

# Hardcopy mit einer Zeile

**Dieser Einzeiler gibt den Text einer Bildschirmseite des C 64 auf einem MPS 801 aus. Ganz ohne PEEKs, POKEs und SYS-Befehle. Trotzdem überrascht das Programm mit einer hohen Druckgeschwindigkeit.**

Mittlerweile existiert eine Vielzahl von Hardcopy-Routinen für die verschiedenen Drucker und Bildschirmformate. Ich kannte aber bisher kein Druck-Programm, das universell einsetzbar ist, keine POKEs und PEEKs braucht und auch nicht in Maschinensprache geschrieben ist. Diese Eigenschaften weist dieser Einzeiler (Zeile 9 des Listings) auf. Das »Programm« enthält nur Basic-Befehle.

Die Programmidee basiert auf der Tatsache, daß der Bildschirm des C 64 wie eine Datei behandelt werden kann. Ähnlich wie der Drucker die Adresse 4 hat und die Floppy mit Adresse 8 angesprochen wird, ist dem Bildschirm die Geräteadresse 3 zugeordnet; fest vom Betriebssystem. Mit OPEN 2,3 wird die Bildschirmdatei eröffnet und mit INPUT #2 oder GET #2 der Inhalt der Datei gelesen. Beim INPUT #-Befehl ist, wie bei jedem anderen INPUT zu beachten, daß die Anweisung bei einem Komma oder einem Doppelpunkt abgebrochen wird. Aus diesem Grund habe ich den INPUT- durch den GET-Befehl ersetzt. Eine GET-Schleife liest ab der aktuellen Cursorposition sämtliche Zeichen als ASCII-Code. Die gelesenen Codes addiert man darauf in einer Variablen. Zum Beispiel durch die Anweisung  $A\$ = A\$ + B\$,$  bei der immer wieder ein neues B\$ zu A\$ addiert wird. Eine einfache Sache sollte man meinen, aber die Schwierigkeiten beginnen erst an dieser Stelle. GET liest nämlich immer als letztes Zeichen einer Zeile grundsätzlich ein Carriage Return, CHR\$(13). Egal, welches Zeichen sich dort befindet. Ferner ist der 80-Zeichen-Editor des C 64 zu beachten, der bei diesem Vorhaben zusätzliche Schwierigkeiten bereitet: Eine Zeile besteht trotz der 40-Zeichen-Darstellung auf dem Bildschirm innerhalb des C 64 aus 80 Zeichen. Denken

Sie nur an die Eingabe von langen Programmzeilen. Nach dem Öffnen der Bildschirmdatei wird Zeichen für Zeichen gelesen, beginnend in der oberen linken Bildschirmecke. Dabei werden in einem Schleifendurchgang immer zwei Zeichen geholt und den Variablen A\$ und B\$ zugeordnet. Ist A\$ gelesen, wird mittels der DELETE-Funktion, CHR\$(20), der restliche Zeileninhalt um eine Stelle nach links geschoben, um das letzte Zeichen einer Zeile eindeutig zu erkennen. Nach dem Einlesen von B\$ wird durch die INSERT-Funktion, CHR\$(148), der ursprüngliche Zeilenzustand wieder hergestellt. Danach wird der ASCII-Code in den beiden Variablen an den Drucker gesandt. Durch ständige, schnell ablaufende Zeilenverschiebung ist deutlich sichtbar, wo der Computer jeweils »arbeitet«; ein brauchbarer Nebeneffekt.

Ein Einzeiler kann natürlich kein perfektes Programm sein. Deshalb sollten auch die Nachteile herausgestellt werden: Die Routine kann keine reversen Zeichen auf dem Drucker darstellen. Reverse Zeichen werden normal gedruckt. Innerhalb des Textes, von dem eine Hardcopy angefertigt werden soll, dürfen keine Anführungsstriche, CHR\$(34), vorkommen. Da der Code 34 innerhalb eines Programms recht selten vorkommt, ist diese Einschränkung aber nur von geringer Bedeutung. Anders in Programm listings, die aber meistens mit OPEN1,4: CMD1:LIST gedruckt werden.

Der Einzeiler ist dafür gedacht, als Unterprogramm Daten auszudrucken; einfach vom Bildschirm weg.

## Tips zur Eingabe

Das reverse »t« steht für DELETE, CHR\$(20). Um diesen Code in der erforderlichen Form eingeben zu können, bedarf es eines kleinen Tricks: An der gewünschten Stelle tippt man zuerst zwei aufeinanderfolgende Anführungszeichen »« ein. Danach drückt man einmal Cursor-Links, damit der Cursor über dem zweiten Anführungszeichen steht und einmal INSERT und DELETE. Mit Cursor-Rechts geht man in die ursprüngliche Schreibposition zurück. Beachten Sie dabei bitte, daß das reverse »t« nur im Moment der Zeilenerfassung sichtbar ist. Nach LIST wird dieser Code vom Editor nicht gezeigt, sondern ausgeführt. Was zu der verstümmelten Version PRINT"; führt. Die anderen beiden revers dargestellten Steuerzeichen sind Cursor-Links, CHR\$(157) und INSERT, CHR\$(148). Nach dem ersten Anführungszeichen drücken Sie dazu Cursor-Links und anschließend INSERT. Sollten Sie mit der Eingabe der Steuerzeichen Probleme haben, können Sie sie durch CHR\$( )-Befehle ersetzen. Sie benötigen dann allerdings mindestens zwei Zeilen.

Die Zeile 4 dient lediglich zum Füllen des Bildschirms für Demo-Zwecke. Zeile 7 ist ein Beispiel für den Aufruf der Hardcopy-Routine. Zunächst wird der Cursor in die Home-Position gebracht und mit OPEN1,4,7 die Drucker- und mit OPEN 2,3 die Bildschirmdatei eröffnet. GOSUB 9 startet schließlich die Hardcopy-Zeile. Wie Sie an den beiden FOR-NEXT-Schleifen erkennen können, werden nacheinander alle Spalten und Zeilen des Bildschirms »abgetastet«. Nach dem Druck werden die beiden Dateien wieder mit CLOSE geschlossen. (Karl-Heinz Heß/hm)

```

1 rem *** einzeiler - hardcopy ***
2 :
3 rem bildschirm fuellen
4 print" ";:fori=1to100:print"123456789a
BC";:nexti
5 :
6 rem aufruf hardcopy
7 print" ";:open1,4,7:open2,3:gotosub9:c lo
se2:c close1:end
8 :
9 fori=1to25:forj=1to20:get#2,a$:print"
:get#2,b$:print" "a$b$;:print#1,a$b$;:n
ext:print#1:next:return

```

Ausführliches Listing der einzeiligen Hardcopy-Routine

```

9fori=1to25:forj=1to20:ge#2,a$:?" ";:ge#2,
b$:?" "a$b$;:pr1,a$b$;:ne:pr1:ne:reT

```

So muß Zeile 9 eingegeben werden.