Hex-DATA-Automat

Der Computer programmiert sich selbst — Maschinenprogramme werden automatisch in DATA-Statements mit Prüfsumme umgewandelt.

Ein Maschinenprogramm in ein korrektes Basic-Ladeprogramm umzusetzen, ist sicherlich eine sehr langweilige Programmieraufgabe, außerdem schleichen sich sehr schnell Fehler ein.

Soll diese Umsetzung automatisch erfolgen, müßte sich der Computer — salopp gesagt — selbst programmieren. Dies ist prinzipiell möglich; doch zuvor einige Grundlagen.

Geben Sie hierzu das nebenstehende kleine Testprogramm ein.

```
100 POKE 2,0
110 ZL=PEEK(2) : POKE 2,ZL+1
120 D$=STR$(ZL+1000)
130 D$=D$+"DATA ABCFDEF"
140 PRINT CHR$(147);D$
150 PRINT "RUN 110"
160 :
170 :
180 END
```

```
1 REM ************************************
*******
2 REM *          ++++ PROGRAMM-NAME ++++
      *
3 REM *
      *
4 REM *  STICHWORT.....................
...   *
5 REM *  COPYRIGHT (C) ................
...   *
6 REM *
      *
7 REM ************************************
*******
8 :
9 :
10 DIM H(75) : FOR I=0 TO 9
20 H(48+I)=I : H(65+I)=I+10 : NEXT
30 FOR I=ANFANG TO SCHLUSS: READ A$ : RE
M HIER AKTUELLE WERTE EINSETZEN !
40 H=ASC(LEFT$(A$,1)):L=ASC(RIGHT$(A$,1)
)
50 D=H(H)*16+H(L) : S=S+D : POKE I,D
60 A=A+1:IF A<20 THEN NEXT : A=-1
65 PRINT "ZEILE:";1000+Z;
70 READ V : Z=Z+1 : IF V=S THEN 85
80 PRINT" PRUEFSUMMENFEHLER !";999+Z:STO
P
85 IF A<0 THEN END
90 S=0 : A=0 : PRINT : NEXT : END
95 :
96 :
97 :
98 :
99 :
200 REM ***********************************
*******
210 REM *  HEX-DATA-AUTOMATIK (VC 20 & C
64)   *
220 REM *
      *
230 REM *  DATA-ZEILEN PROGRAMMIEREN
      *
240 REM *       START MIT RUN 200
      *
250 REM *
      *
260 REM ***********************************
*******
270 :
280 :
500 INPUT "START-ADRESSE";A
510 H=INT(A/256):L=A-H*256:POKE 640,H:PO
KE 641,L
520 INPUT "END-ADRESSE  ";E
530 H=INT(E/256):L=E-H*256:POKE 642,H:PO
KE 643,L
540 POKE 2,1 : IF A>E THEN 500
550 :
560 A=PEEK(640)*256+PEEK(641)
570 E=PEEK(642)*256+PEEK(643)
580 :
590 DIM H$(20):FOR I=0 TO 9 : H$(I)=CHR$
(I+48)
600 H$(I+10)=CHR$(I+65)      : NEXT : Z=3
610 :
620 FOR I=1 TO 20 : D=PEEK(A) : S=S+D
630 H=INT(D/16)    : L=D-16*H
640 A$=A$+H$(H)+H$(L)+","
650 IF A=E THEN Z=2 : D=PEEK(2) : GOTO 6
70
660 A=A+1 : NEXT : D=PEEK(2) : POKE 2,D+
1
670 A$=STR$(999+D)+" DATA "+A$
680 A$=A$+STR$(S)
690 H=INT(A/256):L=A-H*256:POKE 640,H:PO
KE 641,L
700 POKE 631,19: POKE 632,13 : POKE 633,
13
710 POKE 198,Z : PRINT CHR$(147);A$
720 PRINT "RUN 560" : END

READY.
```

Listing »Hex-DATA-Automat«

Die Programmzeile 100 setzt die Speicheradresse 2 auf Null. Anschließend wird der Wert dieser Adresse nach ZL geholt und die Adresse um eins erhöht. Der STR$-Befehl wandelt den Wert ZL+1000 in einen String, und die Zeile 130 erweitert den String mit »DATA ABCDEF«. Die CHR$-Anweisung löscht anschließend den Bildschirm und schreibt den String »1000 DATA ABCDEF« links oben auf den Bildschirm. Zuletzt wird in der zweiten Bildschirmzeile der Text »RUN 110« gedruckt.

Falls Sie nach RUN die Taste HOME drücken, steht der Cursor auf der Zeile »1000 DATA ABCDEF«. Drücken Sie nun die RETURN-Taste, dann wird die Zeile 1000 in das Programm aufgenommen. Der Cursor steht jetzt auf dem »RUN 110«. Drücken Sie jetzt erneut RETURN, so startet das Programm wieder, und es folgt der nächste Durchgang mit:

1001 DATA ABCDEF
RUN 110

Da im ersten Durchlauf der Wert in der Speicherzelle 2 um eins erhöht wurde, lautet die nächste Zeilennummer 1001. Nun könnten Sie wieder (in Handarbeit) HOME/RETURN/RETURN eingeben, doch — und jetzt wird's interessant — auch dies kann der Computer durchführen.

```
160 POKE 198,3
170 POKE 631,19:POKE 632,13:POKE 633,13
```

Geben Sie nun RUN ein. Das Programm erweitert sich nun automatisch — ab der Nummer 1000 — um DATA-Zeilen.

Dies ist möglich, da alle Commodore-Computer mit einem Tastaturpuffer arbeiten. In diesem Zwischenspeicher, der beim VC 20 und C 64 ab der Adresse 631 beginnt, kann sich der Computer bis zu neun Tastatureingaben merken. Die Anzahl der Zeichen in dem Puffer steht in der Adresse 198.

In der vorherigen Programmerweiterung wurde in der Zeile 160 der Wert 3 eingePOKEt. Der Computer meint anschließend, es seien drei Tastatureingaben erfolgt. Die POKE-Befehle in der Zeile 170 simulieren die Eingabesequenz HOME, RETURN, RETURN. Nach dem Programmende vergißt der Computer diese untergeschobenen Eingaben keineswegs, sondern führt sie nachträglich aus.

Aus der Zeit des legendären PET 2001 stammt noch die Bezeichnung »selbsterhaltendes Programm«. Na ja, aber so hatte das Kind wenigstens einen Namen.

Leider hat das beschriebene Verfahren den Nachteil, daß der Computer, sobald er sich selbst die Zeile einprogrammiert, die Variablen löscht. Aus diesem Grund müssen Sie wichtige Werte vor dem Programmabbruch durch POKE sichern und beim Neustart mit PEEK zurückholen. In dem vorherigen Testprogramm wurde beispielsweise der Zähler für die Zeilennummer mit der Adresse 2 weiter gegeben.

Das Programm »HEX-DATA-Automatik« arbeitet im Prinzip genau nach dem zuvor beschriebenen Verfahren. Die Umwandlungsroutine wird mit RUN 200 gestartet. Das Programm fragt dann nach der Anfangs- und Endadresse des Maschinenprogramms und wandelt es anschließend in DATA-Zeilen um. Diese haben das folgende Format:

```
1000 DATA 01,02,03,04,05,... ,20, 1234
1001 DATA 11,12,13,14,15,... ,40, 5678
1002 DATA ...
```

Nach jeweils 20 Hexadezimal-Daten folgt immer eine Prüfsumme.

Anschließend müssen Sie die DATA-Routine (200—720) löschen und den Schleifenzähler in Zeile 30 anpassen. Falls das Programm veröffentlicht werden soll, können Sie zusätzlich ein Copyrightstatement hinzufügen.

Für diesen Zweck wurde diese Routine auch ursprünglich erstellt. Die Zeilen 10—90 wandeln die Hex-Zahlen wieder um und schreiben das Maschinenprogramm in den entsprechenden Speicherbereich zurück (Zeile 30 beachten).

Das Programm kann in dieser Form sehr leicht abgetippt werden, da Prüfsummenfehler wie folgt angezeigt werden:

```
ZEILE 1000
ZEILE 1001
ZEILE 1002
ZEILE 1003 PRÜFSUMMENFEHLER !

BREAK IN 80
```

Anschließend müssen nur 20 Daten überprüft werden, so daß auch längere Basic-Lader vergleichsweise schnell und fehlerfrei abgeschrieben werden können.

(Heino Velder/ev)

Dem »Springvogel« auf die Sprünge geholfen

```
59300 A=0:I=827
59310 READ N:IF N<0 THEN 59340
59320 A=A+N:I=I+1
59330 POKE I,N:GOTO 59310
59340 IF I<>953 THEN PRINT"FALSCHE DATA-ZAHL IN 59400 FF. DIFF.:";I-953:STOP
59350 IF A<>15594 THEN PRINT"DATA-ERROR IN 59400 FF. DIFF.:";A-15594:STOP
59360 SYS 828
59400 DATA 120,169, 75,141, 20,  3,169
59410 DATA   3,141, 21,  3, 88, 96, 10
59420 DATA  70,169, 53,133,  1,206, 73
59430 DATA   3,208, 32,169,  2,141, 73
59440 DATA   3,162,  8,189, 71,226, 24
59450 DATA 106,144,  2,  9,128,157, 71
59460 DATA 226,189, 79,226, 24, 42,105
59470 DATA   0,157, 79,226,202,208,231
59480 DATA 172, 88,226,162,  0,189, 89
59490 DATA 226,157, 88,226,232,224,  7
59500 DATA 208,245,140, 95,226,172,103
59510 DATA 226,189, 95,226,157, 96,226
59520 DATA 202,208,247,140, 96,226,206
59530 DATA  74,  3,208, 24,173,137,226
59540 DATA 208,  6,162, 70,169,126,208
59550 DATA   4,162, 15,169,  0,141,137
59560 DATA 226,141,142,226,142, 74,  3
59570 DATA 169, 55,133,  1, 76, 49,234
59580 DATA  -1

READY.
```

Der »Springvogel« sieht sich ja zahlreichen Unannehmlichkeiten ausgesetzt, die recht aktiv versuchen, seinen Sprüngen ein Ende zu setzen. Aber Bänder und Aufzüge transportieren ihn, ohne daß sie sich selbst bewegen, und warum sind eigentlich die Fallen so tödlich?

Wer hier Alternativen haben möchte, braucht nur das Programm um die nebenstehenden Zeilen zu erweitern: Einfach »Springvogel« laden, Ergänzung dazutippen, abspeichern (!), starten.

Die zusätzliche Bewegung auf dem Bildschirm dürfte den Springvogel noch etwas attraktiver machen: Bänder und Aufzüge bewegen sich nun tatsächlich, und die Fallen zwinkern zumindest diskret. Erreicht wird das durch ein kleines Maschinensprachprogramm im Kassettenpuffer, das im Interrupt mitläuft und systematisch die betreffenden Zeichen im neuen Zeichensatz ändert. Die Geschwindigkeit von Bändern und Aufzügen wurde, soweit möglich, der des Vogels angepaßt. Wen dabei das Rasen der Aufzüge nervös macht, der braucht nur den dritten DATA-Wert in Zeile 59430 (32) in 66 und den Kontrollwert in Zeile 59350 (15594) entsprechend in 15628 zu ändern.

(Thomas Schmidt/aa)