MZ-80B - intern Nr. 7

Das hier beschriebene Verfahren dient dazu, Maschinenprogramme fest mit BASIC-Programmen zu verbinden. Diese Maschinenprogramme werden dabei an den Anfang des BASIC-Programms gelegt und durch Verbiegen von Zeigern vor dem Überschreiben geschützt. Das Verfahren im einzelnen:

Die erste Zeile des BASIC-Programms besteht nur aus dem Befehl REM, direkt gefolgt von <CR>. Die Zeilennummer ist unwichtig, 1 wäre wohl am sinnvollsten. Jetzt folgen entsprechend der Länge des Maschinenprogramms eine Anzahl weiterer REM-Zeilen, wobei auf den REM-Befehl irgendwelche Zeichen folgen müssen. Diese REM-Zeilen werden später durch das Maschinenprogramm ersetzt und unsichtbar gemacht. Jede dieser REM-Zeilen nimmt (7 + Anzahl der dem REM-Befehl folgenden Zeichen) Bytes auf. Berechnen Sie also Ihren Platzbedarf und schreiben Sie eine entsprechende Anzahl REM-Zeilen. Nehmen Sie ruhig ein- oder zwei REM-Zeilen mehr als nötig, dann haben Sie noch Platz für Erweiterungen o. ä. Schreiben Sie nun noch eine weitere REM-Zeile ins Programm, die wie die erste nur aus REM <CR> besteht. Damit ist die Vorbereitung des BASIC-Programms abgeschlossen. Für diese Vorbereitungen ist es belanglos, welcher Interpreter verwendet wird, bei den folgenden Änderungen jedoch nicht mehr. Jetzt geht es nämlich darum, die eingegebenen REM-Zeilen zum Verschwinden zu bringen, ohne daß dieser Bereich wirklich entfernt wird.

Hierzu kurz die Art und Weise, in der der Interpreter BASIC-Zeilen abspeichert: Eine BASIC-Zeile im Speicher beginnt mit zwei Byte, die die Adresse der nächsten BASIC-Zeile angeben. Es folgen zwei Bytes für die Zeilennummer als Hexzahl, der eigentliche Zeileninhalt, wobei alle BASIC-Befehle durch Tokens ersetzt werden, und zum Schluß $0D. Die erste BASIC-Zeile liegt beim Interpreter BASIC SB-5510 bei $511C und bei BASIC SB-6510 bei $675C. In unserem Fall könnte die erste Zeile etwa wie in Bild 1 aussehen. Der in dieser Zeile enthaltene Zeiger auf die nächste Zeile wird jetzt mit dem Monitor geändert, und zwar auf die Adresse der letzten eingegeben REM-Zeile (s. o.), welche sich mit dem Monitor leicht finden läßt, da Sie erstens wie die erste REM-Zeile (Bild 1) aufgebaut ist und zweitens, wenn auch als Hexzahl, an ihrer Zeilennummer identifizierbar ist. Wenn Sie das BASIC-Programm jetzt einmal listen, werden Sie feststellen, daß mit Ausnahme der beiden ganz kurzen REM-Zeilen alle anderen dazwischen verschwunden sind. Dies liegt einfach daran, daß der Interpreter mit der eben durchgeführten Änderung denkt, nach der ersten kurzen REM-Zeile folgt sofort die zweite. Der Rest dazwischen wird nicht gelistet, weil für den Interpreter eine Zeile mit $0D endet, und dieses Zeichen befindet sich ja direkt hinter dem ersten REM.

Sie können nun in den eben dem Interpreterzugriff entzogenen Speicherbereich mit Hilfe des Monitors ihr Maschinenprogramm schreiben. Dieser Bereich liegt logischerweise zwischen dem (Anfang des BASIC-Zeilen-Speicherbereich + 7) und dem (geänderten Zeiger - 1). Darin liegt allerdings auch ein Haken der Angelegenheit: Dieser Speicherbereich liegt also bei jedem Interpreter woanders. Hier gibt es leider keinen Ausweg. Falls es sich bei dem Maschinencode nur um Daten handelt, die nicht auf eine Adresse fixiert sind, so kann man den Anfang dieser Daten auch mit einer kleinen Schleife suchen.

Die hier besprochene Methode funktioniert gut, ist aber nicht ganz einfach. Deshalb ist Sorgfalt bei der Programmierung angebracht, denn durch falsch durchgeführte Änderungen läßt sich das BASIC-Programm zerstören. Ansonsten sind diesem Trick keine Grenzen gesetzt, wie z. B. das Programm Weltzeituhr beweist, in dem der komplette Inhalt der ersten Grafikebene (Weltkarte), immerhin 8 KByte, auf diese Weise mit dem BASIC-Programm verbunden wurde.

Bild 1: Speicherformat einer REM-Zeile

```
511C:     23      51      01      00      80      80      0D
        Nächste BASIC-  Zeilennummer,  Token für REM  Zeilen-
        Zeile beginnt   hier 1                        ende
        bei $5123
```

So würde die Zeile 1 REM<CR> bei BASIC SB-5510 abgespeichert werden. Der Zeiger auf die nächste Zeile (hier liegt er bei $511C/$511D und hat den Inhalt $5123) wird nun verbogen, d. h. er zeigt nicht auf die nächste, sondern übernächste, drittnächste o. a. Zeile. Da eine Zeile unabhängig vom Nächste-Zeile-Zeiger mit $0D endet, ist der übersprungene Bereich für den Interpreter nicht mehr vorhanden, wird aber beim Laden und Abspeichern natürlich mitverwendet. Übrigens darf die erste Zeile nie gelöscht werden, denn dies würde natürlich den Bereich bis zur nächsten Zeile (Nächste-Zeile-Zeiger!) betreffen, und damit wäre auch das Maschinenprogramm weg. Aus dem gleich Grund darf die erste Zeile auch nicht geändert werden. Ab (und einschließlich) der ersten nach diesem Bereich stehenden Zeile ist aber alles erlaubt.

* * *

Beispiel:

```
1   REM [CR]
:   REM          (7 plus Anzahl der dem REM-Befehl folgende Zeichen [Bytes])
:   REM             -"-
:    :
:   REM [CR]
```

sodann im Monitor ($511C) die Zeile-Nummer der letzten REM-Zeile (Ende des Maschinenprogramms) eintragen. [identifizierbar an Zeilennummer (HEX)]

Vor dem Abspeichern auf Kassette:

Monitor SB-1511: $ 0164=02 [01]

- im Monitor ändern: $ 017A = 02 [01]
- dann * S
  - File Name:
  - S-Adr.: $ 511C
  - E-Adr.: Ende des Basic-Programms (steht in $4F0E)
  - J-Adr.: $ 1280
- dann im Monitor $ 017A alten Wert eintragen.