

**KLEINCOMPUTER**

**KC compact**

**KC compact-Floppy** Handbuch  
für den Programmierer

KLEINCOMPUTER

---

# KC compact

KC compact-Floppy  
Handbuch für den Programmierer

*de-pro-ma* *electronic GmbH*  
Mühlhausen

Ohne Genehmigung des Herausgebers ist es nicht gestattet, das Buch oder Teile daraus nachzudrucken oder auf fotomechanischem Wege zu vervielfältigen.

Redaktionsschluß dieser Ausgabe: Juni 1990

---

## Inhaltsverzeichnis

Präambel .....	5
----------------	---

### Hardware

1. Das KC compact-Floppy-System .....	6
1.1. Speicheraufteilung .....	6
1.2. I/O-Adresse .....	6
1.3. Floppy-Disk-Interface .....	7

### Software

#### Teil A

#### MicroDOS-Betriebsart

1. Einführung .....	8
2. Systemdatenstrukturen .....	10
2.1. Systemparameter .....	10
2.2. Der Dateisteuerblock .....	10
2.3. Der Standardpuffer .....	11
2.4. Der Systemsteuerblock .....	12
3. Logische Geräte .....	13
3.1. Laufwerke .....	13
3.2. Konsole und Drucker .....	13
4. Kommandoeingabe .....	15
5. BDOS-Funktionen .....	16
5.1. Vorbemerkungen .....	16
5.2. Aufruf der BDOS-Funktionen .....	16
5.3. Liste der BDOS-Funktionen .....	17
6. BIOS-Funktionen .....	40
6.1. Beschreibung der BIOS-Schnittstelle .....	40
6.2. Initialisierung .....	40
6.3. Logische Ein/Ausgabekanäle .....	41
6.4. Liste der BIOS-Funktionen .....	43
6.5. Verwaltung der Diskettenlaufwerke .....	46

---

## Teil B

### BASDOS-Betriebsart

1.	Systemstart .....	53
2.	Schnittstellenbeschreibung .....	54
2.1.	Verschiebliche RAM-Variablen .....	54
2.2.	Nichtverschiebliche RAM-Variablen des BASDOS .....	56
2.3.	Im BASDOS genutzte Betriebssystemvektoren .....	57
2.4.	RSX-Erweiterungen des BASDOS .....	58
3.	Diskettenformate .....	64
4.	Dateiaufbau .....	65
4.1.	ASCII-Dateien .....	65
4.2.	Nicht-ASCII-Dateien .....	65

### Anlagen

Anlage 1:	Reservierte Speicherplätze (Seite 0) .....	66
Anlage 2:	Bildschirmsteuerzeichen .....	67
Anlage 3:	ESC-Funktionen .....	68
Anlage 4:	BDOS-Übersicht .....	71
Anlage 5:	Programmierbeispiele: .....	72
Anlage 5.1.	DUMP-Ausgabe .....	72
Anlage 5.2.	Datei in Speicher des Grundgerätes kopieren .....	76
Literaturverzeichnis	.....	78
Abkürzungsverzeichnis	.....	79

## **P r ä a m b e l**

Das Erarbeiten von Anwenderprogrammen für ein KC compact-Floppy-System hat vor allem dann Bedeutung, wenn nicht auf Standardsoftware zurückgegriffen werden kann, die Standardsoftware zu modifizieren (installieren) ist oder Sonderprobleme zu lösen sind. Voraussetzung für den Programmierer sind entsprechende Hardwarekenntnisse sowie Kenntnisse in der Assemblerprogrammierung /1/. In diesem Handbuch erfolgt die Beschreibung der Hardware des KC compact-Floppy-Systems aus der Sicht des Programmierers. Die Erläuterung der Systemgrundlagen erfolgt getrennt in zwei Teilen für die beiden möglichen Betriebsarten. Im Teil A erfolgt die Beschreibung der Komponenten des MicroDOS zur Nutzung in Maschinenprogrammen /2/ bzw. In höheren Programmiersprachen /3/, /4/,/5/,/6/. Im Teil B zur BASDOS-Betriebsart werden Hinweise zu deren Nutzung in Anwenderprogrammen und zum Aufbau der Diskettendateien gegeben.

## Hardware

### 1. Das KC compact-Floppy-System

#### 1.1. Speicheraufteilung

Die KC compact-Floppy-Elektronik enthält 64 KByte dynamischen RAM und 8 KByte ROM, die durch den KC compact verwaltet werden. Sie fügen sich folgendermaßen in die Speicherverwaltung des KC compact ein:

Durch die Ausgabe eines Datenwortes auf das Bankswitchregister wird eine von acht möglichen RAM-Konfigurationen eingestellt. Das Bankswitchregister hat die I/O-Adresse 7FXXCX (siehe KC compact-Systemhandbuch S.8f.). Wenn der RAM des KC compact und der KC compact-Floppy-Elektronik in 16-KByte-Bereiche eingestellt wird und diese Bereiche von 0 bis 7 durchnummeriert werden, kann die Speicheraufteilung übersichtlich dargestellt werden. Die Speicherbänke 0 bis 3 sind im KC compact und die Speicherbänke 4 bis 7 in der KC compact-Floppy-Elektronik enthalten. Die folgende Tabelle enthält die Zuordnung der Speicherkonfigurationen zu den auf das Bankswitchregister ausgegebenen Datenbytes.

Datenbyte	Speicherbank auf			
	0000FH-3FFFH	4000H-7FFFH	8000H-BFFFH	C000H-FFFFH
C0H	0	1	2	3
C1H	0	1	2	7
C2H	4	5	6	7
C3H	reserviert			
C4H	0	4	2	3
C5H	0	5	2	3
C6H	0	6	2	3
C7H	0	7	2	3

Tabelle 1: Zuordnung der Speicherkonfiguration zu den Steuerbytes

Das ROM-Select (siehe KC compact-Systemhandbuch S. 17) wird benutzt, um den Erweiterungs-ROM der KC compact-Floppy-Elektronik in den Speicherbereich des KC compact einzublenden. Der ROM hat die Nummer 7.

#### 1.2. I/O-Adressen

Die KC compact-Floppy-Elektronik enthält einen Floppy-Disk-Kontroller und ein Flipflop zum Schalten der Motoren der Diskettenlaufwerke, die im I/O-Adreßraum des KC compact liegen. Sie sind über das Doppelregister BC zu adressieren (siehe KC compact-Systemhandbuch S. 8). Mit

```
LD BC,0FA7EH
OUT (C),C
```

werden die Motoren der Diskettenlaufwerke ausgeschaltet, mit

```
LD BC,0FA7FH
OUT (C),C
```

eingeschaltet. Der Floppy-Disk-Kontroller hat die Adressen FB7EH für sein Hauptstatusregister und FB7FH für das Datenregister.

### 1.3. Floppy-Disk-Interface

Der externe Anschluß für Diskettenlaufwerke hat die in folgender Tabelle aufgeführte Belegung.

Anschluß	Signal	Bedeutung	
A1	/FLT		Masse
A2	SE3		Masse
A3	/IX	Index - Spuranfang	Eingang
A4	SE0	Select Drive 0 - Auswahl	Ausgang
A5	SE2	Select Drive 0 - Auswahl	n.c.
A6	GND		Masse
A7	/ST	Step - Schritt	Ausgang
A8	/WD	Write Data - Schreibdaten	Ausgang
A9	/WE	Write Enable - Schreibbefehl	Ausgang
A10	GND		Masse
A11	GND		Masse
A12	GND		Masse
A13	/RDY	Ready - Bereitschaftsmeldung	Eingang
B1	/MO	Motor On - Motor ein	Ausgang
B2	GND		Masse
B3	GND		Masse
B4			Masse
B5	SE1	Select Drive 0 - Auswahl	Ausgang
B6	/TS	Two Side - zweiseitig	n.c.
B7	/ST	Step Direction - Schrittrichtung	Ausgang
B8	GND		Masse
B9	GND		Masse
B10	/TR0	Track 0 - Spur 0	Eingang
B11	/WP	Write Protected - Schreibschutz	Eingang
B12	/RD	Read Data - Lesedaten	Eingang
B13	/SS	Side Select - Kopfauswahl	Ausgang

n.c. (non connected):

Die KC compact-Floppy-Elektronik benutzt diese Signale nicht.

*Tabelle 2: Anschlußbelegung des Diskettenlaufwerkes*



## Software

### Teil A

## MicroDOS-Betriebsart

### 1. Einführung

Dieser Teil des Handbuches beschreibt die Systemorganisation, die Speicherstruktur und die Eintrittspunkte des Betriebssystems MicroDOS. Es werden die notwendigen Informationen gegeben, um Programme zu schreiben, welche unter MicroDOS laufen.

MicroDOS ist logisch in drei Teile aufgeteilt:

- in das Basis-Ein/Ausgabe-System (BIOS),
- in das Basis-Platten-Betriebssystem (BDOS), welches auch den Kommandoprozessor enthält und
- in das Feld für zeitweilig geladene Programme (TPA).

Das BIOS ist ein hardwareabhängiger Modul, welcher die Anbindung von peripheren Geräten auf einem niedrigen Niveau definiert. Das BDOS gewährleistet einen benutzerorientierten Zugriff zum Massenspeicher und den anderen peripheren Geräten. Als Besonderheit ist hierbei zu beachten, daß zum BIOS gehörende Programmteile im RAM des KC kompakt liegen, während alle anderen Teile des MicroDOS im RAM der KC compact-Elektronik abgearbeitet werden. Der TPA ist der Speicherbereich, in dem verschiedene nichtresistente Teile des Betriebssystems und Anwenderprogramme ausgeführt werden. Die unteren 256 Bytes des Speichers sind für Systemparameter reserviert.

Die Speicher im KC compact und in der KC compact-Floppy-Elektronik werden wie folgt aufgeteilt:

RAM der KC compact-Floppy-Elektronik

RAM des KC compact

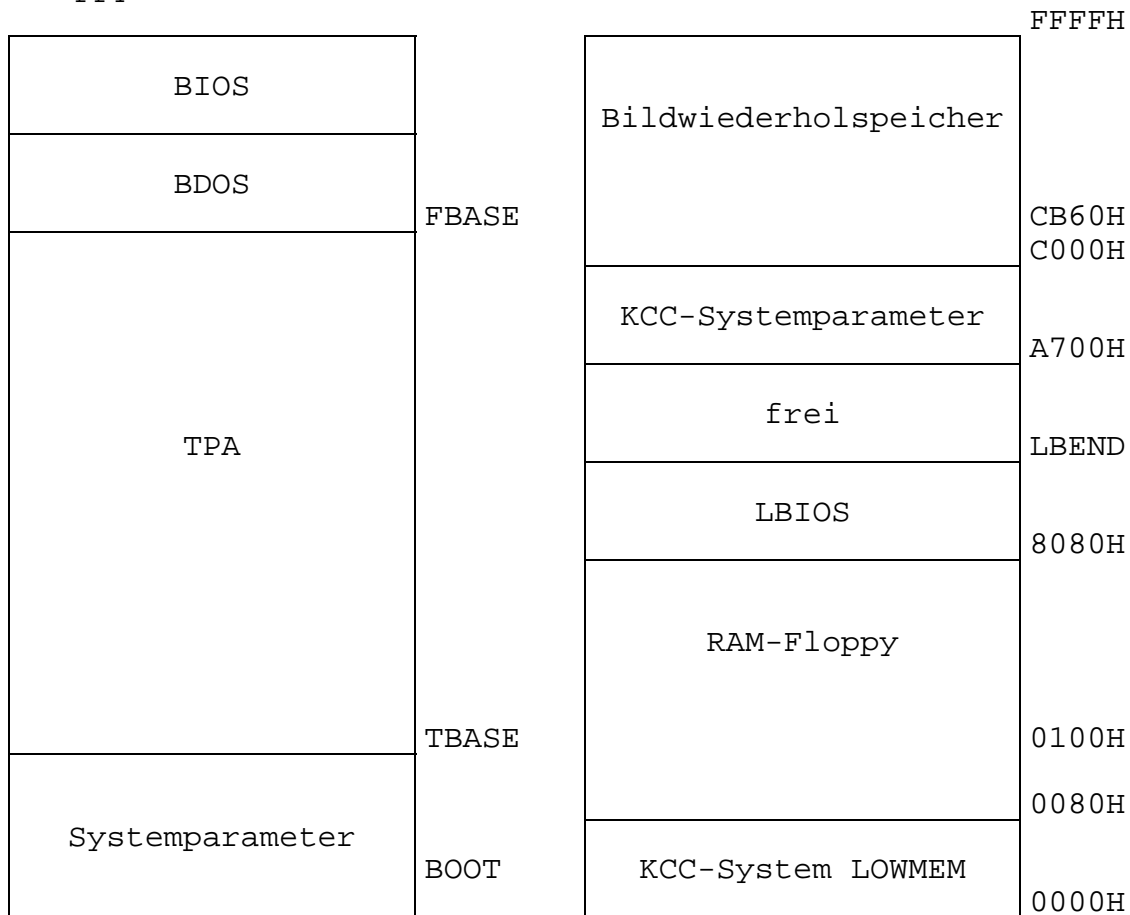


Bild 1: Speicheraufteilung im KC compact-Floppy-System

Die exakte Adresse von FBASE ist abhängig von der konkreten Version des MicroDOS. Der Maschinencode auf der Speicherzelle BOOT führt einen System-Warmstart durch, wobei alle Programme und Variablen geladen bzw. initialisiert werden, welche zur Übergabe der Steuerung an den Kommandoprozessor notwendig sind. Transiente Programme brauchen deshalb nur zur Position BOOT zu springen, um zum Kommandoniveau von MicroDOS zurückzukehren. Der prinzipielle Eintrittspunkt zum BDOS liegt bei BOOT +0005H, wo sich ein Sprung nach FBASE befindet. Die Zelle BOOT +0006H enthält den Wert von FBASE und kann zur Bestimmung des verfügbaren Speichers benutzt werden.

---

## 2. Systemdatenstrukturen

Im folgenden werden einige grundlegende Datenstrukturen des Betriebssystems erläutert.

### 1. Systemparameter

Auf den ersten 256 Bytes des Speichers befinden sich folgende vom System benutzte Datenfelder, die nachfolgend beschrieben werden.

Adresse	Bedeutung
0000H	Sprung zum Warmstart
0003H	I/O-Byte 0004H aktuelles Laufwerk
0005H	Sprung zum BDOS
0040H-0042H	Systemuhr
005CH	erster Dateisteuerblock
006CH	zweiter Dateisteuerblock
0080H	Standard-DMA-Puffer

Der Sprung auf Adresse 0000H führt direkt zum entsprechenden Ansprung in der BIOS-Sprungtabelle. Es werden der Warmstart durchgeführt und die Kontrolle an den Kommandoprozessor übergeben. Durch die Sprungadresse kann die Adresse des CP/M-kompatiblen BIOS Sprungvektors ermittelt werden.

Das I/O-Byte dient der Aufspaltung der logischen Kanäle auf verschiedene physische Geräte.

Die Nummer des aktuellen Laufwerkes wird aus Kompatibilitätsgründen zu CP/M 2.2 auf der Adresse 0004H abgelegt.

Auf Adresse 0005H befindet sich der allgemeine Eintrittspunkt in das BDOS. Gleichzeitig gibt die Sprungadresse den ersten für das System reservierten Speicherplatz oberhalb des TPA an.

Ab Adresse 005CH werden vom CCP, entsprechend den im Kommando angegebenen Parametern, ein oder zwei Dateisteuerblöcke angelegt (siehe auch Abschnitt 2.2). Für die Benutzung im Anwenderprogramm muß der zweite Dateisteuerblock auf einen anderen Speicherplatz verschoben werden, da dieser sonst vom BDOS überschrieben würde.

Auf Adresse 0080H beginnt der Standard-DMA-Puffer, welcher vom System für die Diskettenarbeit benutzt wird. Beim Aufruf eine Kommandos wird hier zusätzlich die Kommandozeile abgelegt.

### 2.2. Der Dateisteuerblock

Der Dateisteuerblock ist eine Datenstruktur, die vom Dateisystem beim Zugriff auf die Dateien über das Verzeichnis verwendet wird. Alle Operationen mit Dateien benötigen als Ausgangsinformation diese Daten. Außerdem belegen die Funktionen des wahlfreien Zugriffs die drei Bytes hinter dem Dateisteuerblock. Beim Aufruf der Dateifunktionen adressiert das Registerpaar DE den Dateisteuerblock für die interessierenden Datei.

Der Dateisteuerblock (FCB) besteht aus einem Feld von 33 Bytes im Fall des sequentiellen Zugriffs und einer Folge von 36 Bytes bei

wahlfreiem Zugriff. Der Standard-FCB auf Adresse 005CH kann für wahlfreien Zugriff benutzt werden, wenn die drei Bytes ab 007DH für diesen Zweck zur Verfügung stehen.

Im folgenden ist das Format des Dateisteuerblockes aufgezeigt:

```
dr f1 f2 --- f8 t1 t2 t3 ex s1 s2 rc d0 --- dn cr r0 r1 r2
00 01 02 --- 08 09 10 11 12 13 14 15 16 --- 31 32 33 34 35
```

mit:

```
dr      Laufwerkscode 0-16
        0 - aktuell ausgewähltes Laufwerk
        1 - Laufwerk A
        2 - Laufwerk B usw.

f1...f8 Dateiname in ASCII-Großbuchstaben Bit 7=0
        t1...t3   Dateityp in ASCII-Großbuchstaben Bit 7=0
                t1: Bit 7=1 - schreibgeschützte Datei
                t2: Bit 7=1 - Systemdatei
        ex       aktuelle Erweiterungsnummer, normalerweise =0, bei Ein/Ausgabe im Bereich 0-31
        s1       für interne Systembenutzung reserviert
        s2       für interne Systembenutzung reserviert
                =0 bei Datei eröffnen, erstellen, suchen
        rc       Datensatzanzahl (von 0-128)
        d0...dn  wird vom Betriebssystem ausgefüllt
        cr       Datensatzzähler für sequentiellen Zugriff
                wird vom Benutzer auf 0 gesetzt
        r0...r2  Nr. für wahlfreien Datensatz von 0-65535 mit Überlauf in r2, niederwertiges Byte in r0, höherwertiges Byte in r1
```

Jede Datei, die mit Hilfe von MicroDOS benutzt wird, muß einen dazugehörigen FCB haben, welcher den Dateinamen und eine Sektorbelegung für alle folgenden Operationen bereitstellt. Wenn auf eine Datei zugegriffen werden soll, ist es Sache des Programmierers, die niederen 16 Bytes des FCB auszufüllen und das "cr"-Feld zu initialisieren. Normalerweise werden die Bytes 1-11 mit ASCII-Zeichen für Dateinamen und Typ belegt, während alle anderen Bytes mit 0 gefüllt werden.

Die Dateisteuerblöcke werden im Inhaltsverzeichnis der Diskette gespeichert und vor der Durchführung einer Dateioperation in den Hauptspeicher gebracht (siehe BDOS-Funktionen Datei eröffnen bzw. erzeugen im Abschnitt 5.3.) Die Speicherkopie des Dateisteuerblockes wird während der Dateioperationen korrigiert und nach Abschluß derselben auf der Diskette abgespeichert (siehe Funktion Datei schließen).

### 2.3. Der Standardpuffer

Der Standardpuffer ab Adresse 80H wird vom System als Zwischenspeicher für Diskettenoperationen genutzt. Auch Anwenderprogramme können diesen Puffer verwenden. Zusätzlich wird vom Kommandointerpreter der Teil der Kommandozeile, der auf den Programmnamen folgt, im Standardpuffer abgelegt (siehe Abschnitt 4.).

### 3.4. Der Systemsteuerblock

Der Systemsteuerblock ist eine Datenstruktur, die sich im Basisdiskettenbetriebssystem befindet. MicroDOS nutzt diesen Bereich für die Sicherung der Systemdaten und Verbindungen von BDOS und BIOS. Die Anwenderprogramme können auch einige Parameter, die sich in diesem Bereich befinden, nutzen. Programme, die den Systemsteuerblock nutzen, können nur mit MicroDOS oder SCP bzw. CP/M Version 3.0 arbeiten.

Im folgenden sind die Felder des Systemsteuerblocks aufgeführt, die den Nutzerprogrammen zugänglich sind. Die Position jedes Feldes wird durch den Versatz bezüglich des Systemsteuerblockanfangs angegeben.

00-04	reserviert
05	Versionsnummer
06-09	Anwenderflags
0A-0F	reserviert
10-11	Rückkehrcode
12-19	reserviert
1A	Konsolenbreite
1B	aktuelle Spalte des Cursors
1C	Seitenlänge
1D-21	reserviert
22-23	Zuordnungsvektor für Konsoleneingabe
24-25	Zuordnungsvektor für Konsolenausgabe
26-27	Zuordnungsvektor für Zusatzeingabe
28-29	Zuordnungsvektor für Zusatzausgabe
2A-2B	Zuordnungsvektor für DruckerAusgabe
2C	Seitenmodus
2D	reserviert
2E	Backspace (^H) aktiv (0: Zeichen wird im Puffer und auf dem Bildschirm gelöscht 1: Zeichen wird nochmals auf dem Bildschirm ausgegeben)
2F	Rubout (DEL) aktiv (Bedeutung wie bei Backspace)
30-34	reserviert
35-36	Pufferadresse für Kalt- und Warmstart
37	Begrenzer für Zeichenkette (Standard \$=24H)
38	Druckerflag
39-3B	reserviert
3C-3D	aktuelle DMA-Adresse
3E	aktuelles Laufwerk
3F-43	reserviert
44	aktuelle Nutzernummer
45-49	reserviert
4A	Multisektorzähler (=1)
4B	BDOS-Fehlermodus (siehe BDOS-Funktion 45)
4C-4F	Laufwerke für Dateisuche
50	Laufwerk für temporäre Dateien
51	Laufwerksnummer für Fehleranzeige
54	Flag für Diskettenwechsel
52-56	reserviert
57	Flag für Länge der Fehlermeldung
58-59	Datum (nicht verwendet)
5A-5C	Zeit (nicht verwendet)
5D-5E	Adresse des gemeinsamen Speicherbereichs
5F-61	Sprung zur Fehlermeldung
62-63	Adresse des TPA-Endes (wird auf 6,7 kopiert)

### 3. Logische Geräte

#### 3.1. Laufwerke

Die auf dem KC compact-Floppy-System implementierte Version des MicroDOS verwaltet maximal acht logische Laufwerke mit den Bezeichnungen A...H.

Laufwerk A ist stets das RAM-Floppy (RAM-Disk), dessen Kapazität 32 KByte beträgt. Laufwerk B ist beim Start das Systemlaufwerk. Es liegt immer auf dem physischen Laufwerk 0 und hat das Format 5\*1024\*80\*2 mit zwei Systemspuren (780K). Die Laufwerke C...H können beliebig installiert werden.

#### 3.2. Konsole und Drucker

Als Konsole werden die Bildschirmausgabe und die Tastatureingabe bezeichnet.

Die Standardbildschirmausgabe (siehe Abschnitt 6.3.) erfolgt mit 24 Zeilen zu 80 Zeichen. Anlage 2 enthält die auch im SCP bzw. CP/M üblichen Steuerzeichen. Darüber hinaus enthält die Konsolenschnittstelle eine Vielzahl zusätzlicher Funktionen zur optimalen Nutzung der Möglichkeiten des KC compact. Zu diesen Funktionen gehören unter anderem die Grafik und die Farbsteuerung. Alle diese Funktionen werden über ESC-Folgen gesteuert. Durch Ausgabe des ESC-Codes und eines Unterscheidungscode wird die Konsolenausgabe auf die Sonderfunktion umgeschaltet. Anschließend wird ggf. eine der Funktion entsprechende Anzahl von Parametern übertragen. Anlage 3 gibt einen Überblick über die ESC-Funktionen. Bei Benutzung der ESC-Funktionen (außer der auch im SCP üblichen Cursorpositionierung) sollte über die BIOS-Schnittstelle (Abschnitt 6.4.) gearbeitet werden, da die BDOS-Schnittstelle (Abschnitt 5.2.) die Codierung 09H in einer Parameterliste als Tabulator interpretiert und in Leerzeichen (20H) umsetzt.

Als Beispiel sei das Setzen eines Punktes auf die Position X=100, Y=50 angegeben:

```
LD    A,18H      ;ESC
CALL  BIOSOUT   ;Konsolenausgabe
LD    A,'A'     ;Funktionscode
CALL  BIOSOUT
LD    A,100     ;X-Wert low
CALL  BIOSOUT
LD    A,0       ;X-Wert high
CALL  BIOSOUT
LD    A,50      ;Y-Wert
CALL  BIOSOUT
```

Die Tastaturbelegung wurde gegenüber der Standardbelegung im KC compact geändert. Durch einige ESC-Folgen können anwenderspezifische Tastaturcodierungen erzeugt werden. Einige Tasten haben die sofortige Ausführung ihrer Funktion unter Umgehung des BDOS und des eventuell laufenden Anwenderprogramms zur Folge.

Einige Tasten und Tastenkombinationen werden im MicroDOS direkt ausgeführt. Auf der Ebene eines Anwenderprogrammes bewirken sie keine Reaktion. Sie dienen zum Einstellen des Bildschirms und der Tastatur. Folgende Funktionen werden ausgeführt:

- CAPS LOCK Die Tastaturbelegung wird gewechselt
- ^ COPY Es wird ein Hardcopy des Bildschirminhaltes ausgedruckt Die Steuerzeichen sind für EPSON-kompatible Drucker gültig.
- ^ Cursor ein/aus. Es wird nur die Taste "." in der Nähe der CTRL-Taste angenommen.
- ^ 7 Wechsel der Vordergrundfarbe. Die 27 Farben werden von 0 bis 9 und von A bis Q durchnummeriert. Zur Eingabe einer entsprechenden Taste wird am unteren Bildrand aufgefordert.
- ^ 8 Wechsel der Hintergrundfarbe (entspricht ^7).
- ^ 9 Wechsel des Zeichensatzes. Mit dieser Tastenkombination kann zwischen amerikanischem und deutschem Zeichensatz hin- und hergeschaltet werden.

Die Standardbelegung aller Tasten im MicroDOS wird in der Anlage des KC compact-Floppy-Benutzerhandbuches aufgelistet.

Die Standarddruckerausgabe des MicroDOS arbeitet über einen Druckertreiber im RAM des KC compact. Dieser wird als CENTRONICS-Treiber mit acht Datenbits installiert.

#### 4. Kommandoeingabe

Beim Programmaufruf über die Kommandoeingabe analysiert der Kommandoprozessor die eingegebene Zeile und entscheidet, ob ein residentes Kommando eingegeben wurde oder ein Programm von Diskette geladen werden muß.

Der Kommandoprozessor konstruiert die ersten sechzehn Bytes von zwei möglichen Dateisteuerblöcken für ein transientes Programm durch Prüfung des Teils der Kommandozeile, der nach dem Namen des transienten Programms folgt. Dabei werden nichtspezifizierte Felder mit Leerzeichen gefüllt. Der erste Dateisteuerblock wird ab Adresse 005CH aufgebaut und kann in dieser Form für nachfolgende Dateioperationen genutzt werden. Der Anfang des zweiten Dateisteuerblocks liegt im d0...dn-Feld der ersten Dateisteuerung und muß vor der Benutzung in einen anderen Teil des Speichers umgeladen werden. Wenn z.B. der Benutzer folgende Zeile eingibt:

```
PROGRAMME B:DATEI1.XXX DATEI2.YYY
```

so wird die Datei "PROGRAMME" in den TPA geladen und der Standard-Dateisteuerblock ab Adresse 005CH mit dem Laufwerk 2, dem Dateinamen "DATEI1" und dem Dateityp "XXX" initialisiert. Der zweite Laufwerkscode bekommt den Wert ( ) zugewiesen, welcher ab Adresse 006CH geladen wird. Der Dateiname "DATEI2" wird ab Adresse 006DH abgelegt und der Dateityp "YYY" wird acht Bytes dahinter (Adresse 0075H) angeordnet.

Alle restlichen Felder bis "cr" werden auf Null gesetzt. Es muß nochmals darauf hingewiesen werden, daß es dem Programmierer obliegt, den zweiten Dateinamen und Dateityp in einen anderen Speicherbereich zu verschieben, bevor die Datei mit dem Steuerblock auf 005CH eröffnet werden kann, angesichts der Tatsache, daß die Eröffnungsfunktion den zweiten Dateinamen und -typ anderenfalls überschreibt.

Falls keine Dateinamen in dem Kommando angegeben wurden, enthalten die Felder ab 005CH und 006CH Leerzeichen. In jedem Fall werden Kleinbuchstaben in Großbuchstaben umgewandelt, um den MicroDOS-Dateinamenvereinbarungen zu entsprechen.

Für das oben angeführte Kommando würde der Puffer folgenden Inhalt aufweisen:

```
00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24
18   B : D A T E I 1 . X X X       D A T E I 2 . Y Y
Y.
```

wobei das erste Byte die Anzahl der gültigen Zeichen angibt und dann folgen die Zeichen selbst, welche vom Kommandoprozessor gegebenenfalls in Großbuchstaben umgesetzt werden. Die Kommando-parameter müssen vom Programm ausgewertet werden, bevor der Puffer durch Diskettenoperationen überschrieben wird.



## 5. BDOS-Funktionen

### 5.1. Vorbemerkungen

Das Diskettengrundbetriebssystem (BDOS) stellt den Kern des MicroDOS-Betriebssystems dar. Es unterstützt die Ein- und Ausgabe auf den logischen Geräten (Konsole, Drucker, Zusatzkanäle) sowie die Dateiarbeit auf den Disketten. Über BDOS-Rufe können folgende Aufträge angefordert werden:

- zeichenorientierte Ein- und Ausgabe
  - \* Zuordnung der logischen zu den physischen Geräten
  - \* zeichen- u. zeichenkettenweise Ein- u. Ausgabe über die Konsole
  - \* zeichenweiser Datenaustausch über die Zusatzkanäle
  - \* zeichenweise Ausgabe über den Druckerkanal
- Arbeit mit Disketten
  - \* Dateiverwaltung und -manipulation allgemein
  - \* sequentieller und direkter Dateizugriff
- Systemverwaltung
  - \* Initialisierung des Systems
  - \* Ermittlung des Systemzustandes
  - \* Zugriff auf Systemparameter

### 5.2. Aufruf der BDOS-Funktionen

Im folgenden werden die einzelnen Systemfunktionen mit ihren Eingangs- und Ausgangsparametern beschrieben. Der Aufruf der Systemfunktionen erfolgt durch einen CALL auf die Adresse 0005H, wobei die Funktionsnummer im Register C und notwendige Eingangsparameter im Register DE übergeben werden. Bei der Rückkehr werden die Ausgangsparameter im Akkumulator und im Registerpaar HL zurückgegeben. Feldeintragungen werden in einem Feld übergeben, dessen Anfang beim Aufruf im Registerpaar DE adressiert wird.

Es ist zu beachten, daß alle Register vom System verändert werden können. Deshalb ist es angebracht, wichtige Registerinhalte auf den Stack zu retten.

Das Betriebssystem benutzt einen eigenen Stapel, so daß das aufrufende Programm nur mit einer Stapelebene durch den BDOS-Aufruf belastet wird.

Bei der Benutzung der BDOS-Funktionen ist zu beachten, daß nur bei den BDOS-Rufen bis einschließlich Nummer 40 volle Kompatibilität zum SCPX gewährleistet ist. Das heißt, daß Programme die nur diese BDOS-Rufe verwenden, auch unter SCPX und CP/M 2.2 lauffähig sind. Programme, die die anderen BDOS-Rufe nutzen, können u.U. leistungsfähiger sein. Bei ihnen ist Kompatibilität, zu CP/M 3.0 und SCP 3.0 (PC1715W) gegeben.

MicroDOS entspricht einer CP/M-Version 2.6 und liegt damit im Funktionsumfang zwischen SCPX (CP/M 2.2) und SCP3 (CP/M 3.0).

### 5.3. Liste der BDOS-Funktionen

Die BDOS-Funktionen sind nach den Funktionsnummern geordnet.

#### **Funktion 0: System rücksetzen**

---

Eingangsparameter:  
Register C: 00H

Diese Funktion gibt die Steuerung an MicroDOS auf dem Kommando-Niveau zurück (Warmstart). Sie hat den gleichen Effekt wie ein Sprung auf die Adresse BOOT. Vor dem Aufruf der Funktion 0 kann durch Funktion 108 ein Rückkehrcode gesetzt werden.

#### **Funktion 1: Konsoleneingabe**

---

Eingangsparameter:  
Register C: 01H

Ausgangsparameter:  
Register A: ASCII-Zeichen

Die Konsoleneingabe-Funktion liest das nächste Zeichen von der Konsole in Register A ein. Darstellbare Zeichen und die Steuerzeichen Wagenrücklauf, Zeilenvorschub und Rückwärtsschritt (CTRL-H) werden zur Konsole zurückgegeben. Tabulatorzeichen (CTRL-I) werden expandiert auf Spalten mit acht Zeichen. Es wird überprüft, ob Start/Stop der Bildschirmausgabe (CTRL-S) oder Start/Stop der Druckerausgabe (CTRL-P) auftreten. Das BDOS kehrt nicht eher zum aufrufenden Programm zurück, bevor nicht ein Zeichen auf der Konsole eingegeben wurde.

#### **Funktion 2: System rücksetzen**

---

Eingangsparameter:  
Register C: 02H  
Register E: ASCII-Zeichen

Das in Register E enthaltene Zeichen wird zur Konsole gesendet. Wie bei Funktion 1 wird die Überprüfung von Start/Stop der Bildschirm- und Druckerausgabe durchgeführt. Ist die parallele Druckerausgabe eingeschaltet, so werden alle Zeichen auch an den Drucker gesendet.

**Funktion 3: Zusatzeingabe**

---

Eingangsparameter:  
Register C: 03H

Ausgangsparameter:  
Register A: ASCII-Zeichen

Die Zusatzeingabe liest ein Zeichen von dem zusätzlichen Kanal in das Register A. Die Steuerung kehrt nicht zurück, bevor nicht ein Zeichen gelesen wurde.

**Funktion 4: Zusatzausgabe**

---

Eingangsparameter:  
Register C: 04H  
Register E: ASCII-Zeichen

Die Zusatzausgabefunktion sendet das im Register E enthaltene Zeichen zum zusätzlichen logischen Kanal.

**Funktion 5: Druckerausgabe**

---

Eingangsparameter:  
Register C: 05H  
Register E: ASCII-Zeichen

Die Druckerausgabefunktion sendet das im Register E enthaltene Zeichen zum logischen Drucker.

**Funktion 6: direkte Konsolen-Ein/Ausgabe**

---

Eingangsparameter:  
Register C: 06H  
E: 0FFH (Eingabe) oder  
ASCII-Zeichen (Ausgabe)

Ausgangsparameter:  
Register A: Zeichen oder Status

Die direkte Ein/Ausgabe wird von MicroDOS unterstützt für solche speziellen Anwendungen, wo einfache Ein/Ausgabeoperationen gefordert sind. Im allgemeinen sollte die Benutzung dieser Funktion vermieden werden, da die normalen Steuerzeichenfunktionen von MicroDOS (d.h. CTRL-D und CTRL-P) umgangen werden. Bei Eintritt in die Funktion 6 enthält Register E entweder den hexadezimalen Wert 0FFH, was eine Konsoleneingabe kennzeichnet, kehrt die Funktion mit A=0 zurück, falls kein Zeichen von der Konsole bereitsteht. Ansonsten enthält Register A das nächste Eingabezeichen von der Konsole. Ist der Wert des Registers E nicht gleich 0FFH, so behandelt die Funktion 6 diesen Wert als gültiges ASCII-Zeichen und sendet dieses zur Konsole.

#### **Funktion 7: Status Zusatzausgabe**

---

Eingangsparameter:  
Register C: 07H

Ausgangsparameter:  
Register A: Status

Die Funktion 7 realisiert in der auf dem KC compact-Floppy-System vorliegenden Implementierung des MicroDOS die Abfrage des Zustandes der Druckerausgabe. Wenn freier Platz im Übergabepuffer zum Druckertreiber ist, wird im Akkumulator der Wert 0FFH übergeben, ansonsten der Wert 00H.

#### **Funktion 8: Status Zusatzausgabe**

---

Eingangsparameter:  
Register C: 07H

Ausgangsparameter:  
Register A: Status

Die Funktion 8 ist in der vorliegenden Implementierung des MicroDOS kurzgeschlossen und übergibt stets im Akkumulator den Wert 00H.

#### **Funktion 9: Zeichenkette ausgeben**

---

Eingangsparameter:  
Register C: 09H  
Registerpaar DE: Adresse der Zeichen

Die Zeichenkettenausgabefunktion sendet Zeichen aus dem Spei-

cher, beginnend ab der in DE übergebenen Adresse, zur Konsole, bis in der Zeichenkette das im Systemsteuerblock gesetzte Endekennzeichen erkannt wird. Standardmäßig wird '\$' als Endekennzeichen angenommen. Tabulatoren werden expandiert, wie in Funktion 2, und Start/Stop-Zeichen für Bildschirm- und Druckerausgabe werden ausgewertet. Bei paralleler Druckerausgabe werden alle Zeichen auf dem Drucker ausgegeben.

### **Funktion 10: Konsolenpuffer lesen**

Eingangsparameter:

Register C: 0AH  
Registerpaar DE: Pufferadresse

Ausgangsparameter:

Zeichen von der Konsole im Puffer

Diese Funktion liest eine Zeile von eingegebenen Zeichen in einen Puffer, der durch DE adressiert ist. Die Eingabe wird abgeschlossen, wenn der Eingabepuffer überläuft oder wenn ein RETURN eingegeben wird.

Der Puffer erhält folgende Form

```
DE +0 +1 +2 +3 +4 +5 +6 +7 ... +n
mx nc c1 c2 c3 c4 c5 c6 ... ??,
```

wobei mx die maximale Anzahl von einzulesenden Zeichen (1 bis 255) und nc die tatsächliche Anzahl von eingelesenen Zeichen (vom BDOS bei Rückkehr gesetzt) ist. Es folgen die eingelesenen Zeichen. Ist nc kleiner als mx, so folgen dem letzten Zeichen unbestimmte Informationen, die im obigen Beispiel mit "??" gekennzeichnet sind. Eine Reihe von Steuerzeichen wird während der Eingabe erkannt.

DEL	löscht das letzte Zeichen
CTRL-C	Neustart von MicroDOS (am Beginn der Zeile)
CTRL-E	gibt das physikalische Ende der Zeile an
CTRL-H	Rückwärtsschritt
CTRL-J	(Zeilenvorschub) beendet die Eingabe
CTRL-M	(RETURN) beendet die Eingabe
CTRL-R	schreibt die aktuelle Zeile neu
CTRL-U	löscht die aktuelle Zeile
CTRL-X	geht zurück zum Anfang der Zeile
CTRL-W	wiederholt den letzten Pufferinhalt
CTRL-^	löscht das letzte Wort

Die Erneuerung des Pufferinhalts bei Eingabe von CTRL-W erfolgt über den Pufferzähler. Wird dieser im Anwenderprogramm verändert, kann der regenerierte Inhalt falsch sein.

**Funktion 11: Konsolenstatus holen**

---

Eingangsparameter:

Register C: 0BH

Ausgangsparameter:

Register A: Konsolenstatus

Die Funktion Konsolenstatus überprüft, ob ein Zeichen auf der Konsole eingegeben wurde. Falls ein Zeichen eingegeben wurde, so wird im Register A ein Wert ungleich 0 übergeben, im allgemeinen der Wert 01H. Ist kein Eingabezeichen vorhanden, kehrt die Funktion mit A=0 zurück.

**Funktion 12: Versionsnummer holen**

---

Eingangsparameter:

Register C: 0CH

Ausgangsparameter:

Registerpaar HL: Versionsnummer

Die Funktion 12 übergibt einen 2-Byte-Wert, wobei H=00H anzeigt, daß es sich um ein CP/M 2.2-kompatibles System handelt (H=01H bei CP/M). Im Register L übergibt MicroDOS einen hexadezimalen Wert entsprechend der Versionsnummer des Systems, also z.B. 26H bei der Version 2.6. Die Verwendung dieser Funktion erlaubt das Schreiben von versionsabhängigen Programmen.

**Funktion 13: Diskettensystem zurücksetzen**

---

Eingangsparameter:

Register C: 0DH

Diese Funktion wird benutzt, um vom Programm aus das Dateisystem in den Anfangszustand zu setzen, Dabei sind alle Laufwerke für Schreiben und Lesen zugelassen (siehe Funktionen 28 und 29), Laufwerk A ist angewählt und die DMA-Adresse ist auf den Standardwert 0080H festgelegt. Diese Funktion kann von Programmen verwendet werden, die während der Abarbeitung einen Diskettenwechsel erfordern. Siehe dazu auch Funktion 37.

**Funktion 14: Laufwerk anwählen**

---

Eingangsparameter:

Register C: 0BH

Register E: Laufwerkscode

Die Funktion kennzeichnet das in E bezeichnete Laufwerk als das aktuelle Laufwerk für die folgenden Dateioperationen, wobei E=0 für Laufwerk A steht, 1 für Laufwerk B usw. bis 7 für Laufwerk H in einem voll ausgebauten System mit acht logischen Laufwerken.

Das Laufwerk wird in einem "on line"-Status versetzt, welcher insbesondere das entsprechende Inhaltsverzeichnis bis zum nächsten Kaltstart, Warmstart oder Systemrücksetzen aktiviert. Dateisteuerblöcke, die den Laufwerkscode 0 aufweisen, beziehen sich automatisch auf das aktuelle Laufwerk. Laufwerkscodes von 1 bis 16 ignorieren die Standard-Anwahl und beziehen sich direkt auf ein Laufwerk von A bis H. War die Operation erfolgreich, wird im Akkumulator der Wert 00H zurückgegeben. Tritt jedoch ein Fehler auf, so wird bei der Fehlerbehandlung durch das System eine entsprechende Meldung ausgegeben und der Warmstart vollzogen. Bei Fehlerbearbeitung durch das Anwenderprogramm enthält Register A den Wert 0FFH und Register H einen der folgenden Fehlercodes:

01: Diskettenfehler

04: Auswahlfehler

**Funktion 15: Datei eröffnen**

---

Eingangsparameter:

Register C: 0FH

Registerpaar DE: FCB-Adresse

Ausgangsparameter:

Register A: Verzeichniscode

Die Dateieröffnung wird benutzt, um eine Datei zu aktivieren, die bereits auf der angewählten Diskette existiert. Das BDOS sucht im Inhaltsverzeichnis nach Übereinstimmung mit den Positionen 1 bis 14 des durch DE adressierten Speicherblocks (Byte s1 ist automatisch auf Null gesetzt), wobei ein Fragezeichen (3FH) in jeder dieser Positionen als Übereinstimmung gewertet wird. Normalerweise ist jedoch keine Fragezeichen eingefügt und weiterhin sind die Bytes "ex" und "s2" gleich Null. Wenn ein Eintrag des Inhaltsverzeichnisses übereinstimmt, wird die relevante Information in die Bytes "d0" bis "dn" des Dateisteuerblocks kopiert, wodurch der Zugriff auf diese Datei bei nachfolgenden Lese- und Schreiboperationen ermöglicht wird. Auf eine Datei darf nicht zugegriffen werden, bevor nicht eine entsprechende Eröffnung erfolgreich durchgeführt wurde. Die Funktion 15 übergibt in A einen

"directory code" mit dem Wert von 0 bis drei, falls die Eröffnung erfolgreich durchgeführt wurde. Dieser Wert kennzeichnet, an welcher Stelle im DMA-Puffer der interessierende Eintrag liegt. Falls die Datei nicht gefunden wurde, wird im Register A der Wert OFFH übergeben. Der aktuelle Datensatz "cr" muß vom Programm auf Null gesetzt werden, wenn die Datei sequentiell vom ersten Datensatz an gelesen werden soll.

Bei Fehlerbehandlung im Anwenderprogramm wird einer der folgenden Fehlercodes im Register H übertragen:

00: Datei existiert nicht  
01: Diskettenfehler  
04: Auswahlfehler  
09: mehrdeutiger Name

#### **Funktion 16: Datei schließen**

---

Eingangsparameter:

Register C: 10H  
Registerpaar DE: FCB-Adresse

Ausgangsparameter:

Register A: Verzeichniscode

Diese Funktion führt die Umkehrung der Dateieröffnung durch. Wurde der durch DE adressierte Dateisteuerblock durch eine vorhergehende Eröffnung oder Erzeugung (siehe Funktionen 15 und 22) aktiviert, so speichert die Dateischließung den Dateisteuerblock in das Inhaltsverzeichnis der Diskette. Der FCB-Suchvorgang ist derselbe wie bei der Eröffnungsfunktion. Der Ausgangsparameter im Akkumulator bei einem erfolgreichen Abschluß ist 0, 1, 2 oder 3, während OFFH übergeben wird., falls der Dateiname im Inhaltsverzeichnis nicht gefunden wurde. Nach Leseoperationen braucht eine Datei nicht geschlossen zu werden. Nach Schreiboperationen ist in jedem Fall die Datei zu schließen, um den neuen Eintrag im Inhaltsverzeichnis abzuspeichern.

Die Fehlercodes für die Fehlerbehandlung im Anwenderprogramm entsprechen denen der Funktion 15.

#### **Funktion 17: ersten Eintrag suchen**

---

Eingangsparameter:

Register C: 11H  
Registerpaar DE: FCB-Adresse

Ausgangsparameter:

Register A: Verzeichniscode

Diese Funktion durchsucht das Inhaltsverzeichnis auf Übereinstim-



mung mit dem durch DE adressierten Dateisteuerblock. Der Wert OFFH wird übergeben, wenn die Datei nicht gefunden wurde, andernfalls die Werte 0, 1, 2 oder 3, welche anzeigen, daß die Datei vorhanden ist. Der aktuelle DMA-Bereich wird mit dem Datensatz des Inhaltsverzeichnisses gefüllt, welcher dem FCB entspricht. Das Register A gibt in diesem Fall die Nummer des Eintrages im Datensatz an. Die relative Anfangsadresse kann folglich mit  $A*32$  berechnet werden. Obwohl es normalerweise für Anwenderprogramme nicht notwendig ist, kann der Inhaltsverzeichniseintrag von dieser Position geholt werden.

Ein Fragezeichen (3FH) in irgendeiner Position von "fl" bis "ex" bedeutet Übereinstimmung mit dem entsprechenden Feld im Inhaltsverzeichnis des angewählten Laufwerkes. Wenn das "dr"-Feld ein Fragezeichen enthält, ist die automatische Laufwerksauswahl unterbunden und das aktuelle Laufwerk wird auf Übereinstimmung geprüft, wobei die Suchfunktion jeden übereinstimmenden Eintrag übergibt, unabhängig von der Benutzernummer. Wenn das "dr"-Feld kein Fragezeichen enthält, wird das "s2"-Byte automatisch auf Null gesetzt.

Auch diese Funktion übergibt die oben genannten Fehlercodes an das aufrufende Programm, wenn die Fehlerbearbeitung durch das Anwenderprogramm eingestellt ist.

#### **Funktion 18: nächsten Eintrag suchen**

---

Eingangsparameter:

Register C: 12H  
Registerpaar DE: FCB-Adresse

Register A: Verzeichniscode

Diese Funktion entspricht der Funktion 17 bis auf die Tatsache, daß die Suche im Inhaltsverzeichnis ausgehend vom letzten übereinstimmenden Eintrag fortgesetzt wird. Wie bei Funktion 17 übergibt die Funktion 18 den Wert (OFFH), wenn kein Eintrag mehr gefunden wurde.

Für die richtige Ausführung der Funktion 18 dürfen zwischen den Operationen 17 und 18 und zwischen aufeinanderfolgenden Aufrufen der Funktion 18 keine anderen Diskettenoperationen des BDOS folgen.

#### **Funktion 19: Datei löschen**

---

Eingangsparameter:

Register C: 13H  
Registerpaar DE: FCB-Adresse

Ausgangsparameter:

Register A: Verzeichniscode

---

Diese Funktion entfernt Dateien, die mit dem durch DE adressierten Dateisteuerblock übereinstimmen. Dateiname und Dateityp können mehrdeutig sein. (d.h., Fragezeichen in verschiedenen Positionen). Der Laufwerkscode darf nicht mehrdeutig sein, wie bei den Funktionen 17 und 18.

Die Funktion 19 übergibt bei erfolgreicher Abarbeitung im Akkumulator den Verzeichniscode von 0 bis 3 und beim Auftreten eines Fehlers den Wert (OFFH). Im Modus der Fehlerbearbeitung durch das System erfolgen eine Fehlerermittlung und anschließend der Warmstart. Bei Fehlerbearbeitung durch das Anwenderprogramm werden die folgenden Codes übergeben:

00: Datei existiert nicht  
01: Diskettenfehler  
03: Datei geschützt (Schreibschutz oder Systemdatei)  
04: Auswahlfehler.

#### **Funktion 20: sequentiell lesen**

---

Eingangsparameter:

Register C: 14H  
Registerpaar DE: FCB-Adresse

Ausgangsparameter:

Register A: Fehlercode

Wurde der durch DE adressierte Dateisteuerblock mittels der Funktion 15 aktiviert,

wird fortgesetzt

mrzvb