

KLEINCOMPUTER 

KC 85/2

M 006

BASIC

EINLEITUNG

Der Modul M006 BASIC ist ein 16 KByte Festwertspeicher (ROM: Read Only Memory - Nur Lesespeicher), in dem das erweiterte Betriebssystem des KC85/2 sowie der BASIC-Interpreter mit Erweiterungen gegenüber der Kassetten-Version abgelegt ist. Diese Beschreibung nennt die Besonderheiten des BASIC-Moduls, die sich aus der Modulstruktur ergeben, sowie die BASIC-Erweiterungen.

INBETRIEBNAHME DES BASIC-MODULS

Stecken und Entfernen des Moduls

Der BASIC-Modul wird zunächst mit dem Gerät verbunden.
Das Stecken und Entfernen des Moduls darf nur im ausgeschalteten Zustand des Systems erfolgen.

Stecken des Moduls

Hieraus ergeben sich folgende Handgriffe für das Stecken des Moduls

- a) Den Computer ausschalten.
- b) Die Kappe des Modulschachtes abnehmen.
- c) Den Modul bis zum fühlbaren Einrasten einschieben. Der M006 ist denn richtig gesteckt, wenn sein hervorstehender Rand unmittelbar an der Gerätewand anliegt.
- d) Nun kann der Computer eingeschaltet werden.

Bevor die weitere Inbetriebnahme erläutert wird, soll an dieser Stelle gleich das Entfernen des Moduls aus dem System beschrieben werden. Zum Entfernen des Moduls sind folgende Schritte notwendig:

Entfernen des Moduls

- a) Den Computer ausschalten.
- b) Den linken und den rechten Zeigefinger unter den Modulkopf legen und mit dem Daumen die seitlich am Modul befindlichen Hebel gleichzeitig nach unten drücken. Dabei rastet der Modul aus und wird etwa einen Zentimeter aus dem Gerät herausgeschoben. Nun kann der Modul leicht aus dem Schacht gezogen werden.
- c) Die Kappe auf die Schachtöffnung stecken.

Zuweisung des BASIC-Moduls

Der BASIC-Modul ist nun mit dem System kontaktiert. Noch dem Einschalten des Computers wird mit der Anweisung JUMP mm (mm – Modulsteckplatzadresse) und dem Drücken der ENTER-Taste der BASIC-Modul zugewiesen und das neue Betriebssystem des Moduls gestartet. Für den Parameter mm ist die Steckplatzadresse des Moduls im Gerätesystem anzugeben. Dabei ist die erste Stelle von mm die Nummer des Aufsatzes und die zweite Stelle die Steckplatzadresse im entsprechenden Aufsatz. Im Grundgerät ist die erste Stelle von mm 0: für die zweite Stelle sind nur 8 (rechter Schacht) und C (linker Schacht) möglich.

Ein Beispiel:

Der BASIC-Modul ist im rechten Schacht des Grundgerätes eingesteckt. So lautet die Anweisung:

JUMP 08 (Enter)

Mit dem Drücken der Enter-Taste erfolgt der Sprung in das Betriebssystem des BASIC-Moduls. Auf dem Bildschirm erscheint die Ausschrift „CAOS 3.1“ mit dem erweiterten Menü und einem neuen Promptzeichen %. Bei diesem Vorgang wird der Festwertspeicher (ROM) des Grundgerätes abgeschaltet. Man erkennt es am Erlöschen der ROM-Kontrollanzeige des Grundgerätes. Die Leuchtdiode des BASIC-Moduls wird eingeschaltet. Mit Hilfe der im Menü enthaltenen Anweisung BASIC läßt sich der BASIC-Interpreter bereits jetzt wie gewohnt starten. Der Aufruf über REBASIC darf nur noch vorherigem Aufruf von BASIC und der Betätigung der RESET-Taste bzw. der BASIC-Anweisung BYE erfolgen.

BESONDERHEITEN DES BASIC-MODULS

Besonderheiten der Modulstruktur

- Anfangsadressen von Maschinenprogrammen

Die Anfangsadressen von Maschinenprogrammen, die mit der Systemanweisung LOAD bzw. mit der BASIC Anweisung BLOAD eingelesen werden sollen, müssen in den letzten zwei Zifferstellen auf 0 gerundet sein, (z. B. 100H, 300H, 500H) Abweichungen hiervon führen zu Fehlern.

- Start von BASIC nach RESET

Verläßt man den BASIC-Interpreter durch Drücken der RESET-Taste, so muß der BASIC-Modul erneut mit JUMP mm (Enter) zugewiesen werden, obwohl die Anweisung BASIC und REBASIC im Menü vorhanden sind. Erst danach kann der BASIC-Interpreter mit den Anweisungen BASIC oder REBASIC gestartet werden,

Besonderheiten des Betriebssystems

Folgende Veränderungen gegenüber dem Kleincomputer KC8512 sind in das Betriebssystem des Moduls eingegangen.

Tastencode

Taste Erstbelegung / Code Zweitbelegung / Code

INS	INS	1AH	CLICK	14H
-----	-----	-----	-------	-----

Speicherplatzbelegung

Funktionstasten	- Speicher von 3900H bis B99BH
Fenstervektor	- Speicher von 699CH bis B9FFH

Menüerweiterungen

KEY

KEY n

KEY Funktionstastennummer

Das Menüwort KEY dient der Belegung der Funktionstasten F1 bis F6 und der Zweitbelegung F7 bis FC. Bei Ausführung der KEY-Anweisung im angegebenen Format befindet sich der Computer im Eingabemodus. Nun wird die einzugebende Tastenfolge für die benannte Funktionstaste definiert. In diesem Eingabemodus können neben den üblichen alphanumerischen- und Sonderzeichen auch Zeichen für die auf der Tastatur verfügbaren Portier- und Steuerfunktionen durch Druck auf die jeweilige Steuer- oder Editiertaste eingegeben werden. Die einzige Korrekturmöglichkeit besteht in diesem Modus durch Drücken der CLR-Taste. Dabei rückt der Cursor eine Position nach links, das davorliegende Zeichen wird gelöscht und kann neu eingegeben werden. Der Eingabemodus wird mit der STOP-Taste beendet. Mit dem Drücken der betreffenden Funktionstaste (F1 bis F6) wird nun stets die im KEY Eingabemodus festgelegte Tastenfolge ausgegeben bzw. die darin als Steuer oder Editierzeichen enthaltenen Funktionen ausgeführt.

Beispiel: Eingabe

KEY 2 KEYFunktionstastennummer (ENTER-Taste als Abschluß)
REBASIC als Tastenfolge für Tastenbelegung STOP-Taste als Abschluß

KEYLIST

KEYLIST erfordert keinen Parameter. Durch Aufrufen dieses Menüwortes werden alle Funktionstastenbelegungen aufgelistet.

DISPLAY

DISPLAY aaaa eeee (ss)

Die Anweisung DISPLAY bewirkt die Ausgabe des Speicherinhaltes von Adresse aaaa bis Adresse eeee
(hexadezimale Angabe der Adressen)

Beispiel:

Eingabe: DISPLAY BA00 BA0F 02

Anzeige Bildschirm:

<u>Speicheradresse</u>	<u>Hexadezimale Angabe</u>	<u>ASCII Darstellung</u>
BA00	41 42 43 44 45 46 47 48	ABCDEFGH
BA08	49 4A 4B 4C 4D 4E 4F 50	IJKLMNOP

Es werden jeweils 8 Bytes in hexadezimaler Darstellung und als ASCII-Zeichen nebeneinander aufgelistet. Die Anzahl der gleichzeitig auf dem Bildschirm angezeigten Zeilen wird durch den Parameter es festgelegt. Wird der Parameter ins nicht eingegeben, sind acht Zeilen für die Bildschirm anzeige vereinbart. Ist die Ausgabe der letzten Speicherzelle eeee noch acht oder ss Zeilen noch nicht erreicht, kann die Anzeige durch Drücken einer beliebigen Taste, mit Ausnahme der BRK- und STOP-Taste, fortgesetzt werden. Die BRK-Taste unterbricht das Auflisten des Speicherbereiches. Durch die STOP-Taste wird das Unterprogramm MODIFY eingeschaltet, wobei der Speicherinhalt nur in der hexadezimalen Darstellung geändert werden kann. Der ASCII-Teil wird dagegen nicht aktualisiert. Für MODIFY gelten alle Zeichenvereinbarungen wie im Bedienhandbuch D001 beschrieben.

Erweiterungen der Tastenfunktion

Mit der Tastenkombination SHIFT und INS kann ein Tasten-Click ein- bzw. ausgeschaltet werden. Die Dauerumschaltung der Tastatur SHIFTLCK wird durch eine zweifarbige Darstellung des Cursors angezeigt.

Erweiterungen des BASIC-Interpreters

Anweisung: RANDOMIZE

Format. RANDOMIZE

Bemerkung: Mit Hilfe der RANDOMIZE-Anweisung wird der Zufallsgenerator mit einem zufälligen Anfangswert gestartet, da sonst beim Kaltstart des BASIC-Interpreters die Zufallszahlenfolge immer die gleiche wäre.

Beispiel:

```
10 RANDOMIZE
20 FOR I=1 TO 5
30 PRINT INT (36 * RND (1)) + 1
40 NEXT I
```

Anweisung: KEY n

Format KEY Funktionstastennummer

Bemerkung: Durch die Ausführung der KEY-Anweisung im angegebenen Format (Funktionstastennummer von 1 bis 12) wird der Computer in den Eingabemodus versetzt. Bei Ausführung der KEY-Anweisung im angegebenen Format befindet sich der Computer im Eingabemodus. Nur wird die einzugebende Tastenfolge für die benannte Funktionstaste definiert. In diesem Eingabemodus können neben den üblichen alphanumerischen und Sonderzeichen auch Zeichen für die auf der Tastatur verfügbaren Editier- und Stammfunktionen durch Druck auf die jeweilige Steuer oder Editiertaste eingegeben werden. Die einzige Korrekturmöglichkeit besteht in diesem Modus durch Drücken der CLR Taste. Dabei rückt der Cursor eine Position nach links, das davor liegende Zeichen wird gelöscht und kann neu eingegeben werden. Der Eingabemodus wird mit der STOP-Taste beendet. Mit dem Drücken der betreffenden Funktionstaste (F1 bis F6) wird nun stets die im KEY-Eingabemodus festgelegte Tastenfolge ausgegeben bzw. die darin als Steuer oder Editierzeichen enthaltenen Funktionen ausgeführt.

Beispiel: KEY 2
WINDOW: COLOR 7.1 : CLS: LIST (Enter)
OK

Anweisung: KEYLIST

Format: KEYLIST

Bemerkung: Durch die Anweisung wird die Funktionstastenbelegung aufgelistet.

Beispiel:

Eingabe: KEYLIST Entertaste

Anzeige:

```
F1: RUN (Enter)
F2: WINDOW : COLOR 7,2 CLS LIST (Enter)
F3: PRINT
F4: EDIT
F5: CLEAR
F6: LINE
F7: CIRCLE
F8: KEY
F9: KEYLIST
FA:
FB:
FC:
OK
```

Anweisung: LINE

Format: LINE XA, YA, XE, YE (,F)

Bemerkung: Die Anweisung zeichnet eine Linie auf den Bildschirm, die im Punkt, mit den Koordinaten XA, YA, beginnt und im Punkt, mit den Koordinaten XE, YE, endet.
Die Farbe der Geraden wird durch den Vordergrundfarbparameter F ($0 < F < 31$) festgelegt. Entfällt der Parameter F, so er scheint die Linie in der zuletzt gewählten Grafikfarbe

Beispiel: LINE 10, 100, 300, 100, 26
OK

Anweisung: CIRCLE

Format: CIRCLE XM, YM, R (.F)

Bemerkung: Diese Anweisung zeichnet auf den Bildschirm einen Kreis, unter Angabe der Mittelpunktkoordinaten XM, YM und des Radius R. Die Farbe der Kreislinie wird durch den Vordergrundfarbparameter F ($0 < F < 31$) festgelegt. Entfällt der Parameter F, so erscheint die Kreislinie in der zuletzt gewählten Grafikfarbe.

Beispiel:
10 CLS
20 FOR R 1 TO 100 STEP 10
30 CIRCLE 159, 127, R, 16 * RND(1)
40 NEXT

Funktion: CSRLIN

Format: CSRLIN (n)

Bemerkung: CSRLIN (n) liefert als Funktionswert die Nummer der Zeile, in der sich der Cursor gerade befindet.
n=0 - Zeilennummer bezüglich des maximalen Fensters
n>0 - Zeilennummer bezüglich des aktuellen Fensters
n<0 - FC-ERROR

Beispiel:
10 WINDOW 0, 31,0,39: CLS
15 PRINTCHR\$(17)
20 RANDOMIZE
30 FOR I =1 TO 100
40 INK (16 * RND (1))
50 PRINTAT (32 * RND (1), 38 * RND (1)); CSRLIN (0)
60 NEXT
70 INK7 : PRINTCHR\$(18)

Anweisung: SWITCH

Format: SWITCH mm, kk

Bemerkung: Die BASIC-Anweisung SWITCH ermöglicht das Schalten von Modulen auch innerhalb eines Programms.

In folgenden Angaben unterscheidet sich die SWITCH -BASICAnweisung von der SWITCH-System Anweisung:

	<u>Systemanweisung</u>	<u>BASIC Anweisung</u>
SWITCH -	(%) Promptzeichen des Systems - Eingabe des Parameters M - Status lesen und schalten - Durch Leerzeichen getrennte Eingaben - Modulsteckplatzadresse Angabe hexadezimal	- Promptzeichen von BASIC - Eingabepflicht der Parameter m und k - Nur schalten - Parameter durch Komma getrennt - Modulsteckplatzadresse Angabe dezimal

Beispiel: Schalten des V24-Moduls

SWITCH C 1
0C EE 01

SWITCH 12,1

Drucken des Namen Emil

10 SWITCH 12,1
20 PRINT #2 "EMIL"
30 SWITCH 12,0

zuweisen V24-Modul
Druckanweisung
abschalten V24-Modul

Anweisung: OPEN

Format: OPEN r # n Name

Bemerkung: OPEN eröffnet eine Kanaloperation mit Namensübergabe auf dem Kanal n für dateifähige Geräte (z. B. Floppy) oder Geräte mit Namensverwaltung. Der Parameter r legt fest, ob es sich um eine Aus oder Eingabe handelt.
r= I Eingabe
r= O Ausgabe

Beispiel: siehe Anweisung PRINT INPUT 11

Anweisung: CLOSE Format CLOSE r 1 n

Bemerkung CLOSE schließt den Kanal n (nach Dateneingabe=r oder nach Datenausgabe r = 0) CLOSE führt der BASIC-Interpreter automatisch bei Programmende oder -abbruch (Taste BRK oder Error) aus.

Beispiel: siehe Anweisung PRINT #, INPUT#

Anweisung: PRINT #, INPUT #

PRINT # n Daten INPUT # n Daten

Bemerkung: PRINT# ermöglicht die Ausgabe von Daten auf einen durch n definierten Kanal.
INPUT # ermöglicht die Eingabe der mit PRINT # ausgegebenen Daten von einem durch n wählbaren Peripheriegerät.

Beide Anweisungen sind nur in Verbindung mit OPEN und CLOSE (Übergabe eines Dateinamens) zu verwenden. Ein Lesen der mit PRINT # ausgegebenen Daten ist nur durch die Anweisung OPEN und CLOSE möglich. Sollen Daten jedoch nur ausgegeben werden (z. B. auf Drucker), können sie auch mit der Anweisung PRINT # ohne OPEN und CLOSE ausgegeben werden.

```

Beispiel
10 CLS
20 DIM A$(4)
30 FOR I=0 TO 4 : INPUT A$(I) : NEXT
40 PRINT "RECORDER AUF AUFNAHME STELLEN": PAUSE20
50 OPEN#1 "KARTEI"
60 FOR I=0 TO 4 : PRINT #1A$(I) NEXT
70 CLOSE # 1
80 END
100 CLS : DIM A$(4)
110 PRINT „RECORDER AUF WIEDERGABE STELLEN“: PAUSE20
120 OPEN # 1 "KARTEI"
130 FOR I=0 TO 4 : INPUT # 1 A$(I): NEXT
140 CLOSE # 1
150 END

```

Anweisung: LIST #, LOAD #

Format: LIST # n „Programmname“ , LOAD n "Programmname"

Bemerkung: LIST ermöglicht die Ausgabe eines Programmlisting (ASCII-Zeichen für ASCII-Zeichen) auf ein durch n wählbares Peripheriegerät. Mit LOAD # können die mit LIST# ausgegebenen Programme wieder eingelesen werden Durch n wird wiederum das Peripheriegerät festgelegt.

Beispiele: LIST # 2 "KALKU" (Ausgabe des Programmlistings KALKU auf Drucker)
LOAD # 1 "TEST" (Einlesen des mit LIST # 1 "TEST" auf Magnetband ausgegebenen Programmlistings)

Peripheriecodierung n für die Anweisungen OPEN, CLOSE, PRINT #, INPUT, LIST # und LOAD #

<u>n</u>	<u>Peripherie</u>
0	Bildschirm und Tastatur (Angabe # 0 sowie OPEN und CLOSE können entfallen)
1	Kassettenrecorder
2 und 3	frei für weitere Anwendungen (Drucker, Plotter, Floppy, Lochstreifen-einheit usw). Einbinden der Treiberrountinen der Systembeschreibung entnehmen.

Funktion: PTEST

Format: PTEST (X)

Bemerkung: Die Funktion testet, ob ein Bildpunkt gesetzt ist, d. h. ob die eingestellte Vordergrundfarbe auf dem Bildschirm er scheint, Die X-Koordinate des zu testenden Punktes wird als Parameter in Klammern hinter die Funktion geschrieben. Als Y-Koordinate benutzt der Computer die zuletzt genannte Y-Koordinate einer Grafikanweisung. Ist der Bildpunkt gesetzt, liefert die Funktion als Ergebnis den Wert 1; anderenfalls den Wert 0.

Beispiel:
40 PRESET 0.56
50 A = PTEST (70)
60 PRINT A

Funktion: VGET\$
Format: String - VGET\$
Bemerkung: Die Funktion VGET\$ liefert den Inhalt der Cursorposition als String.
(1 Zeichen)

Beispiel:
10 CLS
20 PRINT "TEXT"
30 LOCATE 0,1 : AS - VGET\$
40 PRINT AT (10.10); AS
50 LOCATE 1,0

Funktion: INSTR
Format: INSTR (String. String)
Bemerkung: Die Funktion INSTR (AS, B\$ ermittelt die Position ob wo die Zeichenkette AS in der Zeichenkette B\$ enthalten ist

Beispiel:
10 A\$ = "PFERDE": B\$ - BLUMENTOPFERDE
20 PRINT INSTR (A\$, B\$)

Besonderheiten beim Aufruf von Maschinenunterprogrammen

Der BASIC-Interpreter auf dem Modul M006 schaltet bei jedem Zugriff auf seinem Arbeitsspeicher den IRM ab (IRM-LED flackert bei Abarbeitung längerer Programme). Hieraus ergeben sich folgende Konsequenzen beim Aufruf von Maschinenunterprogrammen vom BASIC aus:

- Liegt ein Maschinenunterprogramm im IRM, so muß vor seinem Aufruf der IRM zugeschaltet werden, Dies kann durch ein weiteres Maschinenunterprogramm geschehen das aber im Adreßbereich 0000H bis 7FFFH liegen muß
- Soll von einem Maschinenunterprogramm aus, das im BASIC Arbeitsspeicher liegt, auf den IRM zugegriffen werden, z. B, zur Realisierung von Bildschirmausschriften (Text und/oder Grafik), muß der IRM ebenfalls zugeschaltet werden.
- Vor Rückkehr ins BASIC muß wieder der IRM-Schaltzustand hergestellt werden der vor dem Zugriff auf den IRM herrschte.

Die folgende Befehlsfolge realisiert das Ein- bzw. Anschalten des IRM mit Hilfe spezieller Unterprogramme des Betriebssystems, die außerdem den STACK des BASIC-Interpreters verlegen und den Inhalt des BC-Registers zerstören.

CALL 0F018H CD18F0 Einschalten des IRM

.
.
.

Zugriff zum IRM ist möglich z. B. zum
Maschinenunterprogramm dort bzw. zur Realisierung
von Bildschirmausschriften

.
.

CALL 0F01BH CD1BF0 Abschalten des IRM
RET C9 Rückkehr ins BASIC

Unter Beachtung dieser Gegebenheiten kann die BASIC Anweisung CALL wie beim Bandinterpreter angewendet werden. Bei der Funktion USR (X) sind weiterhin die gegenüber dem Bandinterpreter veränderten Adressen zu berücksichtigen:

Funktion: USR

Format: USR (X)

Bemerkung: Die Funktion ruft ein Maschinenprogramm mit dem Argument X auf. Die Anfangsadresse des Maschinenprogramms ist auf die Speicherplätze 304H und 305H (772 und 773 dezimal) z. B. mit der Anweisung DOKE zu speichern. Die Parameterübergabe vom BASIC-Programm zum Maschinenprogramm wird mit dem Unterprogramm PARAM (Adresse C96FH und die Parameterübergabe vom Maschinenprogramm mit dem Unterprogramm FUWER (Adresse D0B1H des BASIC-Interpreters realisiert, Der Parameter von BASIC wird in das Register DE übernommen und in den Registern A (H-Teil) und B (L-Teil) zum BASIC übergeben.

Beispiel:

Hier kann das gleiche Beispiel ausprobiert werden, das auf Seite 21-7 der Beschreibung zur Kassette C0111 angegeben ist, wenn die entsprechenden Adressen geändert werden.

Beispiel:

Das folgende Beispiel ist ein Rahmenprogramm mit dem es möglich ist, ein Maschinenunterprogramm im IRM aufzurufen. Es wird auf die freien Speicherplätze unterhalb des BASIC-Arbeitsspeichers geregelt, Das Maschinenunterprogramm beginnt auf Adresse 0BE00H im IRM.

Adresse	MC	Anweisung	Bemerkung
0000	CD6FC9	CALL PARAM ;	UP zur Parameterübergabe vom BASIC an das Maschinenprogramm
0003	CD18F0	CALL 0F018H	Zuschalten des IRM
0006	CD00BE	CALL 0BE00H	Ruf des Anwenders Maschinenunterprogramms
0009	CD1BF0	CALL 0F01 BH	Abschalten des IRM
000C	7A	LD A,D	in den Registern A und B
000D	43	LD, B,E	erfolgt die Parameter Rückgabe
000E	CDB1D0	CALL FUWER	UP zur Funktionswertzuweisung an die BASIC Variable
0011	C9	RET	Rückkehr ins BASIC

Das Maschinenunterprogramm, das auf der Adresse 0BE00H beginnt, übernimmt den Parameter X der USR (X) Funktion im DE Register Im obigen Beispiel wird der berechnete Funktionswert auch im DE-Register an das Rahmenprogramm zurückgegeben, Hierfür kann jedes Registerpaar außer dem BC-Register verwendet werden, da das BC-Register durch das Unterprogramm zum Abschalten des IRM zerstört wird.

Das Maschinenunterprogramm ab 0BE00H realisiert die gleiche Funktion wie das Beispiel auf Seite 217 der Beschreibung des BASIC-Interpreters auf Kassette C0111: Es erhöht den eingegebenen Parameter um 1 (INC DE) und gibt diesen Wert als Funktionswert zurück.

Adresse	MC	Anweisung	Bemerkung
BE00	13	INC DE ;	vom BASIC übergebener Parameter; wird um 1 erhöht
BE01	C9	RET ;	Rückkehr zum Rahmenprogramm

Die Eingabe des Rahmenprogramms sowie des Maschinenunterprogramms kann vom Betriebssystem-Niveau aus mit MODIFY erfolgen. Die Eingabe kann auch per BASIC-Programm geschehen, was im folgenden dargestellt wird.

```

10 REM Maschinencode des Rahmenprogramms
12 DATA 205. 111, 201, 205. 24. 240 205, 0, 190
14 DATA 205, 27, 240, 122 67, 205, 177. 208. 201
20 REM Maschinencode des Maschinenunterprogramms
22 DATA 19, 201
30 REM AD - Anfangsadresse des Rahmenprogramms
32 AD=0
34 REM MP - Anfangsadresse des Maschinenunterprogramms minus 8000H
36 MP=15872
40 REM Eingabe des Rahmenprogramms über POKE
42 FOR I=0 TO 17
44 READ B: POKE AD+I,B
46 NEXT
50 REM Eingabe des Maschinenunterprogramms über VPOKE
52 READ B: VPOKE MP,B: READ B: VPOKE MP 1+1,B
60 REM Speichern der Anfangsadresse auf 304H und 305H
70 DOKE 772, AD
80 REM Aufruf des Maschinenunterprogramms über des Rahmenprogramm
90 FOR 1I=250 TO 260
100 K =USR (I)
110 PRINT "I="; I,"USR(I)=";K
120 NEXT

```

Statt der einfachen Erhöhung des Parameters I um 1 kann ab der Adresse 0BE00H jedes beliebige andere Maschinenprogramm laufen, Dazu müßten die Zeilen 22 und 52 entsprechend geändert werden.