

Anhalter — Programmstop auf Tastendruck

Mit dieser kleinen Maschinensprach-Routine können Sie jedes Programm anhalten. Der Computer stoppt so lange, bis Sie eine Taste drücken

Dieses kleine Maschinenprogramm kann Ihnen in vielerlei Hinsicht nützlich sein. Man kann sich beispielsweise folgende Anwendungsgebiete vorstellen: In einem selbstprogrammierten Spiel wird oft eine Pause-Taste benötigt, ähnlich, wie es sie auch in manchen kommerziellen Spielen gibt. Auch das Auflisten von Programmen kann jederzeit angehalten werden und sogar der Datenfluß zur Floppy ist gegen eine Unterbrechung nicht gefeit. Mit dem Programm »Anhalter« steht es Ihnen vollkommen frei, zu entscheiden, an welchen Stellen ein Programm angehalten werden soll.

Nach dem Starten des Programms (siehe Listing 1) werden Sie zunächst gefragt, wohin das Programm verschoben werden soll (bitte Dezimalzahlen eingeben). Da das Programm vollkommen frei verschiebbar ist, brauchen Sie keine weiteren Angaben zu machen. Danach brauchen Sie nur noch die Funktionstaste F1 zu drücken um Ihr Programm zu stoppen. Soll das Programm weiter laufen, so genügt es, eine beliebige Taste zu betätigen.

Falls Sie nicht die F1-Taste, sondern eine andere mit der Anhalte-Funktion belegen möchten, so ist dies möglich. Dazu brauchen Sie nur in die Speicherstelle (Startadresse + 16)

.C000	7B		SEI			;Interrupt abschalten
.C001	A9	0D	LDA	#\$0D		;Interrupt auf C00D setzen
.C006	A9	C0	LDA	#\$C0		;
.C008	8D	15	03	STA	\$0315	;
.C00B	58		CLI			;Interrupt zulassen
.C00C	60		RTS			;Rückkehr zum Basic
.C00D	A5	CB	LDA	\$CB		;hole Wert der letzten gedrückten Taste
.C00F	C9	04	CMP	#\$04		;vergleiche mit Wert für F1-Taste
.C011	F0	03	BEQ	\$C016		;verzweige:ja = eig. Routine, nein=norm. IRQ
.C013	4C	31	EA	JMP	\$EA31	;verlasse Routine zum normalen IRQ
.C016	20	87	EA	JSR	\$EA87	;springe zur normalen Tastaturabfrage
.C019	A5	CB	LDA	\$CB		;hole Wert der letzten gedrückten Taste
.C01B	C9	40	CMP	#\$40		;vergleiche mit \$40 (# 64) = keine Taste
.C01D	F0	F7	BEQ	\$C016		;wenn keine Taste, dann zur Tastaturabf.
.C01F	4C	31	EA	JMP	\$EA31	;wenn Taste gedrückt, dann norm. IRQ

Listing 2. Schnell eingegeben und leicht zu verstehen — der Source-Code von »Anhalter«

den Wert der neuen Taste hineinschreiben. Diesen Wert erhalten Sie, indem Sie folgende Zeile im Direktmodus eingeben, die gewünschte Taste drücken und sich die angezeigte Zahl merken.

FOR I=1 TO 10000 : PRINT PEEK(203) : NEXT

Jetzt können Sie die Abfrage ändern:

POKE Startadresse + 16,X (X=newer Wert)

Listing 2 erläutert die genaue Funktionsweise des Programmes.

(Georg Kramer/aw)

Variable Funktionen

Normalerweise werden Formeln in Programmen fest definiert. Mit dieser kleinen Basic-Erweiterung können Sie Funktionen auch während des Programmablaufs eingeben und ändern.

Um in einem Basic-Programm Formeln zu definieren hatte man bislang nur eine Möglichkeit: die Formel mit dem DEF FN-Befehl in einer Basic-Zeile zu definieren. Vom laufenden Programm aus war das nicht möglich. Dabei wird der Funktionswert zum Beispiel mit A=FNA(X) oder mit PRINT FNB(Y) aufgerufen.

Es geht aber auch anders, nämlich mit diesem kurzen Maschinenprogramm (Listing 1), das mit dem MSE einzugeben ist. Den Quell-Code dazu finden Sie in Listing 2. Das Programm wird mit LOAD "VARFUNKT",8,1 und nachfolgendem NEW geladen, sofern es nicht von einem anderen Programm nachgeladen wird. Das Einbinden in die Interpreter-Routine (=Initialisierung) wird mit SYS(49351) vorgenommen.

```

0 REM *****
1 REM **   +++ ANHALTER   **
2 REM **-----**
3 REM ** WRITTEN BY GEORG KRAMER **
4 REM **   AUF DEM MOORE 15   **
5 REM **   4783 ANROECHTE   **
6 REM **   TEL.: 02947/3511   **
7 REM **   (C) 07.06.1985   **
8 REM *****
9 :
10 PRINT "{CLR,DOWN,3SPACE}++ ANHALTER ++":
    PRINT
15 INPUT "{SPACE,RVSON}STARTADRESSE : {RVDOFF
    ,SPACE}49152 {7LEFT}";S
20 IF S<0 OR S>65535 THEN 10
30 FOR I=0 TO 33:READ A:X=X+A
40 POKE S+I,A:NEXT
50 IF X<>4231 THEN PRINT"DATA-FEHLER !":EN
    D
60 POKE S+7,(S+13)/256
70 POKE S+2,(S+13)-PEEK(S+7)*256
80 SYS(S)
90 PRINT:PRINT" PROGRAMM ANHALTEN => F1"
95 END
100 REM *** MASCHINENPROGRAMM ***
101 DATA 120,169,13,141,20,3,169,192,141,2
    1,3,88,96,165,203,201,4,240,3,76
102 DATA 49,234,32,135,234,165,203,201,64,
    240,247,76,49,234

```

Listing 1. »Anhalter«.

Bitte beachten Sie die Eingabehinweise auf Seite 6

ASSEMBLERLISTING (ASSEMBLER:PROFI-ASS)

NUMERISCHER WERT EINES STRINGS

```

2      : C010      .OPT P2,00
105   : C010      *= #C010
110   : C010      CHRGET = #73
115   : C010      CHRGET = #79
120   : C010      FRMEVL = #AD9E
130   : C010      H1      = 139
140   : C010      H2      = 140
150   : C010      H3      = #C002
160   : C010      H4      = #C003
170   : C010      H5      = #C004
180   : C010      H6      = #C005
;
; NUMERISCHER WERT EINES STRINGS
212   : C010 A5 9D      LDA 157
214   : C012 F0 05      BEQ RUN ; KEIN DIREKT-MODUS
216   : C014 A2 15      LDX #21 ; ILLEGAL DIRECT ERROR
218   : C016 4C 37 A4    JMP #A437
220   : C018 A5 7A      RUN LDA #7A ; PROGRAMMZEIGER SPEICHERN
230   : C010 85 08      STA H1
240   : C01D A5 7B      LDA #7B
250   : C01F 85 0C      STA H2
260   : C021 A0 00      LDY #0
270   : C023 E1 7A      X LDA (#7A),Y ; 1. ZEICHEN NACH DEM BEFEHL EINLADEN
280   : C025 00 9D      BEQ SYNERR ; KEIN ZEICHEN DA
290   : C027 C9 B2      CMP #B2
300   : C029 F0 0E      BEQ PROG ; ZEICHEN EIN '='
310   : C02B E6 7A      INC #7A ; NEIN, DANN NACHSTES ZEICHEN HOLEN
320   : C02D D0 F4      BNE X
330   : C02F C6 7B      INC #7B
340   : C031 4C 23 C0    JMP #C0 ; ZURUECK ZUM 'CHECK'
360   : C034 A2 0E      SYNERR LDX #0E ; "SYNTAX-ERROR"
370   : C036 4C 37 A4    JMP #A437
;
390   : C039 20 73 00    PROG JSR CHRGET ; NACHSTES ZEICHEN IN AKKU
400   : C03C 20 9E AD    JSR FRMEVL ; AUSWERTEN DES TERMES HINTER '='
410   : C03F A5 7A      LDA #7A ; PROGRAMM-U. ZEILENZEIGER SPEICHERN
420   : C041 8D 02 C0    STA H3
430   : C044 A5 7B      LDA #7B
440   : C046 8D 03 C0    STA H4
450   : C049 A5 39      LDA #39
460   : C04B 8D 04 C0    STA H5
470   : C04E A5 3A      LDA #3A
480   : C050 8D 05 C0    STA H6
490   : C053 A0 00      LDY #0
500   : C055 B1 64      LDA (#64),Y ; #64/#65 SIND ZEIGER AUF DEN STRINGD
;
; DESCRIPTOR
510   : C057 85 61      STA #61 ; DESCRIPTOR NACH #61-#63 HOLEN
512   : C059 D0 05      BNE R1
514   : C05B A2 0E      LDX #14
516   : C05D 4C 37 A4    JMP #A437 ; FEHLER, WEIL STRING LEER
520   : C060 C8      RI INV
530   : C061 B1 64      LDA (#64),Y
540   : C063 85 62      STA #62
550   : C065 C8      INV
560   : C066 B1 64      LDA (#64),Y
570   : C068 85 63      STA #63
580   : C06A A0 00      LDY #0
590   : C06C B1 6B      Z LDA (#1B),Y ; 1. TEIL DES TERMS IN EINGABEPUFFER
600   : C06E 99 00 02    STA #0200,Y
610   : C071 C8      INV
620   : C072 C9 B2      CMP #B2 ; BIS ZU '='
630   : C074 D0 F6      BNE Z
640   : C076 99 00 04    LDA #0400,Y ; '=' IN STRINGCODE WANDELN
650   : C078 99 FF 01    STA #01FF,Y
660   : C07B A2 00      LDX #0
670   : C07D A1 62      LY LDA (#62),X ; 2. TEIL DES TERMS UEBERTRAGEN
680   : C07F 99 00 02    STA #0200,Y
690   : C082 E6 62      INC #62
700   : C083 E6 62      INC #62
710   : C085 D0 02      BNE LZ
720   : C087 E6 63      INC #63
730   : C089 C6 61      LZ DEC #61
740   : C08B D0 F0      BNE LY
750   : C08D 99 00 02    STA #0200,Y ; LETZTES ZEICHEN = 0
760   : C091 A9 FF      LDA #FF ; DIREKTMODUS SIMULIEREN
770   : C093 85 7A      STA #7A
780   : C095 A9 01      LDA #01
790   : C097 85 7B      STA #7B
800   : C099 A9 80      LDA #80
810   : C09B 20 18 FE    JSR #FE18
820   : C09E 20 73 00    JSR CHRGET ; 1. ZEICHEN AUS EINGABEPUFFER IN AKKU
830   : C0A1 A2 FF      LDX #FF
840   : C0A3 86 36      STA #36
850   : C0A5 20 79 A5    JSR #A579 ; TERM IN INTERPRETERCODE WANDELN
860   : C0A8 20 73 00    JSR CHRGET
870   : C0AB 20 ED A7    JSR #A7ED ; BEFEHL AUSFUEHREN
880   : C0AE AD 02 C0    LDA H3 ; ZEIGER ZURUECKHOLEN
890   : C0B1 85 7A      STA #7A
900   : C0B3 83 C0      LDA H4
910   : C0B5 85 7B      STA #7B
920   : C0B8 AD 04 C0    LDA H5
930   : C0BB 85 39      STA #39
940   : C0BD AD 05 C0    LDA H6
950   : C0C0 85 3A      STA #3A
960   : C0C2 83 C0      LDA #0 ; RUN-MODUS WIEDERHERSTELLEN
970   : C0C4 85 9D      STA 157
980   : C0C6 60      RTS
990   : C0C7 A9 D2      LDA #CAUS ; ZUR INTERPRETERSCHLEIFE
1000  : C0C9 8D 08 03    STA #0308 ; VEKTOR NEU BESTIMMEN
1010  : C0CB A9 C0      LDA #AUS
1020  : C0CD 8D 09 03    STA #0903
1030  : C0CF 8D 09 03    STA #0903
1040  : C0D1 60      RTS
1050  : C0D2 20 73 00    AUS JSR CHRGET ; BEFEHL "? "
1060  : C0D5 C9 AE      CMP #FAE
1070  : C0D7 F0 0A      BEQ S
1072  : C0D9 C9 0B      CMP #FB
1074  : C0DB F0 0C      BEQ S ; IF-STRUKTUR
1080  : C0DD 20 79 00    ST JSR CHRGET
1090  : C0E0 4C E7 A7    JMP #A7E7 ; NEIN
1100  : C0E3 20 10 C0    S JSR WERT ; NEUEN BEFEHL AUSFUEHREN
1110  : C0E5 4C AE A7    JMP #AAE ; ZUR INTERPRETERSCHLEIFE
1120  : C0E8 20 02 C0    T JSR IF ; IF-BEFEHL AUSFUEHREN
1130  : C0EC 4C AE A7    JMP #AAE ; ZUR INTERPRETERSCHLEIFE
1140  : C0EF 4C FB A8    R JMP #AFB8
1200  : C0F2 20 73 00    IF JSR CHRGET
1200  : C0F5 20 9E AD    JSR #AD9E
1210  : C0F9 20 79 00    JSR CHRGET
1220  : C0FB C9 89      CMP #89
1230  : C0FD F0 05      BEQ U
1240  : C0FF A9 A7      LDA #AA7
1250  : C101 20 FF AE    JSR #FAEF
1260  : C104 A5 61      U LDA #61
1270  : C106 D0 05      BNE V
1280  : C108 20 09 A9    JSR #A909
1290  : C10B F0 E2      BEQ R
1300  : C10D 20 79 00    V JSR CHRGET
1310  : C110 8D 03      BCS W
1320  : C112 4C A0 A0    JMP #AA0 ; NEUEN BEFEHL AUSFUEHREN
1330  : C115 C9 AE      CMP #FAE
1340  : C117 F0 06      BEQ WE
1350  : C119 C9 8B      CMP #8B
1360  : C11B F0 05      BEQ IF2
1370  : C11D D0 BE      BNE ST
1380  : C11F 4C 10 C0    WE JMP WERT ; JMP, WEIL NOCH INNERHALB DER IF-STRUKTUR
1390  : C122 4C F2 C0    IF2 JMP IF

```

Listing 2.
Assembler-Quell-Listing
zu »Variable Funktionen«.

programm : varfunkt c010 c125

```

c010 : a5 9d f0 05 a2 15 4c 37 d3
c018 : a4 a5 7a 85 8b a5 7b 85 bd
c020 : 8c a0 00 b1 7a f0 0d c9 2a
c02B : b2 f0 0e e6 7a d0 f4 e6 82
c030 : 7b 4c 23 c0 a2 0b 4c 37 d4
c03B : a4 20 73 00 20 9e ad a5 c2
c040 : 7a 8d 02 c0 a5 7b 8d 03 8c
c04B : c0 a5 39 8d 04 c0 a5 3a 2c
c050 : 8d 05 c0 a0 00 b1 64 85 ce
c05B : 61 d0 05 a2 0e 4c 37 a4 20
c060 : c8 b1 64 85 62 c8 b1 64 c7
c06B : 85 63 a0 00 b1 8b 99 00 a5
c070 : 02 c8 c9 b2 d0 f6 a9 3d 85
c07B : 99 ff 01 a2 00 a1 62 99 6f
c080 : 00 02 c8 e6 62 d0 02 e6 13
c08B : 63 c6 61 d0 f0 8a 99 00 8a
c090 : 02 a9 ff 85 7a a9 01 85 1b
c09B : 7b a9 80 20 18 fe 20 73 ed
c0a0 : 00 a2 ff 86 3a 20 79 a5 9b
c0aB : 20 73 00 20 ed a7 ad 02 5d
c0b0 : c0 85 7a ad 03 c0 85 7b ca
c0bB : ad 04 c0 85 39 ad 05 c0 df
c0c0 : 85 3a a9 00 85 9d 60 a9 e7
c0cB : d2 8d 08 03 a9 c0 8d 09 ac
c0d0 : 03 60 20 73 00 c9 ae f0 65
c0dB : 0a c9 8b f0 0c 20 79 00 6f
c0e0 : 4c e7 a7 20 10 c0 4c ae a3
c0eB : a7 20 f2 c0 4c ae a7 4c e5
c0f0 : fb a8 20 73 00 20 9e ad 8d
c0fB : 20 79 00 c9 89 f0 05 a9 96
c100 : a7 20 ff ae a5 61 d0 05 40
c10B : 20 09 a9 f0 e2 20 79 00 4a
c110 : b0 03 4c a0 a8 c9 ae f0 de
c11B : 06 c9 8b f0 05 d0 be 4c 6e
c120 : 10 c0 4c f2 c0 ff 00 ff 0e

```

Listing 1. »Variable Funktionen«.
Beachten Sie die Eingabeinweise auf Seite 8.

Aufruf aus dem Programm

Die Übergabe der Variablen erfolgt in der denkbar einfachsten Art: die Formel steht in einer String-Variablen wie zum Beispiel X\$. Da X\$ ein normaler String ist, kann man die Formel natürlich auch über den INPUT-Befehl oder eine GET-Schleife eingeben, wobei man bei INPUT nicht über 80 Zeichen kommt und GET eine programmbedingte Längenbegrenzung von 127 Zeichen hat. Die Syntax ist dabei die des normalen Basic mit sämtlichen Basic-Funktionen. Das heißt, daß man zum Beispiel die Formel »5 SIN X« als »5 * SIN(X)« eingibt. Dabei bedeutet X wiederum eine Variable. Natürlich kann die Formel beliebig viele Variablen, beziehungsweise beliebig verschachtelte Basic-Funktionen enthalten.

Die Syntax des neuen Befehls ist

↑ Y=B\$

Hier wird der Variablen Y der Wert des berechneten Strings X\$ zugewiesen.

Hierzu noch ein Beispielprogramm, das das Programm »Varfunkt« auf Diskette erwartet:

```

10 IF A=0 THEN A=1 : PRINT "{home}": LOAD "VARFUNKT",8,1
20 SYS 49351
30 INPUT "{down} FUNKTION UEBER X ":X$
40 INPUT " WERT FUER X ":X
50 ↑ Y=X$
60 PRINT " FUNKTIONSWERT = ",Y
70 GOTO 30

```

In Zeile 10 wird beim Start die Maschinenroutine »Varfunkt« nachgeladen. Die IF-Abfrage ist nötig, da der Interpreter nach einem LOAD-Befehl das Programm erneut startet.

Der SYS-Aufruf in 20 bindet das nachgeladene Programm in das Betriebssystem ein.

In den Zeilen 30 und 40 werden Funktion und Wert für die Funktion eingegeben.

Zeile 50 schließlich übergibt den berechneten Wert aus X\$ an die Variable Y, die in 60 ausgegeben wird.

Zeile 70 schließt das Beispielprogramm ab, indem wieder in Zeile 30 zur Funktionseingabe gesprungen wird.

(Jens Weber/og)