

# Maschinenprogramme auf Diskette speichern

**Reine Maschinenprogramme haben gegenüber einem Basic-Lader zwei entscheidende Vorteile. Sie sind wesentlich kürzer und können direkt, ohne zeitraubendes POKEn, geladen werden. Dieses Programm soll Ihnen helfen, Maschinenprogramme auf Diskette zu speichern.**

Ein einfacher Basic-Lader besteht aus einer READ/POKE-Schleife, in der DATAs in den Speicher geschrieben werden. Jeder Wert, der in dieser Schleife gePOKEt wird, ist ein Byte eines Maschinenprogramms, ein Maschinenbefehl oder ein Teil davon. Anstatt die DATAs in den Speicher zu POKEn, kann ein Basic-Lader dazu benutzt werden, das Maschinenprogramm auf Diskette abzuspeichern. Der Lader wird dazu nur etwas geändert.

Ein Maschinenprogrammfile auf Diskette enthält, vor den Maschinenbefehlen, in den ersten beiden Bytes, die Startadresse des Programms. Der Computer erkennt daran, ab welcher Speicherstelle er das Programm laden soll.

Der Basic-Lader muß also ein Programmfile eröffnen, die Startadresse des Maschinenprogramms ins File schreiben und anschließend alle Werte aus den DATA-Zeilen. Wie das im einzelnen programmiert wird, können Sie am Listing des Beispielprogramms sehen, das Sie ohne weiteres an jedes Ladeprogramm anpassen können.

Der wichtigste Befehl in diesem Beispiel steht in Zeile 50130: OPEN 1,8,1, "Programmname"

Er öffnet eine Datei. Allerdings keine sequentielle, sondern eine Programmdatei. Die Sekundäradresse 1 sagt der Floppy 1541, daß nun ein Programm (NAME.PRG) geschrieben wird. Häufig findet man für diesen Befehl die Syntax OPEN 2,8,2, "Programmname,P, W"

bei der man die Sekundäradresse frei wählen darf. Hier muß jedoch nach dem Namen angegeben werden, daß eine Programmdatei (P) geöffnet werden soll, in die geschrieben wird (Write). Entsprechend kann bei sequentiellen Dateien optional »S,W« bei Schreib- oder »S,R« bei Lesezugriffen angegeben werden.

Nachdem in Zeile 50130 die Programmdatei angelegt wurde, muß die Startadresse des Maschinenprogramms gespeichert werden. Die Startadresse wird dazu in den Zeilen 50170 und 50180 in Low- und High-Byte zerlegt und die zwei Bytes in Zeile 50190 in das Programmfile geschrieben. Jetzt wird das eigentliche Programm gespeichert. Die numerischen Werte müssen dazu mit der CHR\$-Funktion in die entsprechenden Strings übersetzt werden. Um die erforderliche Dichtedichte beim Schreiben des Files zu erreichen, muß jedem String ein Semikolon »;« folgen. Ohne Semikolon ist das Programm später nicht lauffähig. Sind alle Werte, die der Basic-Lader in den Speicher gePOKEt hätte, gespeichert, wird die Datei mit CLOSE geschlossen. Damit ist das Maschinenprogramm auf Diskette gespeichert und kann mit

LOAD "NAME";8,1 geladen und, wenn dies notwendig ist, mit dem entsprechenden SYS-Befehl gestartet werden.

## Laden von Maschinenprogrammen in Basic

Sie können Maschinenroutinen mit Hilfe von Basic-Programmen laden, ohne sie zu zerstören. Dabei ist zu beachten, daß nach dem Laden das Basic-Programm von Neuem startet. Die Basic-Zeile

```
10 LOAD "KEIN ENDE";8,1
```

bewirkt deshalb eine Endlosschleife, die immer wieder das Maschinenprogramm »KEIN ENDE« lädt. Da angelegte Variablen dabei erhalten bleiben, kann mit

```
10 IF A=0 THEN A=1:LOAD "KEIN ENDE";8,1
```

ein mehrmaliges Laden verhindert werden, da beim Neustart A=1 ist. (S. Wengler/hm)

## Listing des Beispielprogramms

```
50000 REM BASIC-LADER
50010 DATA .....
50020 DATA .....
50030 DATA .....
50040 :
50050 REM MASCHINENPROGRAMME AUS
50060 REM DATALADERN
50070 :
50080 REM S KOENNEN WEGGELASSEN WERDEN
50090 :
50100 REM SA = STARTADRESSE
50110 REM EA = ENDADRESSE
50120 :
50130 OPEN 1,8,1, "NAME"
50140 :
50150 REM SA U. EA ZUORDNEN
50160 :
50170 SH = INT(SA/256) :REM HIGH-BYTE
50180 SL = SA-SH*256 :REM LOW-BYTE
50190 PRINT#1,CHR$(SL);CHR$(SH);
50200 :
50210 FOR I=SA TO EA
50220 READ WERT
50230 PRINT#1,CHR$(WERT);
50240 NEXT I
50250 :
50260 CLOSE 1
```