MZ-80B - intern Nr. 3

BASIC SB-5510 und BASIC SB-6510 akzeptieren zwar lange Variablennamen, zur Identifizierung werden aber nur die ersten beiden Zeichen herangezogen. Im Speicher gehen diese beiden Zeichen dem Variableninhalt voran, falls der Variablenname nur ein Zeichen umfaßt, wird als zweites Zeichen ein Leerzeichen verwendet (Bild 1).

Bei Feldern steht im Speicher erst der Name (zwei Zeichen), dann zwei Byte für die Dimension des Feldes in jede Richtung (bei eindimensionalen Feldern ist das zweite Byte Null), und dann folgen ohne Trennungen sämtliche Daten des Feldes (Bild 2).

Die obigen Angaben gelten für numerische Variablen, Stringvariablen werden genauso abgelegt, nur das dem Variableninhalt direkt ein Byte mit der Länge vorangeht und $0D direkt nach dem Variableninhalt als Endzeichen folgt (Bild 3).

Numerische Variable werden nicht als Klartext, sondern als 5-Byte-Binärzahlen gehandhabt. Das höchstwertige Bit des ersten Bytes ist das Vorzeichen der Mantisse, 1 heißt positiv, 0 heißt negativ. Das folgende Bit ist das Vorzeichen des Exponenten. Die restlichen sechs Bit stellen den Exponenten selbst dar. Die restlichen vier Byte sind die Mantisse, diese vier Byte sind in der Reihenfolge Byte 5, Byte 4, Byte 3, Byte 2 zu lesen, das höchstwertige Bit hat die Wertigkeit 2^-1 (2 hoch -1) (Bild 4).

Alle in einem Programm definierten Variablen des gleichen Typs werden im Speicher in der Reihenfolge ihres Auftretens ohne irgendwelche Zwischenräume abgelegt. Die Startadresse des Speicherbereichs für jeden Variablentyp wird in einer Tabelle aufbewahrt. Zwischen den einzelnen Speicherbereichen werden zwei Nullbytes eingefügt (Bild 5).

Bild 1: Speicherung einer numerischen Variablen

```
Die Variable soll Pi heißen und den entsprechenden Wert enthalten:
              50 - 69 - C2 - A1 - DA - 0F - C9
              Name...   5-Byte-Binärzahl......
```

Bild 2: Speicherung eines numerischen Feldes

```
Das eindimensionale Feld soll A heißen und 50 Elemente umfassen:
              41 - 20 - 32 - 00 - XX - XX - ...
              Name...   Dim....   Elemente. ...
```

Bild 3: Speicherung einer Stringvariablen

```
Die Variable soll TEST$ heißen und den String "Sharp" enthalten:
         54 - 45 - 05 - 53 - 68 - 61 - 72 - 70 - 0D
         Name... Länge  Sharp.................
```

Bild 4: Speicherung einer Zahl als 5-Byte-Binärzahl:

Die Zahl soll 455 sein. Sie wird dann gespeichert als

C9 - 00 - 00 - 80 - E3.

C9 entspricht binär 11001001, d. h. das Vorzeichen von Mantisse und Exponent ist positiv, der Exponent ist 2^9 (001001=9), also 512. Die restlichen vier Byte lauten in richtiger Reihenfolge binär

11100011 10000000 00000000 00000000,

d. h. 2^-1+2^-2+2^-3+2^-7+2^-8+2^-9, also 0.888671875. Multipliziert man jetzt Mantisse und Exponent, erhält man die gesuchte Zahl 455.

e = 201 a = 00 b = 07 c = 128 d = 227

Zahl = (2*(a<128)-1) * 2 ↑ (e-160) * (((256*(a+128*(a<128))+b)*256+c)*256+d)

Bild 5: Tabelle für die Speicherbereiche der Variablentypen:

BASIC SB-5510	BASIC SB-6510	
$4F46	$6586	2-dimensionale numerische Felder
$4F48	$6588	1-dimensionale numerische Felder
$4F4A	$658A	numerische Variablen
$4F4C	$658C	2-dimensionale Stringfelder
$4F4E	$658E	1-dimensionale Stringfelder
$4F50	$6590	Stringvariablen

In diesen Speicherplätzen stehen die Anfangsadressen der betreffenden Speicherbereich.

* * *

AUTO-LOAD plus AUTO-RUN aus BASIC heraus

altes Programm evtl. Suchlaufprogramm gem. MZ80-U Intern Nr. 5 sowie

```
POKE$01C0,$0D  [$46]
POKE$01CB,$00  [$CD]
POKE$01CC,$00  [$C5]
POKE$01CD,$00  [$05]
POKE$1AF1,$C5  [$E5]
USR(&01C0)
```

} wird im neuen Programm reinstalliert

neues Programm: nach Ende BASIC-Programm in Maschinensprache

```
NOP / NOP          00 00
LD HL,0E5CDH       21 CD E5
LD (&01A8),HL      22 A8 01
LD A,05H           3E 05
LD (&01AB),A       32 AB 01
plus
POP HL             E1
LD HL,0000H        21 00 00
LD (&4AD0),HL      22 D0 4A
LD HL,86B0H        21 B0 86
LD (&4AD2),HL      22 D2 4A
LD A,00H           3E 00
LD (&4AD4),A       32 D4 4A
LD SP,(stack)      ED 7B C8 49
LD HL,12E0H        21 E0 12
PUSH HL            E5
JP RJUMP           C3 E0 12
plus
im BASIC in irgendeiner Zeile z. B. POKE&1AF1,&E5: POKE $01C0, $46
```

"FILE NAME" unterdrücken:
$01C0 : $0D [$46]

abgespeichert auf Cassette als Object-File, J-Adr. bei den NOP's !