- und eines Ausgabefensters, so daß unten der Benutzer seine Eingaben macht, und im oberen Teil der Computer die Reaktionen zeigt.

(Michael Patra/tr)

IRQ-Routinen in Basic mit dem Ascompiler

Mit dem Ascompiler hat der Basic-Programmierer die Möglichkeit, zeitkritische Unterprogramme in Maschinensprache zu realisieren. Manche Routinen (zum Beispiel Musik, Spritebewegung) hätte man aber schon gern als Interrupt. Dies ist mit dem Ascompiler leider nur durch Tricks möglich. Eine Methode werde ich nun beschreiben.

Beim Programmieren der Interruptroutine müssen folgende Regeln beachtet werden:

- Die Ablaufzeit eines Durchgangs sollte nicht zu lang werden (zum Beispiel bei Musik nicht erst am Ende des Stücks die Routine verlassen, sondern in regelmäßigen Abständen zwischen und während den einzelnen Tönen), da sonst die normale Arbeit des Computers sehr verlangsamt würde.

- Die Interruptroutine wird mit »SYS59953« verlassen.

- In der Interruptroutine darf nur einmal der Befehl »SYS 59953« stehen. Zum Initialisieren der Interruptroutine muß sie vorher kompiliert werden. Dann das Programm »IRQ INIT« eingeben, beziehungsweise laden (Listing 1), in Zeile 10 und in Zeile 20 das Lo- und Hi-Byte der Anfangsadresse der Interruptroutine ändern. Nun wird das Programm »IRQ INIT« com-

piliert und gestartet. Der Interrupt sollte dann laufen. Wenn man ihn mit <RUN/STOP-RESTORE> stoppt, darf man das Programm »IRQ INIT« nicht nochmal aufrufen. Dann nimmt man zum Initialisieren das kompilierte Programm »IRQ INIT2« (Listing 2), welches eine gekürzte Form der Routine »IRQ INIT« ist. Das Demo-Programm (Listing 3) zeigt, wie man eine Funktionstastenbelegung als Interrupt programmiert. Es fragt die Adresse 197 nach der zur Zeit gedrückten Taste ab. Wenn es eine Funktionstaste ist, wird in den Tastaturpuffer eine Zeichenfolge gePOKEt.

Beschreibung des Programms »IRQ INIT«:

Zeile 10—20	Lo- und Hi-Byte speichern.
Zeile 30—160	Suchen nach der Zahlenfolge 32,49,234, welche dem Maschinenbefehl »JSR \$EA31«, beziehungsweise in Basic »SYS89953«, entspricht.
Zeile 170	Ändert »JSR \$EA31« in »JMP \$EA31« um (zum Interrupt verlassen).
Zeile 210	Setzt Speicher A auf 24696, dann steht in Adresse 828 der Wert 120 und in 829 der Wert 96.
Zeile 220	Dies entspricht den Befehlen »SEI« und »RTS«.
Zeile 230—240	Startet die zwei Befehle, um einen Interrupt zu verhindern.
Zeile 250	Interruptvektoren auf eigene Routine setzen.
Zeile 260	Wie Zeile 210, nur mit den Befehlen »CLI« und »RTS«.
Zeile 260	Interrupts wieder freigeben.

Vorgehensweise:

1. Programm »IRQ INIT2« laden und compilieren. Die Startadresse ist die Startadresse des fertigen Programms. »Endadresse + 1« aufschreiben.
2. Lo- und Hi-Byte im Programm »IRQ INIT2« auf die »Endadresse + 1« ändern und noch mal compilieren.
3. Ihre eigene Interruptroutine laden und compilieren. Die Startadresse ist die aufgeschriebene »Endadresse + 1«.
4. Programm »IRQ INIT« laden, Lo- und Hi-Byte ändern, compilieren und starten.
5. Fertiges Programm speichern. (Thomas Froitzheim/tr)

```

9 REM LO BYTE INTERRUPT ANFANG
10 LET L=00
19 REM HI BYTE INTERRUPT ANFANG
20 LET H=128
30 LET A=H*256
40 LET A=A+L
50 LET B=A
60 LET C=PEEK(B)
70 IF C<32 THEN 190
80 IF C>32 THEN 190
90 LET C=B+1
100 LET C=PEEK(C)
110 IF C<49 THEN 190
120 IF C>49 THEN 190
130 LET C=B+2
140 LET C=PEEK(C)
150 IF C<234 THEN 190
160 IF C>234 THEN 190
170 POKE B,76
180 GOTO 210
190 LET B=B+1
200 GOTO 60
210 LET A=24696
220 SYS 828
230 POKE 788,L
240 POKE 789,H
250 LET A=24664
260 SYS 828

```

Listing 1. »IRQ INIT«

```

10 LET L=00
20 LET H=128
30 LET A=24696
40 SYS 828
50 POKE 788,L
60 POKE 789,H
70 LET A=24664
80 SYS 828

```

Listing 2. »IRQ INIT2«

```

10 LET C=PEEK(197)
11 IF X=1 THEN 440
12 IF X=2 THEN 560
20 IF B=C THEN 700
30 LET B=C
40 IF C=4 THEN 60
50 GOTO 120
60 POKE 198,5
70 POKE 631,76
80 POKE 632,75
90 POKE 633,83

```

```

100 POKE 634,84
110 POKE 635,13
120 IF C=5 THEN 140
130 GOTO 190
140 POKE 198,4
150 POKE 631,82
160 POKE 632,85
170 POKE 633,78
180 POKE 634,13
190 IF C=3 THEN 210
200 GOTO 380
210 POKE 198,8
220 POKE 631,76
230 POKE 632,207
240 POKE 633,34
250 POKE 634,36
260 POKE 635,34
270 POKE 636,44
280 POKE 637,56
290 POKE 638,13
300 IF C=6 THEN 320
310 GOTO 700
320 POKE 198,10
330 POKE 631,76
340 POKE 632,207
350 POKE 633,29
360 POKE 634,29
370 POKE 635,29
380 POKE 636,29
390 POKE 637,29
400 POKE 638,29
410 POKE 639,29
420 POKE 640,29
430 LET X=1
435 GOTO 700
440 POKE 198,10
450 POKE 631,29
460 POKE 632,29
470 POKE 633,29
480 POKE 634,29
490 POKE 635,29
500 POKE 636,29
510 POKE 637,29
520 POKE 638,29
530 POKE 639,29
540 POKE 640,29
550 LET X=2
555 GOTO 700
560 POKE 198,7
570 POKE 631,29
580 POKE 632,29
590 POKE 633,44
600 POKE 634,56
610 POKE 635,44
620 POKE 636,49
630 POKE 637,13
640 LET X=8
700 SYS 59953

```

Listing 3. Ein Demo-Programm zu Listing 1 und 2

0 64'er

AKUSTIKKOPPLER HITRANS 300 C

Mit einem Akustikkoppler öffnen Sie Ihrem Computer das Tor zur ganzen Welt. Der HITRANS 300 C stach im Akustikkoppler-Test der Ausgabe 3/86 durch die besten Übertragungseigenschaften hervor. Sie erhalten ihn bei uns als Fertiggerät, lediglich eine Blockbatterie muß eingesetzt und das Gehäuse zugeschraubt werden. Sie können den Koppler auch über ein 12-Volt-Netzteil, das in jedem Elektronikgeschäft preisgünstig erhältlich ist, betreiben. Die Bauanleitung für ein RS 232-Interface finden Sie in der Ausgabe 3/85.

Preis für Akustikkoppler

HITRANS 300 C

(ohne Batterie)

Achtung: Nicht für Wiederverkäufer

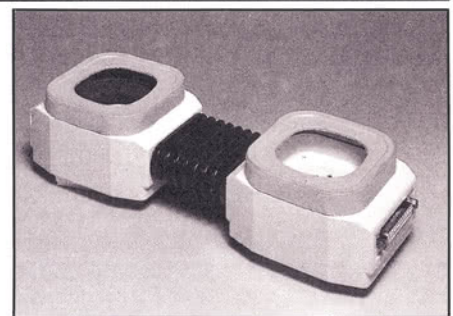
Bisher
~~DM 248,-~~

Jetzt nur noch

DM 198,-* (sFr. 178,-)

* inkl. MwSt. Unverbindliche Preisempfehlung

Bestellnummer: HW 072



Betriebssoftware auf Diskette

Bestellnummer: HW 071 **DM 14,80*** sFr. 13,90
Die Betriebssoftware befindet sich außerdem auf der Programm-Service-Diskette des 64er-Sonderheftes SH 7/85.

Bitte verwenden Sie für Ihre Bestellung immer die abgedruckte Postgiro-Zahlkarte oder einen Verrechnungsscheck.
Sie erleichtern uns damit die Auftragsabwicklung, und dafür berechnen wir Ihnen keine Versandkosten.

Bestellungen aus der Schweiz bitte direkt an:
Markt & Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Tel. 042/41 56 56

Bestellungen aus Österreich bitte direkt an:
Ueberreuter Media Handels- und Verlagsges. mbH, Alser Straße 24, 1091 Wien,
Tel. 02 22/48 15 38-0

Markt & Technik

Unternehmensbereich Buchverlag
Hans-Pinsel-Straße 2, 8013 Haar bei München