

# Erste Hilfe für den C 64: RENEW

**Manchmal läuft auch alles schief! Da waren die Finger wieder einmal schneller als die Gedanken. Man hat NEW eingetippt und erst nach dem Drücken der RETURN-Taste wird einem der angerichtete, ungewollte Schaden klar: Das Programm ist weg! Oder?**

Die VC 20-Besitzer können jetzt nur lächeln — wenn sie die Ausgabe 4 kennen. Dort ist nämlich ein Programm abgedruckt, daß ein mit NEW gelöscht Programm wieder sichtbar macht. Hier ist die angepaßte Version für den C 64.

Eine Programmbeschreibung erübrigt sich, da in der Ausgabe 4 Helmut Welke dies bereits sehr ausführlich gemacht hat. Es sind auch nur sehr kleine Änderungen zu machen. Im Originallisting für den VC 20 (4/84 Seite 89) sind in Zeile 1005 der Wert 207 durch 175 und in Zeile 1015 der Wert 198 durch 166 zu ersetzen und schon läuff's auch im C 64. Bei diesem hilft dieses RENEW auch nach einem Reset oder SYS 64738. Auch dieses Programm kann selbstverständlich mit dem DATA-WANDLER abgespeichert werden.

## Listing von »RENEW«

```
100 REM *** RENEW ***
110 FOR I=678 TO 755:READ A:POKE I,A:NEXT
111 DATA 165,43,24,105,4,133,253,165,44,1
112 DATA 05,0,133,254,160,0,177,253,240,8
113 DATA 200,192,88,208,247,76,8,175,200,
114 DATA 152,160,0,24,101,253,145,43,133,253
115 DATA 144,2,230,254,165,254,200,145,43
116 DATA 136,177,253,170,200,177,253,240
117 DATA 7,133,254,134,253,76,213,2,165,0
118 DATA 24,105,2,133,45,165,254,32,85,166
119 DATA 76,156,166
READY.
```

## Datawandler

**Mit dem Datawandler ist es auch dem »Nur-Basicprogrammierer« möglich, Maschinensprachprogramme (oder Teile davon), die in Form eines Basic-Laders (also über DATAs und POKE-Befehle) eingetippt wurden, als Maschinensprache abzuspeichern. Dadurch ergibt sich auf der Diskette eine Platz-, beim Laden eine Zeitersparnis.**

»Bitte warten — ich lese Daten« — so oder ähnlich wird der »Basicprogrammierer« nach dem Programmstart darauf aufmerksam gemacht, daß bei jedem RUN eines Basicprogramms die Maschinenspracheteile DATA für DATA in die Speicherzellen gePOKEt werden. Warum also nicht die DATAs gleich wieder als Maschinensprache abspeichern! Aber es ergibt sich nicht nur eine Zeitersparnis beim Programmablauf, sondern auch beim Laden, da auf der Diskette oder Kassette weniger Speicherplatz benötigt wird (zum Vergleich: der Schatzsucher aus der Ausgabe 6 belegt auf der Diskette als Basicprogramm 72 Blöcke, als Kombination Basic/Maschinensprache nur noch 44 — außerdem startet dieses Programm dann in Sekundenbruchteilen).

Darüber hinaus bestehen viele Hilfsprogramme, die auf Maschinenroutinen zurückzugreifen, nur aus der Zeile »FOR I=xTOxx:READx:POKEI,x:NEXTI« und vielen, vielen DATAs. Derartige Programme bieten sich für den Datawandler von selbst an, da sie nach der Umformung geladen werden können, ohne ein eventuell im Basicspeicher stehendes Programm zu zerstören.

### Zum Programm selbst

Vor dem Start des Datawandlers müssen die Daten im Speicher stehen — falls sie nur in Form von DATA-Zeilen vorhanden sind, wird in Zeile 60130 eingefügt:

```
60130 FOR I=AA TO EA:READX:POKEI,X:NEXTI
```

In den Zeilen 60030 — 60040 wird (in dezimaler Form) die Anfangs- und Endadresse abgefragt, unter der die DATAs »abgelegt« sind und den Variablen AA beziehungsweise EA zugeordnet. Die Zeilen 60050 bis 60100 dienen der Abklärung, ob das abzuspeichernde Programm auch wieder an die Adresse geladen werden soll, wo derzeit die DATA stehen (durch das Verschieben der Ladeadresse ist auch ein leichteres Experimentieren in den Autostartbereichen möglich, die dem Programmierer gelegentlich nach den ersten »POKES« das Konzept aus der Hand nehmen).

Wird die Abfrage mit »N« beantwortet, so wird nach dieser späteren Ladeadresse gefragt — lautet die Antwort »J«, so wird die Ladeadresse = derzeitige Anfangsadresse (LA=AA) und nach dem Namen gefragt, unter dem das Programm nun