# Bedienung

(J.Müller, Hilfe zu JTCEMU)

## Das originale 2K-System

Das 2 KByte große Betriebssystem bietet die wichtigsten Funktionen zur Eingabe und Verwaltung von BASIC-Programmen. Des weiteren verwendet das Betriebssystem auch den im ROM des Einchipmikrorechners U883 enthaltenen BASIC-Interpreter.




### Kommandos

Nach dem Start bzw. nach einem RESET befindet sich das Betriebssystem im Kommandomodus. Das erkennt man an dem flimmernden Rechteck, unter dem sich der Buchstabe K verbirgt (K-Cursor). Ist der K-Cursor zu sehen, können Sie durch Drücken einer Taste (evtl. in Verbindung mit der Shift-Taste) folgende Kommandos eingeben, die auch sofort ausgeführt werden:

| **Eingabe** | **Kommando** | |
|---|---|---|
| Backspace | NEW | Löschen des BASIC-Programms, Achtung! Wenn Sie ein BASIC-Programm eingegeben haben und drücken die Backspace-Taste, wird das Programm ohne Nachfrage gelöscht. |
| - (Minus) | LIST | Anzeige einer Zeile des BASIC-Programms, Durch nochmaliges Drücken der Minus-Taste wird die nächste Programmzeile angezeigt. |
| + (Plus) | RUN | BASIC-Programm starten |
| * | CONT | BASIC-Programm nach der STOP-Anweisung fortsetzen |
| Shift - O | LOAD | BASIC-Programm von Kassette laden. Achtung! Sie sollten das Kommendo erst dann starten, wenn der Kennton zu hörern ist. |
| Shift - P | SAVE | BASIC-Programm auf Kassette speichern, Diese Kommando gibt das im Speicher befindliche BASIC-Programm auf dem Kassettenrekorderanschluss aus und führt danach, d.h. entsprechend der Länge des zu speichernden Programms nach einigen Sekunden bis einigen Minuten, eine Systeminitialisierung durch. |
| Shift - ENTER | OFF | Bisherige Eingabe löschen |

| Eingabe | Kommando | |
|---|---|---|
| A…Z (Buchstabe) | | Anzeige des Inhalts der gleichnamigen BASIC-Variable |
| 0…9 (Ziffer) | | Eingabe bzw. Löschen einer BASIC-Programmzeile |

## Eingabe von BASIC-Programmzeilen

Möchten Sie eine BASIC-Programmzeile eingeben, müssen Sie folgendermaßen vorgehen:

1. Eingabe der Zeilennummer beim K-Cursor
2. ENTER drücken. Es erscheint der sogenannte A-Cursor (flimmerndes Rechteck über einem A)
3. Eingabe einer BASIC-Anweisung durch Drücken des entsprechenden Zeichens (siehe Tabelle)
4. Eingabe der evtl. notwendigen Argumente der BASIC-Anweisung
5. Eingabe evtl. weiterer BASIC-Anweisungen, die mit einem Semikolon zu trennen sind.
6. ENTER

Beim A-Cursor können Sie folgende BASIC-Anweisungen eingeben:

| BASIC-Anweisung | Einzugebendes Zeichen |
|---|---|
| CALL | C |
| ELSE | > (größer gefolgt von Semikolon) |
| END | E |
| GOSUB | S |
| GOTO | G |
| IF…THEN… | F…,… (THEN: Komma) |
| INPUT | I |
| LET | L |
| PRINT | P |
| PTH (PRINTHEX) | H |
| PROC | O |
| RETURN | R |
| REM | M |
| STOP | T |
| TOFF | / |
| TRAP…TO… | !…,…(TO: Komma) |
| WAIT | W |

Bei der Eingabe prüft der Editor die Syntax. Ist diese falsch, wird die Eingabe mit ERROR 0 abgebrochen. Achtung! Wenn der A-Cursor zu sehen ist, hat der Editor die Zeilennummer bereits in das BASIC-Programm eingefügt. Sie sollten nun auf keinen Fall RESET auslösen, denn dann bleibt die Zeilennummer ohne Programmzeile im BASIC-Programm stehen, was einen Syntaxfehler bedeutet. Das BASIC-Programm ist somit an dieser Stelle defekt. Sie können aber die störende Zeile einfach löschen und so das Programm wieder reparieren.

## Einschränkungen bei Zeilennummern

Das Betriebssystem verwendet ein Null-Byte als Endekennung für das BASIC-Programm. Aus diesem Grund darf kein Null-Byte im BASIC-Programm vorkommen, weder in Zeichenketten noch in

Zeilennummern. Die Zeilennummern 0, 256, 512, 768, 1024 usw. enthalten jedoch in ihrem niederwertigen Byte eine Null. Aus diesem Grund wird bei der Eingabe einer solchen Nummer ihr Wert einfach um eins erhöht, d.h., aus 256 macht das Betriebssystem 257. Im höherwertigen Byte der Zeilennummer kann ein Null-Byte nicht vorkommen, da dort das Bit 7 immer gesetzt ist.

### Löschen von BASIC-Programmzeilen

Möchten Sie eine BASIC-Programmzeile löschen, drücken Sie nach Eingabe der Zeilennummer zweimal die ENTER-Taste, d.h., Sie lassen einfach die Anweisung weg. Zwar meldet der Editor ERROR 0, jedoch wird die Zeile trotzdem gelöscht.

## Das 4K-System EMR-ES 1988

Das 4 KByte große Betriebssystem ist eine Erweiterung des 2K-Systems und bietet neben der BASIC-Unterstützung auch Funktionen zur Eingabe und Verwaltung von Maschinencodeprogrammen sowie zum manuellen Beschreiben von Speicherbereichen.

### Startmenü

Nach dem Einschalten bzw. nach einem RESET meldet sich das Betriebssystem mit einem Startmenü, das folgende Einträge enthält:




### BASIC

Durch Drücken auf die B-Taste gelangt man in den BASIC-Modus, der weitestgehend dem 2K-Betriebssystem entspricht. Geändert wurden nur zwei Punkte:

- Unbekannte Schlüsselwörter werden bei ihrer Eingabe ignoriert und führen nicht mehr zur Initialisierung des Systems.
- Beim Speichern eines BASIC-Programms auf Kassette wird nun die Eingabe eines Dateinames erwartet. Das Speichern beginnt erst mit ENTER.

## INIT

Ein RESET initialisiert das System, jedoch bleibt der Inhalt des Arbeitsspeichers erhalten. Für die Eingabe eines Maschinencodeprogramms mit Marken ist jedoch ein gelöschter Arbeitsspeicher erforderlich. Dafür gibt es die INIT-Funktion, die neben der Systeminitialisierung auch den Arbeitsspeicher löscht. Um eine versehentliches Löschen des Speichers zu verhindern, müssen Sie nach dem Buchstaben I ein Ausrufezeichen eingeben.

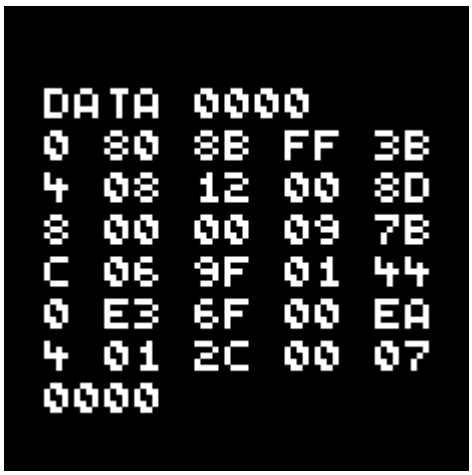
## SAVE

Mit der S-Taste gelangen Sie zum Programmmodul für das Auslagern von Speicherbereichen auf Kassette. Der Bildschirm wird gelöscht, und Sie geben nun die Anfangs- und Endadresse (jeweils vierstellig) des zu speichernden Bereichs sowie den Dateinamen ein. Der Bereich ab %FD00 darf nicht gespeichert werden. Mit Betätigen der ENTER-Taste beginnt das Speichern.

## LOAD

Mit der L-Taste und einem anschließenden ENTER wird eine Datei von Kassette geladen. Vor der ENTER können Sie optional eine Adresse eingeben, auf die dann das Programm geladen wird. In dem Fall wird die Ladeadresse, die in der Datei steht, ignoriert.

## DATA




Die DATA-Funktion, die Sie mit der Taste R erreichen, dient zum manuellen Beschreiben von Speicherbereichen. Es werden 24 Bytes des Speichers und darunter die aktuellen Adresse angezeigt. Nach Drücken von jeweils zwei Hexadezimalziffern wird das so eingegebene Byte in den Arbeitsspeicher geschrieben und die Anzeige aktualisiert.

Folgende Eingabe sind möglich:

| | |
|---|---|
| 0…9 A…F | Eingabe eines Datenbytes |
| - | Adresse - 1 |
| + | Adresse + 1 |

| | |
|---|---|
| ENTER | nächste Zeile |
| G<Adr.> | Adresse eingeben |
| OFF | DATA-Funktion verlassen |

### PROG

Das Programmmodul für die Eingabe von Maschinencodeprogrammen erreichen Sie mit der Taste P. Als erstes wird die Adresse für die Markentabelle (MTB) abgefragt. Diese setzt einen leeren Arbeitsspeicher voraus (ggf. vorher initialisieren) und läuft ähnlich einem Stack abwärts, d.h., Sie geben die Endadresse der Markentabelle ein. Danach befinden Sie sich im eigentlichen Eingabemodus, in dem neben den aus dem DATA-Modus bereits bekannten Eingabemöglichkeiten (z.B. Ändern der Adresse) folgende zusätzliche Funktionen geboten werden:

| | |
|---|---|
| :<ASCII><ASCII><ASCII> | Eingabe einer Marke (drei Stellen) |
| Q | Eingabe in Zeile 1 |
| X | Befehl streichen |
| I | Byte einfügen |
| T <Adr.> | Binden auf Adresse <Adr.> |
| S | Schritt-Test |
| N | Schleifen-Test |
| H <Adr.> | Haltepunkt auf Adresse <Adr.> |
| L | Echtzeit-Test |
| R | Aufruf DATA |

Maschinencodeprogramme wie das FORTH werden über LOAD geladen und dann über BASIC oder über PROG gestartet:

BASIC: 10 Call %... (20 End), starten (+)
PROG: Gxxxx, L

## Das 6K-System ES4.0




Das Betriebssystem beinhaltet einen BASIC-Editor und einen Maschinenmonitor. Nach dem Einschalten, d.h. nach dem Starten des Emulators, und nach einem RESET meldet sich der BASIC-Editor.

### BASIC-Editor

Ist als Prompt die Zeichenkette Edi zu sehen, befindet man sich im BASIC-Editor. Dieser kennt folgende Kommandos:

| Kommando | Bedeutung |
|---|---|
| C | Setzt das BASIC-Programm nach einer STOP-Anweisung fort |
| E | Listet das BASIC-Programm zeilenweise auf, Mit der Leertaste wird die nächste Zeile angezeigt. Hinter dem E kann auch eine Zeilennummer angegeben werden. Dann wird das Programm ab dieser Zeilennummer angezeigt. |
| L | Lädt ein Programm von Kassette auf die Standardadresse für BASIC-Programme (%E000), Dazu muss die Emulation der Kassettenschnittstelle aktiviert werden. Wurde das Programm fehlerfrei geladen, meldet sich der BASIC-Editor mit einer Null zurück, sonst mit 255. (siehe auch Hinweise zum Laden von Kassette) |
| N | Löscht das BASIC-Programm aus dem Arbeitsspeicher |
| M | Ruft den Maschinenmonitor auf |
| R | Startet das BASIC-Programm |
| S | Speichert ein BASIC-Programm auf Kassette, Dazu muss die Emulation der Kassettenschnittstelle aktiviert werden. |
| ?X | Gibt den Wert der Variable X aus |
| <Zahl> | Eingabe oder Löschen einer BASIC-Zeile, Wird hinter der Zahl nichts weiter angegeben, löscht der Editor die BASIC-Zeile mit dieser Zeilennummer. Anderenfalls wird eine BASIC-Zeile mit dieser Zeilennummer eingegeben. Dabei müssen die BASIC-Anweisungen in der Kurzform des komprimierten Speicherformats geschrieben werden. |

### Maschinenmonitor

Der Maschinenmonitor dient zum Anzeigen und Beschreiben von Speicherbereichen mit Hexadezimalzahlen und/oder ASCII-Zeichen. Die Tabelle zeigt die Kommandos des Maschinenmonitors, wobei alle einzugebenden Werte mit Ausnahme bei ASCII-Zeichen hexadezimal (groß geschrieben und ohne vorangestelltes Prozentzeichen) angegeben werden müssen.

| Kommando | Bedeutung |
|---|---|
| ,aaaa xx xx xx xx xx xx xx xx | Schreibt die max. 8 angegebenen Bytes in den Speicher ab Adresse aaaa |
| ;aaaa cccccccccccccccc | Schreibt die max. 16 angegebenen ASCII-Zeichen in den Speicher ab Adresse aaaa |
| ?aaaa bbbb | Berechnet aaaa+bbbb und aaaa-bbbb und gibt die beiden Ergebnisse aus |
| %aaaa | Gibt den vorzeichenbehafteten Dezimalwert der Hexadezimalzahl aaaa aus |
| #aaaa | Gibt den vorzeichenlosen Dezimalwert der Hexadezimalzahl aaaa aus |
| !rrxx | Schreibt den Wert xx in das Register rr |
| Faaaa nnnn xx | Füllt den Speicher ab Adresse aaaa mit nnnn Bytes xx |
| Haaaa | Zeigt ab Adresse aaaa 8 Bytes hexadezimal an |
| Jaaaa | Startet ein Maschinenprogramm auf Adresse aaaa mit einem CALL-Befehl, d.h., das Maschinenprogramm kann mit einem RET-Befehl beendet werden. |
| Laaaa | Lädt eine Datei von Kassette in den Arbeitsspecher ab Adresse aaaa. Nach dem Laden wird die Anzahl der geladenen Bytes und der Rückgabecode (Null bedeutet kein Fehler) ausgegeben. |
| Q | Beendet den Maschinenmonitor |
| Maaaa bbbb nnnn | Verschiebt nnnn Bytes im Arbeitsspeicher von Adresse aaaa nach bbbb, Das funktioniert auch bei Überschneiden der Speicherbereiche. |

| Kommando | Bedeutung |
|---|---|
| Rrr | Gibt der Wert des Registers rr aus |
| Saaaa nnnn | Speichert nnnn Bytes ab der Adresse aaaa auf Kassette, Das zweite Argument gibt nicht die Endadresse sondern die Anzahl der Bytes an. |

Es bedeuten:

```
aaaa bbbb nnnn 16-Bit-Hexadezimalzahlen
rr xx 8-Bit-Hexadezimalzahlen
c ASCII-Zeichen
```

**Editierfunktionen** Alle Eingaben erfolgen mit Hilfe eines Full-Screen-Editors. Dieser bietet komfortable Editierfunktionen, die in nachfolgender Tabelle gezeigt werden. In der ersten Spalte steht der hexadezimale Code der Funktion. Gibt man in einem Programm ein Zeichen mit diesem Code auf dem Bildschirm aus, wird die entsprechende Funktion ausgeführt. Für manche Funktionen gibt es auf der Tastatur des Emulator-Rechners keine speziellen Tasten. Diese Editierfunktionen können jedoch mit Hilfe einer CTRL- bzw. Strg-Tastenkombination benutzt werden (siehe letzte Spalte in der Tabelle).

| Code | Name | Bedeutung | CTRL-Tastenkombination |
|---|---|---|---|
| 01 | | Kursor links | CTRL-A |
| 02 | | Kursor rechts | CTRL-B |
| 03 | | Kursor hoch | CTRL-C |
| 04 | | Kursor runter | CTRL-D |
| 05 | HOM | Kursor in linke obere Ecke | CTRL-E |
| 06 | SOL | Kursor an Zeilenanfang | CTRL-F |
| 07 | DEL | Zeichen unter Cursor löschen | CTRL-G |
| 08 | DBS | Zeichen links von Kursor löschen | CTRL-H |
| 09 | INS | Zeichen einfügen | CTRL-I |
| 0A | LDE | Zeile löschen | CTRL-J |
| 0B | LIN | Zeile einfügen | CTRL-K |
| 0C | CLS | Bildschirm löschen | CTRL-L |
| 0D | RET | Eingabe abschließen | CTRL-M |
| 0E | ESC | Nächstes Zeichen nicht als Steuerzeichen interpretieren | CTRL-N |

From:
https://hc-ddr.hucki.net/wiki/ - **Homecomputer DDR**

Permanent link:
**https://hc-ddr.hucki.net/wiki/doku.php/tiny/bedienung?rev=1558515786**

Last update: **2019/05/22 09:03**