

```
*****
*
*           E O S           *
*
*   Betriebssystem für Microcomputer   *
*   mit Z80-CPU                       *
*
*           Version 3           *
*
*           Systemhandbuch        *
*
*****
```

Copyright (C) 1983 Däumling & Zimmermann

Die Vervielfältigung auch von Teilen dieses Handbuchs bedarf unserer schriftlichen Genehmigung.

EOS ist ein eingetragenes Warenzeichen der Firma Däumling & Zimmermann, Seevetal, Deutschland.

Z80 ist ein eingetragenes Warenzeichen von Zilog, Cupertino, California.

CP/M ist ein eingetragenes Warenzeichen von Digital Research, Pacific Grove, California.

Inhaltsverzeichnis

Abschnitt A: Systemschnittstelle	2
A.1. Einleitung	3
A.2. Systemaufrufe	4
A.3. Die Zuordnung von Ein/Ausgabegeräten	5
A.4. Der Datei - Versorgungsbereich FCB	7
A.5. Sonderstrukturen im Directory	9
A.5.1. Der Directory Label	9
A.5.2. Die Zeiteinträge	10
A.6. Die Initialisierung der Zero Page	11
A.7. Funktionsmodule	13
A.7.1. Die Speicherorganisation nach dem Laden von Funktionsmodulen	14
A.7.2. Der Programm - Header	15
A.7.3. Die Struktur eines Funktionsmoduls	16
A.7.4. Beispiel eines Funktionsmoduls	18
A.8. Der System - Variablenbereich	19
A.9. Die EDS - Flags	23
A.10. Systemfehler	24
A.11. Liste der verfügbaren Systemaufrufe	26
 Abschnitt B: Hardware - Schnittstelle	125
B.1. Einleitung	126
B.2. Aufruf der Hardware - Funktionen	127
B.3. Die Hardware - Einsprungtabelle	128
B.4. Fehlermeldungen der Hardware - Aufrufe	129
B.5. Fest installierte Laufwerke	131
B.6. Der Skew Factor	131
B.7. Der Disk Parameter Block	132
B.8. Tabelle der Hardware - Funktionsaufrufe	135
 Abschnitt C: Grafik - Schnittstelle	165
C.1. Einleitung	166
C.2. Die Definition des Bildschirms	167
C.3. Die Definition von Bit Images	168
C.4. Tabelle der Grafikfunktionen	169
 Abschnitt D: Bildschirm - Schnittstelle	183
D.1. Einleitung	184
D.2. Die implementierten Normen	184
D.3. Tabelle der vorhandenen SteuerCodes	185
D.4. Nähere Beschreibung der einzelnen Sequenzen	187
D.4.1. Cursorposition setzen	187
D.4.2. Cursorposition lesen	187
D.4.3. Attribute für Zeichen setzen	188
D.4.4. Cursor - Attribute setzen	189
D.4.5. Bildschirm ein- und ausblenden	189
D.4.6. Bildschirm-Editor einschalten	190
D.4.7. Monitor Mode	190
D.4.8. Hardcopy des Bildschirms	190
D.4.9. Uhrzeit ein- und ausschalten	191
D.5. Die Statuszeile im TV950-Mode	192
D.5.1. Die System - Statuszeile	192
D.5.2. Die Benutzer - Statuszeile	193
D.5.3. Unterdrückung der Statuszeile	193
D.6. Die Tastatur	194
D.6.1. Die alphanumerische Gesamt-Tastatur	194
D.6.2. Die numerische Zehner-Tastatur	195
D.6.3. Frei definierbare Funktionstasten	196
D.6.3.1. Steuer-Codes für die Tastendefinition	197
D.6.4. Vordefinition der Funktionstasten	198
D.6.5. Beispiel für eine Tastenprogrammierung	199

Dieses Handbuch beschreibt den logischen Aufbau und die innere Struktur des Systemkerns von EOS und dessen Systemaufrufe. Voraussetzung für das Verständnis sind profunde Kenntnisse der Z80 - CPU und deren Assemblersprache. Da einige Aufrufe, falsch angewendet, zu Datenverlusten und Systemabstürzen führen können, ist von ihrer Verwendung von Laien dringend abzuraten.

Das Handbuch ist in vier Abschnitte unterteilt. Im ersten Abschnitt wird die Systemschnittstelle zum EOS - Systemkern hin beschrieben. Es wird näher auf die Aufrufmechanismen und die dort verwendeten Datenstrukturen eingegangen. Der zweite Abschnitt behandelt die Schnittstelle zu der Hardware des MZ-3500, die dort verwendeten Systemfunktionen und Datenstrukturen. Der dritte Abschnitt ist der Grafik gewidmet. Dort sind alle Möglichkeiten, aus der Assemblerebene her die Grafik zu bedienen, aufgeführt. Schließlich behandelt der letzte Abschnitt den Bildschirm und die dort implementierten Funktionen.

Abschnitt A
die EOS - Systemschnittstelle

A.1. Einleitung

Dieser Abschnitt behandelt die Schnittstelle zum Systemkern von EOS hin. Diese Schnittstelle ist kompatibel zu den Betriebssystemen CP/M und DiCOS ausgelegt, wobei die CP/M - Version 3.0 und die DiCOS - Version 1.1 unterstützt werden. Ferner sind in diesem Abschnitt die Verwaltung der Plattenlaufwerke sowie die dazugehörigen Datenstrukturen beschrieben. Die Möglichkeit, das Verhalten des Systemkerns durch das Ändern von internen Variablen zu beeinflussen, ist ebenfalls Gegenstand dieses Abschnitts. Nicht dokumentiert ist hier die Hardware - Schnittstelle sowie die Schnittstelle hin zur Grafik und zum Bildschirm. Diese Dokumentationen sind Gegenstand anderer Abschnitte.

A.2. Systemaufrufe

Das Format aller Systemaufrufe ist gleich: Der Funktionscode wird in Register C, eine eventuelle Zusatzversorgung in Register E bzw. im Registerpaar DE übergeben. Eventuelle Ergebnisse werden im Registerpaar HL zurückgeliefert. Zusätzlich ist Register A mit dem Inhalt von L und Register B mit dem Inhalt von H belegt. Der Aufruf selbst erfolgt durch einen Unterprogrammsprung auf die Adresse 0005H. Ein Aufruf, um ein Zeichen auszugeben, hätte z. B. folgendes Format:

LD	C,2	; Funktionscode
LD	E,'*'	; Sternchen ausgeben
CALL	5	; Systemaufruf

Das System selbst hat seinen eigenen Stack - Bereich. Es ist also nicht nötig, einen ausreichend großen Stackbereich zur Verfügung zu stellen.

A.3. Die Zuordnung von Ein/Ausgabegeräten

Die Gerätezuweisung wurde unter EOS komplett neu gestaltet. Es wurden fünf logische Geräte definiert. Diese Geräte sind:

Console Input	für die Funktionen 1, 6, 10, 11
Console Output	für die Funktionen 2, 6, 9, und 111
Aux Device Input	für die Funktionen 3 und 7
Aux Device Output	für die Funktionen 4 und 8
List Output	für die Funktionen 5 und 112

Die physikalischen Geräte sind vom Implementator frei definierbar. Diese Geräte werden über eine Gerätetabelle wie folgt definiert:

DB	NAME	Gerätename, 6 Bytes lang
DB	ATTRIBUT	Attribut - Byte
DB	BAUD	Einstellbare Baud Rate

Das Attribut - Byte beschreibt die Eigenschaften des Geräts näher:

- Bit 0 gesetzt - Gerät ist für Eingabe geeignet
- Bit 1 gesetzt - Gerät ist für Ausgabe geeignet
- Bit 2 gesetzt - die Baud Rate kann softwaremäßig eingestellt werden
- Bit 3 gesetzt - Gerät benutzt CTS - Protokoll
- Bit 4 gesetzt - Gerät benutzt XON/XOFF - Protokoll

In dieser Tabelle sind sämtliche Ein/Ausgabegeräte außer den Platteneinheiten definiert. Diese Geräte sind intern durchnummeriert, wobei der erste Eintrag in der Gerätetabelle die Nummer 0 bekommt, der zweite die Nummer 1 etc. Jedem logischen Gerät ist ein Bitvektor von 16 Bits entsprechend 16 physikalischen Geräten zugeordnet, wobei das höchstwertige Bit dem physikalischen Gerät 0 entspricht, das zweithöchste dem Gerät 1 usw. Um ein logisches Gerät mit einem physikalischen Gerät zu verbinden, wird in dessen Vektor einfach das entsprechende Bit gesetzt.

Dieses Konzept gestattet es natürlich, einem logischen Gerät gleichzeitig mehrere physikalische Geräte zuzuordnen, indem man mehr als ein Bit gleichzeitig in dem entsprechenden Bitvektor setzt. Per Definition geschieht dann folgendes:

Handelt es sich um ein Eingabegerät, werden alle angeschlossenen Geräte von 0 an aufwärts überprüft, ob ein Zeichen einzulesen ist. Das erste Gerät, das ein Zeichen anzubieten hat, wird dann von EOS angesprochen.

falls das Gerät jedoch ein Ausgabegerät ist, wird das Zeichen auf allen angeschlossenen Geräten ausgegeben. Es wird jeweils solange gewartet, bis das entsprechende Gerät bereit ist, das auszugebende Zeichen zu empfangen. Erst nachdem die Ausgabe erfolgt ist, wird das nächste Ausgabegerät angesprochen. Die Wartezeit auf ein Ausgabegerät kann durch die Funktion 73 gesteuert werden. Nachdem die eingestellte Zeit abgelaufen ist, erscheint auf dem Bildschirm die Meldung:

Gerät <name> nicht bereit

wobei an der durch "<name>" gekennzeichneten Stelle der Name des physikalischen Geräts eingesetzt wird, das nicht bereit ist. Danach wird das Eingabegerät CONIN: überprüft, ob ein Zeichen anliegt. falls dies der Fall ist, wird das Zeichen eingelesen und die Ausgabe abgebrochen. Ein eventuell parallel laufender Drucker wird ebenfalls abgeschaltet. Ist hingegen kein Zeichen von der Tastatur eingegeben worden, wird der Wartezyklus wiederholt.

Eine Sonderstellung nimmt hierbei das physikalische Gerät 0 ein. Dieses wird per Definition zum Bildschirm - Terminal erklärt. Das hat zur Folge, daß keine Fehlermeldung erfolgt, wenn der Bildschirm nicht bereit ist. Zusätzlich unterbleibt die parallele Drucker - Ausgabe auf dem Bildschirm, wenn der parallel mitlaufende Drucker eingeschaltet ist und der Drucker dem Bildschirm zugeordnet wurde.

Auf dem MZ-3500 wurden die folgenden Geräte implementiert:

SCREEN	- der Bildschirm
KEYBD	- die Tastatur
RS232	- die eingebaute V24 / RS232 - Schnittstelle
CENTRO	- die eingebaute Centronics - Schnittstelle
CE332P	- der Drucker SHARP CE-332P
GRAPH	- der Graphik - Interpreter
DUMMY	- ein leeres Gerät, das alle Ausgaben aufnimmt und bei Eingabe den Wert 1AH (Control-Z, das Dateiendekennzeichen) zurückliefert.

Die Tabelle der Baud Rate - Codes für die RS232 - Schnittstelle:

0 - 50 Baud	8 - 1800 Baud
1 - 75 Baud	9 - 2000 Baud
2 - 110 Baud	10 - 2400 Baud
3 - 134 Baud	11 - 3600 Baud
4 - 150 Baud	12 - 4800 Baud
5 - 300 Baud	13 - 7200 Baud
6 - 600 Baud	14 - 9600 Baud
7 - 1200 Baud	15 - 19200 Baud

Falls das Bit 7 gesetzt ist, hat das eine Einstellung von 5 Datenbits und 1 1/2 Stopbits, sonst die Einstellung von 8 Datenbits und einem Stopbit zur Folge. Gerade Parität kann durch das Setzen von Bit 6 verlangt werden, ansonsten ist die Parität unterdrückt. Zum Setzen verwende man die EOS - Funktion 50 (siehe auch Abschnitt C, Funktion 21).

A.4. Der Datei - Versorgungsblock FCB

Alle Systemaufrufe, die Dateien betreffen, erwarten einen Pointer auf einen Datei - Versorgungsblock, im nachfolgenden FCB (File Control Block) genannt. Dieser FCB hat eine Länge von mindestens 33 Bytes, im Random - Mode 36 Bytes mit folgendem Aufbau:

Byte 1: Nummer des Laufwerks, auf dem sich die Datei befindet. Falls das aktuelle Laufwerk angesprochen werden soll, ist dieses Byte 0, ansonsten hat dieses Byte den Wert 1 für Laufwerk A, 2 für B usw. bis maximal 16 für Laufwerk P. Der Directory - Eintrag hat an dieser Stelle entweder den der Datei zugeordneten Benutzerbereich stehen oder aber das Byte DE5H, das einen gelöschten Eintrag kennzeichnet.

Bytes 2- 9: Dateiname. Falls der Name kürzer als 8 Zeichen ist, ist das Feld mit Leerzeichen zu füllen. Es werden bei der Suche im Directory nur die untersten 7 Bits interpretiert. Das achte Bit jedes Zeichens wird von EOS als Attributbit definiert. Diesen Bits sind teilweise bereits verschiedene Funktionen fest zugeordnet:

Byte 2 (f1'): frei für Anwender.

Byte 3 (f2'): frei für Anwender.

Byte 4 (f3'): frei für Anwender.

Byte 5 (f4'): diese Datei wird nach dem De-Spooling bzw. nach dem Abarbeiten von darin enthaltenen Kommandos gelöscht.

Byte 6 (f5'): Interface - Bit für Systemaufrufe.

Byte 7 (f6'): Interface - Bit für Systemaufrufe.

Byte 8 (f7'): für EOS reserviert.

Byte 9 (f8'): für EOS reserviert.

Bytes 10-12: Dateityp. Für die Interpretation gilt das gleiche wie oben angegeben. Allerdings ist das achte Bit jedes Zeichens wie folgt reserviert:

Byte 10 (t1'): Datei ist schreibgeschützt.

Byte 11 (t2'): System - Attribut. Diese Datei wird von den eingebauten Kommandos wie DIR, ERA usw. nicht erfaßt, sofern nicht ausdrücklich darum gebeten wurde. Zusätzlich kann diese Datei von einem anderen Benutzerbereich aus geöffnet werden, allerdings dann nur im Lesezugriff.

Byte 12 (f3'): Backup - Attribut. Dieses Bit kann von einem beliebigen Programm (meist einem Datensicherungsprogramm) gesetzt werden. Es wird von EOS nach jeder Änderung dieser Datei wieder gelöscht.

Die Bedeutung dieses Bits wurde im Gegensatz zu DiCOS umgekehrt, um CP/M - kompatibel zu bleiben.

- Byte 13: Extent - Nummer. Diese Extentnummer beginnt bei 0 für den 1. Satz und wird alle 128 Sätze um 1 hochgezählt. In diesem Byte werden 5 Bits der Extentnummer aufbewahrt. Weitere 6 Bits werden in Byte 15 aufbewahrt.
- Byte 14: Enthält im Directory den Byte Count. Wenn die Datei geöffnet ist, enthält dieses Byte EOS - Verwaltungsflags, die nicht verändert werden dürfen.
- Byte 15: Zweites Extentbyte. Die unteren 6 Bits stellen die oberen 6 Bits des Extents dar.
- Byte 16: Satzzahl des in Bytes 13 und 15 eingetragenen Extents. Immer kleiner als 129. Die Anzahl der Sätze in einer Datei ergibt sich also aus (Extentnummer * 128) + dieses Byte.
- Bytes 17-32: Allocation Map. Hier werden vom System die Nummern der Clusters eingetragen, die die Datei belegt. Wenn diese Liste überläuft, wird einfach ein neuer Eintrag im Directory eingerichtet.
- Byte 33: Aktuelle Blocknummer. Dieses Byte gibt die Nummer des Blocks innerhalb des aktuellen Extents an, der als nächstes zu bearbeiten ist. Beim Öffnen einer Datei werden die ersten 32 Bytes aus dem Directory in den FCB hineinkopiert. Byte 33 sollte also bei jedem Öffnen auf 0 gesetzt werden, da das System sonst bei der dort eingetragenen Satznummer im Extent 0 anfängt zu lesen bzw. zu schreiben!
- Bytes 34-36: Random - Blocknummer. Dies ist eine 3 Bytes lange Zahl, die für Random - Satzzugriffe benutzt wird. Die Ablage erfolgt in umgekehrter Reihenfolge, d.h. das niederwertigste Byte wird in Byte 34 und das höchstwertige Byte in Byte 36 abgelegt.

A.5. Sonderstrukturen im Directory

Um das Festhalten von Datum und Uhrzeit im Directory durchzuführen, wurden einige neue Datenstrukturen eingeführt. Diese umfassen erstens den Directory Label und zweitens die Zeiteinträge. Ein derartiger Eintrag hat immer eine Länge von 4 Bytes und ist wie unter den Systemfunktionen 104 und 105 beschrieben definiert:

Bytes 1-2 : Julianisches Datum, wobei eine 1 dem 1. 1. 1978 entspricht
Byte 3 : Stunde im BCD - Format
Byte 4 : Minute im BCD - Format

A.5.1. Der Directory Label

Die Steuerung der Zeiteinträge wird durch einen sogenannten Directory Label durchgeführt. Dieser Label besteht aus einem ganz normalen Directory - Eintrag im Benutzerbereich 32. Der Dateiname bzw. der Dateityp kann dazu benutzt werden, um dem Datenträger einen Namen zuzuweisen.

Im Einzelnen ist der Directory Label wie folgt aufgebaut:

Byte 1 : eine 32 als Kennzeichnung des Directory Labels.

Bytes 2-11 : der Name des Datenträgers.

Byte 12 : das Data Byte. Dieses ist ein Bitvektor mit folgender Bedeutung:

Bit 6 - Zeiteintrag bei Dateizugriffen.

Bit 5 - Zeiteintrag bei Dateiänderungen.

Bit 4 - Zeiteintrag beim Neueinrichten einer Datei.

Die restlichen Bits sind reserviert. Es kann entweder Bit 6 oder Bit 4 des Data Bytes gesetzt sein, da es nur ein Datenfeld für diese beiden Einträge gibt. Das Eintragen bei Dateizugriffen schließt also den Eintrag beim Neueinrichten aus, und umgekehrt.

Bytes 13-24 : reserviert.

Bytes 25-28 : Zeitpunkt der Einrichtung des Directory Labels.

Bytes 29-32 : Zeitpunkt der letzten Änderung des Directory Labels.

A.5.2. Die Zeiteinträge

Für die Zeiteinträge werden besondere Directory - Einträge benötigt. Diese werden nicht durch das System eingerichtet, sondern müssen mit Hilfe des Hilfsprogramms \$INITDIR in das Directory eingetragen werden. Diese Einträge nehmen dann jeweils den letzten von 4 Directory - Einträgen je 128 - Bytes - Block ein. Ein derartiger "Zeit - FCB" ist im User Byte durch eine 33 gekennzeichnet. Ab dem zweiten Byte sind dann jeweils 10 Bytes für je einen der drei anderen Einträge im jeweiligen 128 - Bytes - Block reserviert. Dies bedeutet, daß die ersten 10 Bytes (Bytes 2 bis 11) die Zeiteinträge des ersten Directory - Eintrags im Block aufnehmen, die zweiten 10 Bytes (Bytes 12 bis 21) die Einträge des zweiten Directory - Eintrags aufnehmen, und die letzten 10 Bytes (Bytes 22 bis 31) die Einträge des dritten Directory - Eintrags. Das letzte (das 32.) Bytes des Eintrags ist reserviert.

Ein derartiger 10 Bytes langer Zeiteintrag hat das folgende Aussehen:

Bytes 1-4 : Zeitpunkt der Neueinrichtung oder des letzten Zugriffs

Bytes 5-8 : Zeitpunkt der letzten Änderung

Byte 9 : Paßwort - Modus, unter EOS immer 0

Byte 10 : reserviert.

Es sei noch erwähnt, daß die Einrichtung eines Directory Labels abgelehnt wird, falls das Datenbyte die Protokollierung von Zeiten anfordert, aber keine "Zeit - FCB's" im Directory vorhanden sind.

A.6. Die Initialisierung der Zero Page

Beim Start eines Programms wird die Zero Page, d.h. der Adreßbereich von 0000H bis 00FFH, von EOS initialisiert. Es werden eventuell zusätzlich eingegebene Dateinamen in definierten Bereichen abgelegt. Der Rest der Kommandozeile wird ebenfalls zur Verfügung gestellt. Die Initialisierung der Zero Page unterscheidet sich in nichts von der Weise, in der CP/M plus die Initialisierung durchführt. Im Einzelnen gelten folgende Vereinbarungen:

- 000H - 002H : Sprungbefehl auf den Restartpunkt des Systems. Ein Sprung auf diesen Punkt beendet das laufende Programm und startet den Kommando - Interpreter neu.
- 003H : Das ehemalige IOByte. Wird von EOS auf den Wert 0 gesetzt.
- 004H : Das ehemalige User/Disk - Byte des CP/M - CCP. EOS besetzt die oberen 4 Bits mit dem aktuellen Benutzerbereich und die unteren 4 Bits mit dem aktuellen Laufwerk bei Programmstart.
- 005H - 007H : System - Sprungvektor. Zeigt auf den Start von EOS bzw. auf den Start des zuletzt geladenen Funktionsmoduls. Ein Sprung auf diese Adresse hat die Ausführung einer Systemfunktion zur Folge. Gleichzeitig gibt der Adreßteil des Sprungbefehls das erste durch EOS belegte Byte des Benutzerspeichers an.
- 008H - 03AH : Reserviert für Interrupt - Vektoren.
- 03BH - 04FH : Reserviert für EOS.
- 050H : Laufwerk, von wo das gestartete Programm geladen wurde. Eine 1 entspricht dem Laufwerk A:, eine 2 dem Laufwerk B: etc.
- 051H - 052H : Zeiger auf den Start des Paßworts der ersten Dateiangabe. Dieser Zeiger zeigt in den Kommandorest hinein, der sich ab der Adresse 80H befindet. Ist bei der ersten Dateiangabe kein Paßwort angegeben, wird eine 0 abgelegt.
- 053H : Länge des Paßworts der ersten Dateiangabe. Ist kein Paßwort angegeben, enthält dieses Byte den Wert 0.
- 054H - 055H : Zeiger auf den Start des Paßworts der zweiten Dateiangabe. Dieser Zeiger zeigt ebenfalls in den Kommandorest hinein, der sich ab der Adresse 80H befindet. Ist bei der zweiten Dateiangabe kein Paßwort angegeben, wird eine 0 abgelegt.
- 056H : Länge des Paßworts der zweiten Dateiangabe. Ist kein Paßwort angegeben, enthält dieses Byte den Wert 0.

057H - 05BH : Reserviert.

05CH - 06BH : FCB der ersten Dateiangabe. Dieser Bereich enthält die ersten 16 Bytes eines FCB, falls das erste Wort nach dem Kommandowort zu einem Dateinamen umfunktioniert werden konnte, andernfalls enthält die Adresse 05CH eine 0 und die Adressen 05DH bis 067H Leerstellen. Der Bereich 05BH bis 07FH kann vom Programm direkt als FCB genutzt werden; allerdings ist dann der zweite FCB, der sich ab 06CH befindet, vor Gebrauch erst zu retten.

06CH - 07BH : FCB der zweiten Dateiangabe. Falls das zweite Wort nach dem Kommandowort zu einem Dateinamen umfunktioniert werden konnte, sind die ersten 16 Bytes des dadurch entstandenen FCBs hier abgelegt. Falls nicht, enthält die Adresse 06CH den Wert 0 und die Adresse 06DH bis 077H Leerstellen.

07CH - 07FH : frei für Verwendung als FCB - Blockzähler.

080H : Start des DMA - Puffers. Enthält als Wert die Länge des folgenden Kommandorests.

081H - 0FFH : Kommandorest. Dies ist der Rest der Kommandozeile, beginnend unmittelbar hinter dem Kommandonamen. Im Normalfall beginnt dieser Zeilenrest also mit einer oder mehreren Leerstellen.

Beim Start eines Programms wird eine ganze Reihe von Systemvariablen vom Kommando - Interpreter auf bestimmte Werte gesetzt. Diese sind im einzelnen:

Multisector Count - 1
DMA - Adresse - 80H
Stringendezeichen - "\$"
Programmencode - 0 (nicht bei einem Start durch Funktion 47).
Console Mode - 0

Der Stackpointer zeigt auf einen Stackbereich von 32 Bytes, dessen oberster Wert mit 0 initialisiert wurde. Ein Return - Befehl des laufenden Programms hat im Gegensatz zu CP/M 2.2 keine Rückkehr in den Kommando - Interpreter, sondern demnach vielmehr einen Warmstart zur Folge.

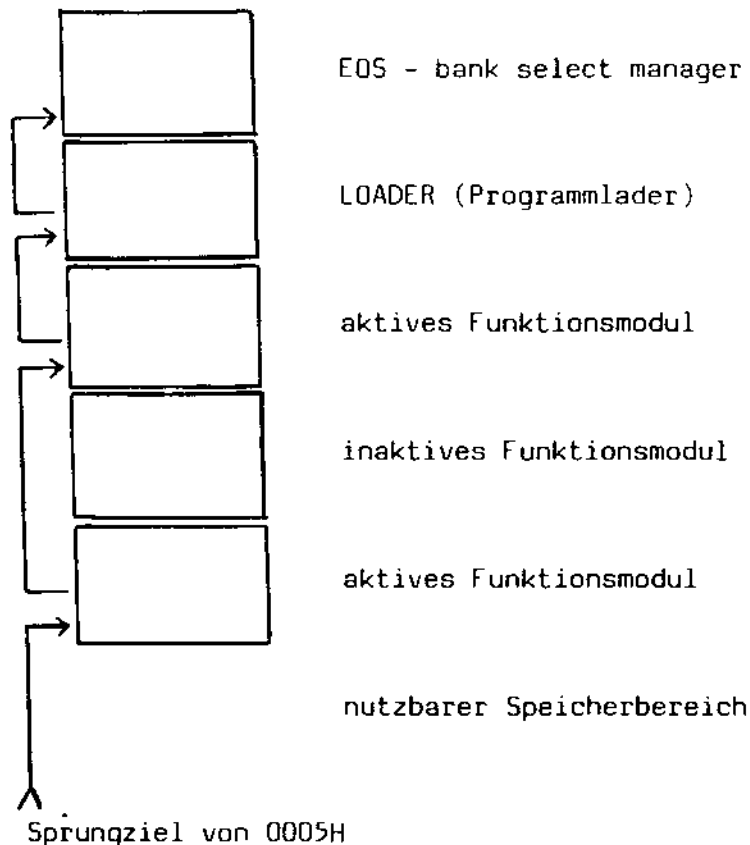
A.7. Funktionsmodule

Funktionsmodule sind spezifische Erweiterungen des EOS - Systemkerns. Diese Module werden von einem speziellen Programm an normale Programmdateien angebunden. Beim Laden eines Programms lädt dieser Lader alle angebundenen Funktionsmodule direkt unterhalb des EOS - Systemkerns (oder unterhalb aller anderen bereits geladenen Module). Anschließend wird der System - Sprungvektor auf Adresse 005H derart geändert, daß dieser Vektor in das soeben geladene Funktionsmodul zeigt. Dies hat zur Folge, daß alle Systemaufrufe durch dieses Modul hindurchgehen, so daß diese Aufrufe abgefangen bzw. abgeändert werden können. Zahlreiche Programme nutzen diese Möglichkeit, wie z. B. das Hintergrund - Druckprogramm, die Routinen zum Umdirigieren der Console und List I/O etc. Der Programmlader selbst ist ein derartiges Funktionsmodul. Er ist normalerweise nicht aktiv, kann aber durch das Laden von Funktionsmodulen im Speicher gehalten werden.

Ein Funktionsmodul verbleibt so lange im Speicher, bis es selbst ein spezielles Flag setzt, das signalisiert, daß das Modul seine Arbeit beendet hat. Der Kommando - Interpreter überprüft bei jedem Warmstart, ob irgendein Modul dieses Flag gesetzt hat. Findet er ein solches Modul, wird es aus der Systemaufruf - Kette herausgenommen. Liegt ein anderes Modul vor, d.h. unterhalb des entfernten Moduls, kann der Speicher so lange nicht freigegeben werden, bis alle anderen davor liegenden Module entfernt wurden.

A.7.1. Die Speicherorganisation nach dem Laden von Funktionsmodulen

Machen wir einmal ein Bild einer Situation, wo drei Module im Speicher liegen, davon eines inaktiv. Die Pfeile geben den Pfad an, dem ein Systemaufruf folgen muß:



Wie ersichtlich, führt der Pfad eines Systemaufrufs durch sämtliche Funktionsmodule hindurch. Werden einmal zwei gleichartige Module geladen, die beide den selben Systemaufruf abfangen, ergibt sich durch diese Konstruktion, daß der zuerst geladene so lange nicht angesprochen wird, bis der zuletzt geladene seine Funktion beendet hat. Dies macht sich beispielsweise der Datei - Executor zunutze, um verschachtelte Aufrufe von Kommandodateien zuzulassen.

A.7.2. Der Programm - Header

Wenn ein Programm Funktionsmodule enthält, bekommt es gleichzeitig einen 256 Bytes langen Datenbereich vorgeschaltet, in dem steht, wo die jeweiligen Module zu finden sind. Gleichzeitig kann dieser Header Informationen über zu ändernde Systemvariable enthalten, die vom Lader nach dem Ladevorgang entsprechend gesetzt werden. Dieser Header kann auch leer sein, was zur Folge hat, daß der Programmlader "überlistet" wird und sich nicht selbst aus dem Speicher entfernt. Er kann dann vom Programm aus mit Hilfe der Funktion 59 angesprochen werden. Andererseits kann ein Programm auch nur aus einem derartigen Header mit den entsprechenden Modulen bestehen. Ein Ladevorgang einer derartigen Datei hat zur Folge, daß der Kommando - Interpreter die Suche nach inaktiven Modulen beim nächsten Start unterdrückt, so daß auch Module geladen werden können, die bereits ihr Flag gesetzt haben. Diese Funktionen stehen dann dem nachfolgenden Programm zur Verfügung und werden erst nach dem Ende dieses Programms entfernt.

Um mit diesen Modulen zu kommunizieren, wurde der Systemaufruf 60 definiert. Dieser Aufruf hat verschiedene Parameter, die von allen geladenen Modulen überprüft werden können. Fühlt sich keines der geladenen Module angesprochen, liefert der EOS - Systemkern 0FFH in A und 00H in H zurück.

Und nun ein Blick auf diesen Header. Zuerst kommt ein 16 Bytes langer Verwaltungsbereich:

DB	0C9H	; Kennzeichnung als header
DW	CODESIZE	; Laenge des Programmcodes
DB	0C9H	; Sprung auf Parameter-Setz-Routine
DS	11	; reserviert
DB	NFUNC	; Anzahl der Funktionsmodule (0-15)

Nach diesem Verwaltungsbereich kommen bis zu 15 Einträge, einer pro angehängten Funktionsmodul:

DW	OFFSET	; Anfang des Moduls relativ zum ; Dateianfang
DW	CODESIZE	; Laenge des Moduls
DW	EXTRA	; benötigtes zusätzliches RAM
DB	'NAME	; Name des Moduls, 8 Bytes
DW	0	; reserviert

Fehlende Einträge sind mit Nullen aufgefüllt.

A.7.3. Die Struktur eines Funktionsmoduls

Ein Funktionsmodul muß drei Bedingungen erfüllen, um vom System anerkannt zu werden: Es muß im PRL - Format der Firma Digital Research vorliegen, es muß mit einem speziellen Header ausgerüstet sein, damit der Programmlader bzw. der Kommando - Interpreter darauf zugreifen können und der Dateityp muß nicht "PRL", sondern "RSX" heißen. Das PRL - Format kann am einfachsten mit Hilfe des Linkers der Firma Digital Research erzeugt werden; seine Anschaffung sei empfohlen. Eine derartige Datei hat ebenfalls einen speziellen Header, der sich wie folgt zusammensetzt:

DB	0	; reserviert
DW	SIZE	; Länge des Codes
DB	0	; reserviert
DW	EXTRA	; benötigter zusätzlicher Speicher
DS	250	; reserviert

Der Code beginnt demnach beim Dateianfang + 0100H. Dieser Code startet immer bei der Adresse 0100H, d.h. daß beim Assemblieren des Programms eventuell der Befehl "ORG 100H" vorangehen muß. Hinter diesem Code liegt eine Bittabelle. Für jedes Byte des Codes gibt ein Bit an, ob es an die tatsächliche Ladeadresse angepaßt werden muß. Das erste Bit (Bit 7) des ersten Bytes zeigt diese Korrektur für das erste Byte des Code an, Bit 6 für das zweite Byte etc. Die Startadresse dieser Tabelle ergibt sich aus

Dateianfang + 0100H + SIZE (der Länge des Codes).

Das Funktionsmodul selbst muß mit der folgenden Datenstruktur beginnen:

	DB	0,0,0,0,0,0	; reserviert
START:	JP	HINEIN	; Sprung in das Modul
HERAUS:	JP	0	; Sprung aus dem Modul heraus
LASTP:	DW	LAST	; Adresse des vorangehenden Moduls
FLAG:	DB	0	; Flag, um Inaktivität anzuzeigen
	DB	0	; reserviert
NAME:	DB	'NAME	; Name des Moduls
	DB	0,0,0	; reserviert

..

Die Bedeutung der Labels im Einzelnen:

- START: Dies ist der Einsprung in das Modul. Der Programmlader ändert den System - Sprungvektor auf Adresse 005H derart, daß er auf diese Adresse zeigt.
- HERAUS: Dies ist der Ausgang aus dem Modul. Falls ein Systemaufruf weitergereicht werden soll, ist das Modul durch einen Sprung auf diese Adresse zu verlassen. Der Programmlader ändert diesen Sprungbefehl derart, daß er auf das nächste Modul bzw. auf den Programmlader zeigt. Der Vektor im Lader selbst zeigt auf den EOS - Systemkern.
- LASTP: Zeiger auf das vorhergehende Modul in der Reihenfolge. Das zuletzt geladene Modul enthält hier den Wert 005H.
- FLAG: Falls diese Adresse einen Wert ungleich 0 annimmt, wird das Modul beim nächsten Warmstart aus dem Speicher entfernt.
- NAME: Name des Moduls, acht Zeichen lang.

Beim Programmieren des Moduls muß streng auf Stack - Disziplin geachtet werden, da das rufende Programm u. U. keinen Stackbereich zur Verfügung stellt.

A.7.4. Beispiel eines Funktionsmoduls

Als Beispiel sei hier ein Modul aufgezeigt, das beim Aufruf 1 (lies von der Tastatur) für jedes Line Feed - Zeichen einen Wagenrücklauf zurückliefert. Zusätzlich soll ein Control - C dieses Modul deaktivieren:

```

      ORG      100H                ; Start bei Adresse 0100H
      DB      0,0,0,0,0,0        ; reserviert
      JP      TEST                ; Einsprung ins Modul
BDOS:  JP      0                  ; Verlassen des Moduls
      DW      0,0                ; Platz für Zeiger
FLAG:  DB      0                ; nicht entfernen
      DB      0                  ; reserviert
      DB      'LF - CR          ; Name des Moduls
      DB      0,0,0              ; reserviert

TEST:  LD      A,C                ; Prüfen des Funktionsaufrufs
      CP      1                  ; ist es der gesuchte?
      JR      NZ,BDOS            ; nein, durchreichen
      LD      (SPSAVE),SP        ; sonst neuen Stack einrichten
      LD      SP,STACK           ; neu setzen
      CALL   BDOS                ; BDOS aufrufen
      LD      SP,(SPSAVE)        ; alten Stack wieder herstellen
      CP      0AH                ; Line Feed?
      JR      Z,LF               ; ja, verzweigen
      CP      03H                ; Control-C?
      RET     NZ                 ; nein, Zeichen zurück
      LD      OFFH               ; sonst das Flag
      LD      (FLAG),A           ; aktivieren
      JP      0                  ; und Warmstart durchführen
LF:    LD      A,0DH              ; bei LF ein CR zurück
      RET

SPSAVE: DW      0                ; Platz für alten SP
      DS      4                  ; Hilfs - Stack

STACK  EQU      $                ; Top of Stack
      END

```

--

A.8. Der System - Variablenbereich

Alle Variablen des Systemkerns wurden zusammengefaßt und in einen speziellen Variablenbereich abgelegt. Dieser Variablenbereich wird im Common Memory abgelegt (beim MZ-3500 liegt dieser Bereich ab 0FF00H). Die darin enthaltenen Werte können gelesen bzw. geändert werden. Zum Zugriff auf den Variablenbereich sollte man die Funktion 49 verwenden, da die Anfangsadresse des Variablenbereichs implementationsabhängig ist.

Die folgende Tabelle enthält 5 Spalten: Die erste Spalte enthält den Offset zum Anfang des Variablenbereichs, die zweite den Namen der Variable. Die dritte Spalte gibt an, inwieweit diese Variable der entsprechenden Variable unter CP/M plus entspricht. Die vierte Spalte informiert darüber, ob diese Variable geändert oder nur gelesen werden darf.

Offset	Name	CP/M	Ändern	Funktion
00H-04H			nein	reserviert.
05H	EOSVERS	ja	ja	enthält die Versionsnummer für die Funktionsnummer 12. Kann auch durch Funktion 72 geändert werden.
06H-09H	USERFLG	ja	ja	vier Bytes, frei für eigene Anwendungen.
0AH-0FH			nein	reserviert.
10H-11H	RETCODE	ja	ja	Programm-Rückkehrcode für Funktion 108.
12H-17H			nein	reserviert.
18H-19H	FILENR	nein	ja	eine laufende Nummer, die von den Hilfsprogrammen für die Generierung von eindeutigen Dateinamen benutzt wird ("SYSnnnnn.\$\$\$").
1AH	CWIDTH	ja	ja	Zeichen pro Bildschirmzeile; wird von EOS nicht benutzt.
1BH			nein	reserviert.
1CH	CLENGTH	ja	ja	Zeilenzahl des Bildschirms; wird von EOS nicht verwendet.
1DH	TABSTOP	nein	ja	aktueller Tab Stop.
1EH-1FH			nein	reserviert.

Offset	Name	CP/M	Ändern	Funktion
20H-21H	FXDVEC	nein	nein	Fixed media login vector. Alle Laufwerke, die sich selbst als fest installiert zu erkennen geben, (siehe Beschreibung des DPB), setzen in diesem Vektor ein Login - Bit beim ersten Zugriff. Nach einem Disk Reset (EOS - Funktion 13) wird der Login - Vektor mit dem Wert dieses Vektors besetzt.
22H-23H	CIVEC	ja	ja	Bitvektor für das Gerät CONIN:
24H-25H	COVEC	ja	ja	Bitvektor für das Gerät CONOUT:
26H-27H	AIVEC	ja	ja	Bitvektor für das Gerät AUXIN:
28H-29H	AOVEC	ja	ja	Bitvektor für das Gerät AUXOUT:
2AH-2BH	LOVEC	ja	ja	Bitvektor für den Drucker LSTOUT:
2CH			nein	reserviert.
2DH	PRTIME	nein	ja	Time Out - Konstante für die Ausgabegeräte. Kann durch die Funktion 73 gesetzt werden.
2EH	DQLVL	nein	nein	Tiefe der aktuellen Verschachtelung von \$DQ - Ausführungen.
2FH-32H			nein	reserviert.
33H-34H	CMODE	ja	ja	Console Mode - Vektor. Kann durch die Funktion 109 gesetzt bzw. abgefragt werden.
35H-36H	BNKBUF	ja	nein	Adresse des EOS - Arbeitspuffers. Dieser Puffer hat eine Länge von 128 Bytes und kann für eigene Zwecke genutzt werden, solange kein EOS - Aufruf benutzt wird.
37H	PRTEND	ja	ja	String Output Delimiter für Funktion 9. Dieser Wert kann durch die Funktion 110 abgefragt bzw. geändert werden.
39H-3BH	SERNR	nein	nein	aktuelle Seriennummer des Systemkerns.
3CH-3DH	CURDMA	ja	nein	aktuelle DMA - Adresse. Kann durch Funktion 26 gesetzt werden.
3EH	CURDSK	ja	nein	aktuelles Laufwerk. Kann durch Funktion 14 gesetzt werden.

Offset	Name	CP/M	Ändern	Funktion
3FH-40H	PARAM	ja	nein	Parameter (Inhalt von DE) der letzten EOS - Funktion.
41H			nein	reserviert.
42H	BRKCH	nein	ja	aktuelles zweites Stopzeichen. Kann durch die Funktion 75 neu gesetzt werden.
43H	FUNC	ja	nein	Nummer der letzten EOS - Funktion.
44H	CURUSR	ja	nein	Aktueller Benutzerbereich. Kann durch die Funktion 32 gesetzt bzw. abgefragt werden.
45H	DSPDRV	nein	ja	Das Laufwerk, das in der Statuszeile angezeigt wird.
46H-49H			nein	reserviert.
4AH	SECCNT	ja	ja	Aktueller Multisector Count. Kann durch die Funktion 44 gesetzt werden.
4BH	?ERROR	ja	ja	Aktueller Fehlermodus. Kann durch die Funktion 45 gesetzt werden.
4CH-4FH	DRVCHN	ja	ja	Suchfolge für Kommandos. Wird vom Kommando - Interpreter benutzt. Es sind maximal 4 Werte möglich, wobei eine 1 dem Laufwerk A, eine 2 dem Laufwerk B etc. entspricht. Eine 0 bezeichnet das aktuelle Laufwerk. Der Tabellenrest wird mit Nullen aufgefüllt.
50H	TMPDRV	ja	ja	Laufwerk, das für temporäre Dateien benutzt werden soll. Diese Variable wird u. A. vom Spooler und von \$DO.COM ausgewertet. Eine 0 entspricht dem aktuellen Laufwerk, eine 1 dem Laufwerk A, eine 2 dem Laufwerk B etc.
51H	ERRDRV	ja	nein	Laufwerk, auf dem ein aktueller physikalischer Fehler auftrat.
52H-54H			nein	reserviert.
55H-56H	FILENR	nein	nein	Enthält eine Nummer, die von \$SPOOL und \$DO zum Generieren von eindeutigen Dateinamen für temporäre Dateien verwendet wird. Die Nummer wird nach jeder Generierung eines Namens um 1 hochgezählt.

Offset	Name	CP/M	Ändern	Funktion
57H	EOSFLAG	ja	nein	Aktuelle EOS - Flags. Die Flag Bits können durch die Funktion 76 neu gesetzt werden. Eine Erklärung der einzelnen Bits ist im Kapitel A.9 nachzulesen.
58H-59H	DATE	ja	nein	Das aktuelle Tagesdatum im Julianischen Format, wobei eine 1 dem 1. Januar 1978 entspricht. Die Datum/Uhrzeit-Variablen können durch die Funktionen 104 und 105 gesetzt bzw. abgefragt werden.
5AH	HOURS	ja	nein	Die aktuelle Stunde im BCD - Format.
5BH	MINUTES	ja	nein	Die aktuelle Minute im BCD - Format.
5CH	SECONDS	ja	nein	Die aktuelle Sekunde im BCD - Format.
5DH-61H			nein	reserviert.
62H-63H	MXTPA	ja	nein	Top of Memory. Wird durch den Programmlader auf den aktuellen Stand gebracht.
64H-65H			nein	reserviert.
66H	DIRLBL	nein	nein	Datenbyte des aktuellen Directory Labels. Kann durch die Funktion 101 abgefragt werden.
67H-69H			nein	reserviert.
6AH-6BH	PH\$CSV	nein	nein	Adresse des Checksum Vectors des aktuellen Laufwerks.
6CH-6DH	PH\$ALV	nein	nein	Adresse des Allocation Vectors des aktuellen Laufwerks. Kann durch die Funktion 27 abgefragt werden.
6EH-7FH	DPB	nein	nein	Disk Parameter Block des aktuellen Laufwerks. Diese Adresse wird beim Funktionsaufruf 31 zurückgeliefert.

Der Bereich von 6AH bis 7FH ist nach einem Systemaufruf 13 (Platten normieren) oder 14 (Laufwerk anwählen) undefiniert. Er wird erst aktiviert, wenn tatsächlich auf die Platte zugegriffen wird. Es wird daher geraten, diese Werte über die dazu zur Verfügung stehenden EOS - Funktionen abzufragen.

A.9. Die EOS - Flags

Im System - Variablenbereich von EOS ist ein Flag Byte abgelegt. Es ist im Kapitel A.8 unter dem Namen "EOSFLAG" und dem Offset 57H zu finden. Dieses Flag Byte enthält verschiedene Flags, die durch den Benutzer änderbar sind und die das globale Verhalten von EOS beeinflussen, d.h. daß diese Bits nicht durch einen Systemrestart gelöscht werden. Diese Bits sind:

- Bit 7 : reserviert zwecks Kompatibilität zu CP/M plus.
- Bit 6 : reserviert zwecks Kompatibilität zu CP/M plus.
- Bit 5 : das Setzen dieses Bits bewirkt die Aktivierung des Line Editors. Wenn die EOS - Funktion 10 zwecks Einlesen einer Zeile aufgerufen wird, wird diese Zeile über den Bildschirm - Editor Ihres Rechners eingelesen. Im Normalfall wird zur Eingabe der Zeile der Mini - Editor, der in EOS fest eingebaut ist, herangezogen.
- Bit 4 : Wenn dieses Bit auf 1 gesetzt wird, erfolgt bei der Eingabe eines Stopzeichens kein Programmabbruch; vielmehr wird die Unterbrechung am Bildschirm angezeigt und der Bediener aufgefordert, dazu Stellung zu nehmen.
- Bit 3 : Um eine generelle Abbruchmöglichkeit zu schaffen, wurde dieses Bit eingeführt. Wenn Sie dieses Bit setzen, erfolgt bei jedem Einsprung in EOS eine Abfrage auf Stopzeichen. Wenn eines erkannt wurde, erfolgt die Anzeige der Adresse, wo der Systemaufruf stattfand zusammen mit einer Interruptmeldung, auf die der Benutzer wiederum unterschiedlich reagieren kann.

Es sei nicht verschwiegen, daß diese Option auch Nachteile mit sich bringt: der Durchsatz des System wird verringert, das Type Ahead funktioniert nicht mehr und manche Programme, die ein Stopzeichen als reguläre Eingabe erwarten, laufen nicht mehr einwandfrei. Dieses Feature ist nur als Testhilfe gedacht.

Die restlichen drei Bits sind für internen Gebrauch reserviert. Die Bits 6 und 7 können von Benutzerprogrammen frei verwendet werden, allerdings mit der Einschränkung, daß diese Programme dann nicht mehr CP/M - kompatibel sind. Es wird für solche Zwecke empfohlen, sich der User Flags im Systemvariablenbereich zu bedienen.

A.10. Systemfehler

Alle Systemaufrufe, die auf Platten zugreifen, können Fehler zur Folge haben. Diese Fehler werden in logische und in physikalische Fehler unterteilt. Im Gegensatz zu den physikalischen Fehlern beruhen logische Fehler auf dem Fehlverhalten des Programms bzw. des Systems. Ein typischer logischer Fehler wäre beispielsweise das Lesen einer Datei hinter dem Dateiende.

Logische Fehler werden im Register A zurückgemeldet. Sie können die folgenden Werte annehmen:

- 1 - Daten fehlen. Dieser Fehler tritt nur beim Lesen einer Datei auf, und zwar, wenn versucht wurde, hinter dem Dateiende oder in einer durch Random - Zugriffe entstandenen Lücke Daten zu lesen. Desweiteren zeigt dieser Fehler beim sequentiellen Schreiben an, daß das Directory voll ist.
- 2 - Platte voll. Tritt nur beim Schreiben auf.
- 3 - Beim Wechsel des aktuellen Extents ist ein Fehler aufgetreten. Dieser Fehler tritt eigentlich nur dann auf, wenn die Directory - Struktur zerstört ist.
- 4 - Positionierungsversuch auf nicht existierendes Extent. Tritt bei Random - Lesezugriffen auf. Er ist im Grunde nur eine Erweiterung des Fehlers 1.
- 5 - Directory voll. Es kann kein neues Extent mehr eingerichtet werden.
- 6 - Die angegebene Random - Blocknummer überschreitet die zulässige Maximalgröße (zur Zeit beträgt diese 65535). Der Fehler kann auch bei dem Versuch auftreten, eine Datei zu erzeugen, die größer als 8 Megabytes ist.

Die möglichen physikalischen Fehler haben im Normalmodus eine Fehlermeldung auf dem Bildschirm zur Folge. Diese sieht wie folgt aus:

EOS-Systemfehler in Funktion n, Datei d:nnnnnnnn.eee
Meldung

oder:

EOS-Systemfehler in Funktion n, Laufwerk d:
Meldung

wobei n die Funktionsnummer angibt, bei der der Fehler auftrat, d: das Laufwerk bezeichnet, wo der Fehler auftrat und nnnnnnnn.eee die Datei, die unter dem Fehler leiden muß, bezeichnet. Die Fehlermeldungen selbst gliedern sich in die folgenden Möglichkeiten:

- 1 - Schreib/Lesefehler. Beim Schreiben oder Lesen eines Datenträgers ist ein unkorrigierbarer Fehler aufgetreten.
- 2 - Laufwerk ist schreibgeschützt. Passiert entweder durch einen vorzeitigen Plattenwechsel oder durch einen Aufruf von Funktion 28.
- 3 - Datei ist schreibgeschützt. Dieser Fehler tritt bei einem Schreibversuch auf eine schreibgeschützte Datei auf. Dieser Schreibschutz kann bei zwei verschiedenen Gelegenheiten auftreten:
 1. Wenn die Datei im Namen das entsprechende Attributbit gesetzt hat.
 2. Wenn die Datei im Benutzerbereich 0 liegt, aber von einem anderen Benutzerbereich aus geöffnet wurde.
- 4 - Laufwerk nicht anwählbar. Tritt auf, falls das Laufwerk, auf das zugegriffen werden sollte, nicht existiert.
- 5-7 - reserviert.
- 8 - Datei existiert bereits. Tritt bei den Funktionen 22 (Datei einrichten) und 23 (Datei umbenennen) auf.
- 9 - Dateiname enthält "?". Es gibt eine ganze Reihe von Funktionen, die einen eindeutigen Dateinamen verlangen und mehrdeutige Namen mit diesem Fehler zurückweisen.
- 10 - reserviert.
- 11 - Ein Stopzeichen wurde erkannt.

Physikalische Fehler werden immer im H - Register zurückgeliefert. Das A - Register enthält dabei den Wert OFFH. Der Wert OFFH kann allerdings auch bei logischen Fehlern auftreten. In diesem Falle ist H gleich 0. Die physikalischen Fehler können unter EOS unterschiedlich behandelt werden. Die Art der Behandlung kann mit Hilfe der Funktion 45 eingestellt werden. Es gibt drei Möglichkeiten:

1. Der Fehler wird dem rufenden Programm gemeldet.
2. Der Fehler wird angezeigt und dem rufenden Programm gemeldet.
3. Der Fehler wird angezeigt. Der Benutzer hat die Möglichkeit, angemessen auf den Fehler zu reagieren. Entscheidet er sich für das Weitermachen, wird wiederum der Fehlercode an das rufende Programm gemeldet. Diese Möglichkeiten sind:
 - A. Den Vorgang wiederholen.
 - B. Das laufende Programm abbrechen.
 - C. Den Fehler ignorieren.
 - D. Den Fehler dem Programm zurückmelden.

A.11. Liste der verfügbaren Systemaufrufe

Nr	Name	Versorgung	Ergebnis
0	Programmende	--	keine Rückkehr
1	Zeichen einlesen	--	A = Zeichen
2	Zeichen ausgeben	E = Zeichen	--
3	Zusatzgerät lesen	--	A = Zeichen
4	Zusatzgerät schreiben	E = Zeichen	--
5	Zeichen drucken	E = Zeichen	--
6	Terminal - I/O	E = Zeichen/Code	A = Zeichen, falls E=Code
7	Zusatzg. Eing-Status	--	A = 0 oder 1
8	Zusatzg. Ausg-Status	--	A = 0 oder 1
9	Text ausgeben	DE ^ Text	--
10	Zeile einlesen	DE ^ Zeilenpuffer	Zeile im Puffer
11	Terminalstatus	--	A = 0 oder 1
12	Versionsnummer	--	HL = 0031H
13	Disk Reset	--	AH = Fehlercode
14	Laufwerksanwahl	E = Laufwerk	--
15	Datei öffnen	DE ^ FCB	AH = Fehlercode
16	Datei schließen	DE ^ FCB	AH = Fehlercode
17	Datei suchen	DE ^ FCB	AH = Fehlercode
18	Nächster Eintrag	--	AH = Fehlercode
19	Datei löschen	DE ^ FCB	AH = Fehlercode
20	Block lesen	DE ^ FCB	AH = Fehlercode
21	Block schreiben	DE ^ FCB	AH = Fehlercode
22	Datei einrichten	DE ^ FCB	AH = Fehlercode
23	Datei umbenennen	DE ^ FCB	AH = Fehlercode
24	Loginvektor	--	HL = Bitvektor
25	Laufwerk erfragen	--	A = aktuelles Laufwerk
26	DMA-Adresse setzen	DE = DMA-Adresse	--
27	Allocvektor	--	HL = Adresse oder OFFFHH
28	Schreibschutz	--	--
29	R/O - Vektor	--	HL = Bitvektor
30	Dateiattribute	DE ^ FCB	AH = Fehlercode
31	DPB erfragen	--	HL ^ DPB oder = OFFFHH
32	Usercode setzen	E = User / OFFH	A = User, falls E = OFFH
33	Lesen Random	DE ^ Random-FCB	AH = Fehlercode
34	Schreiben Random	DE ^ Random-FCB	AH = Fehlercode
35	Größe berechnen	DE ^ Random-FCB	R0-R2 mit Blocknummer
36	Blocknummer setzen	DE ^ Random-FCB	R0-R2 mit Blocknummer
37	Laufwerke normieren	DE = Bitvektor	A = 0
38	Laufwerke sperren	DE = Bitvektor	--
39	Laufwerke freigeben	DE = Bitvektor	--
40	Schreiben Random	DE ^ Random-FCB	AH = Fehlercode

Nr	Name	Versorgung	Ergebnis
41	Schreiben und Testen	DE ^ FCB	AH = 0
42	Block sperren	DE ^ FCB	AH = 0
43	Block freigeben	DE ^ FCB	AH = 0
44	Multisector Count	E = Anzahl Blöcke	--
45	Fehlerbehandlung	E = 0/OFEH/OFFH	--
46	Plattenbelegung	E = Laufwerk	Freier Platz im DMA-Buffer
47	Program Chain	E = Chain FLAG	keine Rückkehr
48	Puffer schreiben	--	AH = Fehlercode
49	Systemvariable	DE ^ Vektor	HL = Wert
50	Hardware - Aufruf	DE ^ Parameter	HL = Rückkehrwert
59	Programm laden	DE ^ FCB	AH = Fehlercode
60	Funktionsmodul rufen	DE ^ Parameter	HL = Fehlercode
64	Fehlerbehandlung	E = 0 oder OFFH	A = alter Code
65	Prozeß wechseln	E = Prozeßnummer	--
66	Dump	DE ^ Dumpvektor	--
67	Stoptest	E = Zeichen/OFFH	A = 0 oder OFFH
68	Datei ausführen	DE ^ FCB	AH = Fehlercode
69	Datei drucken	DE ^ FCB	AH = Fehlercode
70	Drucksteuerung	E = Steuercode	A = 0 oder OFFH
71	Laufwerksbereitschaft	E = Laufwerk	HL = Bereit - Code
72	Version setzen	E = Versionsnr.	A = Versionsnummer
73	Time Out setzen	E = Zeitkonstante	A = Time Out
74	Tab Stop setzen	E = Tab Stop	A = Tab Stop
75	Stopzeichen setzen	E = Stopzeichen/0	A = Stopzeichen
76	EOS - Flags setzen	E = Flag Byte	A = Flag Byte
98	Datenpuffer freigeben	--	AH = 0
99	Datei kürzen	DE ^ FCB	AH = Fehlercode
100	Dir Label schreiben	DE ^ Dir Label	AH = Fehlercode
101	Dir Label Data	E = Laufwerk	AH = Fehlercode / Data
102	Zeiteinträge lesen	DE ^ FCB	AH = Fehlercode
103	XFCB schreiben	DE ^ XFCB	A = OFFH
104	Datum setzen	DE ^ Datum	--
105	Datum lesen	DE ^ Datum	A = Sekunden
106	Paßwort setzen	DE ^ Paßwort	--
107	Seriennummer lesen	DE ^ Zielfeld	Zielfeld gefüllt
108	Programmcode setzen	DE = Code / OFFFFFH	HL=Code, falls DE=OFFFFFH
109	Console Mode	DE = Mode / OFFFFFH	HL=Mode, falls DE=OFFFFFH
110	Stringende setzen	E = Char / OFFFFFH	A =Char, falls DE=OFFFFFH
111	Block ausgeben	DE ^ Blockparameter	--
112	Block drucken	DE ^ Blockparameter	--
115	Grafik aufrufen	DE ^ Parameter	HL = Fehlercode
152	FCB aufbauen	DE ^ Parameter	HL = Stringende / 0

Diese Seite wurde für Ihre Notizen freigelassen.

```

*****
*
*           0 - Beenden eines Programms
*
*****

```

Versorgung: C = 00H
DE = ---

Rückkehr: kehrt nicht zurück

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Das rufende Programm wird beendet. Der Warmstartpunkt auf Adresse 0000H wird angesprungen. Dieser Aufruf hat den gleichen Effekt wie ein Direktsprung auf die Adresse 0000H.

Ein eventuell gesetzter Programmfehlercode (siehe Funktion 108) wird an den Kommandointerpreter übergeben.

..


```

*****
*
*           1 - Lesen eines Zeichens vom Bildschirm
*
*****

```

Versorgung: C = 01H
 DE = ---

Rückkehr: A = eingelesenes Zeichen

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Die Eingabe von Ctrl-S, Ctrl-Q und Ctrl-P war nur während der Bildschirmausgabe möglich.

Das nächste anstehende Zeichen wird von der Tastatur eingelesen. Falls dieses Zeichen ein druckbares Zeichen oder eines der ASCII - Steuercodes CR, LF oder BS ist, wird dieses Zeichen wieder auf dem Bildschirm ausgegeben. Das ASCII - Steuerzeichen HT (Tabulatorsprung) führt zum Sprung der Eingabeposition auf den vorgegebenen Tabulatorwert. Falls jedoch durch den Aufruf 109 die einfache Bildschirmausgabe angewählt wurde, wird die Ausgabe des Tabulatorsprungs unterdrückt.

Die Bildschirmausgabe kann durch Eingabe des ASCII - Steuerzeichens DC3 (Ctrl-S) angehalten werden, falls diese Möglichkeit nicht durch den Aufruf 109 blockiert wurde. In diesem Fall wird DC3 an das rufende Programm zurückgegeben.

Wenn die Bildschirmausgabe durch Eingabe von DC3 (Ctrl-S) angehalten wurde, sind folgende Eingaben möglich:

- | | |
|--------------|---|
| DCI (Ctrl-Q) | |
| DC3 (Ctrl-S) | - Bildschirmausgabe wird fortgeführt. |
| DLE (Ctrl-P) | - der Drucker wird parallel zum Bildschirm ein- bzw. ausgeschaltet. Diese Eingabe hat die Wirkung eines Ein/Ausschalters. Falls durch den Aufruf 109 die einfache Ausgabe gewählt wurde, ist das Ein- bzw. Ausschalten des mitlaufenden Druckers nicht möglich. |
| Stopzeichen | - Das Programm wird abgebrochen. Auch diese Eingabe kann durch die Funktion 109 unterdrückt werden. |

```
*****
*
*           2 - Ausgabe eines Zeichens auf dem Bildschirm
*
*****
```

Versorgung: C = 02H
 E = auszugebendes Zeichen

Rückkehr: ---

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Das in Register E übergebene Zeichen wird auf dem Bildschirm ausgegeben. Falls der Drucker parallel eingeschaltet ist, wird das Zeichen ebenfalls auf diesem ausgegeben. Die unter Funktion 1 erwähnten Bildschirm - Steuermöglichkeiten der Ausgabe gelten hier ebenfalls.

```

*****
*
*           3 - Zeichen von Zusatzgerät einlesen
*
*****

```

Versorgung: C = 03H
 DE = ---

Rückkehr: A = eingelesenes Zeichen

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Das Zusatzgerät war bislang in die Geräte RDR: (Leser) und PUN: (Stan-
 zer) unterteilt.

Es wird ein Zeichen vom Zusatzgerät eingelesen. Normalerweise ist
 dieses Zusatzgerät fest auf die serielle Schnittstelle eingestellt.
 Diese Definition kann jedoch beliebig geändert werden.

```
*****
*
*           4 - Zeichen auf Zusatzgerät ausgeben
*
*****
```

Versorgung: C = 04H
 E = auszugebendes Zeichen

Rückkehr: ---

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Das Zusatzgerät war bislang in die Geräte RDR: (Leser) und PUN: (Stan-
zer) unterteilt.

Das in Register E enthaltene Zeichen wird auf dem Zusatzgerät ausge-
geben. Normalerweise ist dies die serielle Schnittstelle. Diese Defini-
tion kann jedoch beliebig geändert werden.

```
*****
*
*           5 - Zeichen auf Drucker ausgeben
*
*****
```

Versorgung: C = 05H
 E = auszugebendes Zeichen

Rückkehr: ---

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Das in Register E übergebene Zeichen wird auf dem Drucker ausgegeben. Falls der Drucker nicht bereit ist, wird nach einem durch die Funktion 73 einstellbaren Time Out eine Meldung ausgegeben. Wenn zu diesem Zeitpunkt eine Taste gedrückt wurde, wird die Druckausgabe abgebrochen. Falls der Drucker parallel zum Bildschirm mitlaufen sollte, wird dieser Modus wieder gelöscht.

```
*****
*
*           6 - Direkte Bildschirm - Ein/Ausgabe
*
*****
```

Versorgung: C = 06H
 E = Ausgabesteuerung

Rückkehr: A = Status oder Zeichen, falls angefordert

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Die Parameter E = 0FEH und E = 0FDH wurden neu eingeführt.

Diese Funktion dient der direkten Bildschirmsteuerung. In Register E ist ein Steuercode abgelegt, der die folgenden Werte annehmen kann:

0FFH - Tastatur prüfen, ob ein Zeichen eingegeben wurde. Falls kein Zeichen ansteht, wird das A - Register auf 0 gesetzt, andernfalls enthält A das eingegebene Zeichen.

0FEH - Status der Tastatur prüfen. Falls ein Zeichen zum Einlesen ansteht, wurde, enthält das Register A den Wert 0FFH, ansonsten 0H.

0FDH - Zeichen von der Tastatur einlesen. Es wird solange gewartet, bis ein Zeichen eingegeben wurde. Dieses Zeichen wird in A zurückgeliefert.

andere - alle anderen Werte werden als auszugebende Zeichen interpretiert und ausgegeben.

Da dieser Aufruf eine reine Ein/Ausgabefunktion ist, sind alle unter Funktion 1 aufgeführten Steuermöglichkeiten außer Betrieb.

```
*****  
*                                     *  
*               7 - Eingabestatus des Zusatzgeräts prüfen               *  
*                                     *  
*****
```

Versorgung: C = 07H
 DE = ---

Rückkehr: A = Status

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Bislang wurde diese Funktion zum Lesen des IOBytes auf Adresse 003H benutzt. Da sich die Philosophie der Gerätezuweisung unter EOS 3.0 entscheidend geändert hat, wird das IOByte nicht mehr unterstützt und diese Funktion somit frei.

Das Zusatzgerät wird überprüft, ob ein Zeichen eingegeben wurde. Ist dies der Fall enthält A den Wert 01H, andernfalls den Wert 00H.

Im Normalfall ist das Zusatzgerät als serielle Schnittstelle definiert. Diese Definition kann jedoch beliebig geändert werden.


```

*****
*
*           8 - Ausgabestatus des Zusatzgeräts prüfen
*
*****

```

Versorgung: C = 08H
DE = ---

Rückkehr: A = Status

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Bislang wurde diese Funktion zum Setzen des IOBytes auf Adresse 003H benutzt. Da sich die Philosophie der Gerätezuweisung unter EOS 3.0 entscheidend geändert hat, wird das IOByte nicht mehr unterstützt und diese Funktion somit frei.

Das Zusatzgerät wird überprüft, inwieweit es zum Senden eines Zeichens bereit ist. Falls ein Zeichen gesendet werden kann, enthält A den Wert 01H, andernfalls den Wert 00H.

Im Normalfall ist das Zusatzgerät als serielle Schnittstelle definiert. Diese Definition kann jedoch beliebig geändert werden.

```

*****
*
*           9 - Text ausgeben
*
*****

```

Versorgung: C = 09H
 DE = Zeiger auf den auszugebenden Text

Rückkehr: ---

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Das Stringendezeichen kann mittels der Funktion 110 beliebig undefiniert werden.

DE zeigt auf die Startadresse eines Textstrings. Dieser Text wird auf dem Bildschirm ausgegeben. Die in Funktion 1 erwähnten Steuermöglichkeiten sind in Betrieb, falls sie nicht durch den Aufruf 109 modifiziert wurden. Das Textende wird durch ein Dollarzeichen "\$" markiert. Dieses Zeichen kann allerdings durch die Funktion 110 beliebig geändert werden.

```

*****
*
*           10 - Zeile einlesen
*
*****

```

Versorgung: C = DAH
 DE = Pufferadresse oder 0000H

Rückkehr: ---

Mögliche logische Fehler:

Mögliche physikalische Fehler:

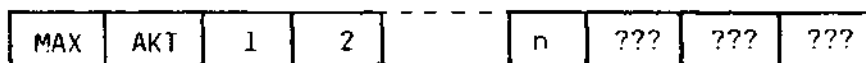
Unterschiede zu CP/M plus:

Es kann ein vom System - Implementator zu schreibender Zeilen - Editor eingebunden werden, um beispielsweise die Hardware - Eigenschaften eines Bildschirms auszunutzen. Dieser Editor kann durch die Funktion 76 ein- bzw. ausgeschaltet werden.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

- 1) Vorbesetzung der Eingabezeile ist möglich.
- 2) Die maximale Zeilenlänge wurde auf 127 Bytes begrenzt.

Es wird vom Bildschirm eine Zeile angefordert, die ab der durch DE bezeichneten Adresse abgelegt wird. Der Puffer hat die folgende Struktur:



MAX wird vor dem Aufruf gesetzt und enthält die maximale Zeichenzahl, die eingegeben werden kann. Wird diese Zahl überschritten, wird die Eingabe abgebrochen. Die maximale Puffergröße beträgt 126 Zeichen.

AKT enthält nach Abschluß der Eingabe die Anzahl der tatsächlich eingegebenen Zeichen. Die Zeichen werden oben durch 1,2 bis n markiert. Hinter dem letzten Zeichen folgen undefinierte Zeichen, die in der obigen Grafik mit "???" bezeichnet sind.

Die folgenden Sonderzeichen sind als Eingabe möglich:

Stopzeichen - falls ein Stopzeichen an erster Stelle eingegeben wurde, wird das Programm abgebrochen. Dieses kann durch einen vorhergehenden Aufruf 109 verhindert werden.

CAN (Ctrl-X) - die gesamte Eingabe wird gelöscht.

BS (Ctrl-H)

DEL (Rubout) - das zuletzt eingegebene Zeichen wird gelöscht.

DLE (Ctrl-P) - der parallel laufende Drucker wird ein- bzw. ausgeschaltet, sofern diese Eingabe nicht durch Aufruf von Funktion 109 unterdrückt wurde.

Fernerhin besteht die Möglichkeit, durch Aufruf der Funktion 76 die Eingabe einer Zeile implementationsabhängig zu gestalten, um beispielsweise die Line Edit - Funktion eines Bildschirmterminals auszunutzen.

Falls DE den Wert 0 enthält, wird der aktuelle DMA - Puffer (siehe Funktion 26) zur Eingabe benutzt. Dieser Puffer kann vorher durch einen beliebigen Text vorbesetzt werden. Der Text muß mit einem Byte mit dem Wert 00H abgeschlossen werden. Die maximale Puffergröße muß ebenfalls vorher besetzt werden. Falls der Text ein Carriage Return (CR) oder ein Line Feed (LF) enthält, wird die gesamte Funktion abgebrochen und die bislang erfolgte Ausgabe als Eingabe behandelt. Das gleiche gilt auch für den Fall, daß der Text länger als die angegebene maximale Puffergröße ist.

```

*****
*
*           11 - Status der Tastatur prüfen
*
*****

```

Versorgung: C = 0BH
 DE = ---

Rückkehr: A = Status

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Die Statusabfrage kann auf die Abfrage von Stopzeichen begrenzt werden.

Falls auf der Tastatur ein Zeichen eingetippt wurde, wird in A der Wert 01H zurückgeliefert, andernfalls enthält A den Wert 0.

Dieser Aufruf kann durch den Aufruf 109 so modifiziert werden, daß der Wert 01H nur dann zurückgeliefert wird, wenn ein gültiges Stopzeichen erkannt wurde.

```

*****
*
*           12 - Version des Systems erfragen
*
*****

```

Versorgung: C = OCH
 DE = ---

Rückkehr: HL = Versionsnummer

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Die Versionsnummer kann durch die Funktion 72 neu gesetzt werden.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

s. o.

In H wird eine 0 zurückgeliefert (MP/M II liefert dort eine 1 zurück). L enthält den Wert 31H, der die kompatible CP/M - Version bezeichnet.

Diese Funktion kann dazu benutzt werden, versionsabhängige Programme zu schreiben, so daß beispielsweise die CP/M 3.0 - kompatiblen Funktionen unter CP/M 2.2 nicht benutzt werden.

```

*****
*
*           13 - Platten normieren
*
*****

```

Versorgung: C = 0DH
DE = ---

Rückkehr: A = 0 oder 0FFH
H = physikalischer Fehlercode

Mögliche logische Fehler:

└ Mögliche physikalische Fehler:

1 Schreib/Lesefehler
4 - Laufwerk nicht anwählbar

Unterschiede zu CP/M plus:

Es können physikalische Fehler auftreten.
Fest installierte Laufwerke werden nicht normiert.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

s. o.

└ Alle angeschlossenen Laufwerke werden "vergessen", d.h. deren Definition wird gelöscht. Beim nächsten Zugriff auf ein Laufwerk werden sämtliche Parameter, wie z. B. das Diskettenformat, neu definiert. Als Bezugslaufwerk wird das Laufwerk A eingesetzt. Die DMA - Adresse wird auf 80H zurückgesetzt.

Die Laufwerke, die sich selbst als fest installiert definiert haben (siehe dazu die Beschreibung des Disk Parameter Blocks im Abschnitt B), werden von dieser Funktion nicht normiert; sie bleiben ein für alle Mal fest definiert. Falls derartige Laufwerke dennoch normiert werden sollen, ist dazu die Funktion 37 zu benutzen. Diese normiert auch fest installierte Laufwerke.

Falls noch ungeschriebene Datenpuffer existieren, werden diese rausgeschrieben. Wenn dabei ein physikalischer Fehler auftrat, wird nach der eingestellten Fehlerbehandlungsart verfahren.

```
*****
*
*          14 - Laufwerk anwählen
*
*****
```

Versorgung: C = 0EH
 E = anzuwählendes Laufwerk

Rückkehr: A = 00H

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Die Laufwerksanwahl wird bis zu einem tatsächlichen Zugriff zurückgestellt.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

s. o.

Das in E enthaltene Laufwerk wird als Bezugslaufwerk für künftige Plattenoperationen definiert. Dabei wird das Laufwerk noch nicht physikalisch angewählt; dies ist erst der Fall, wenn ein Plattenzugriff tatsächlich erfolgt.


```

*****
*
*           15 - Datei öffnen
*
*****

```

Versorgung: C = 0FH
 DE = FCB - Adresse

Rückkehr: A = Fehlercode
 H = physikalischer Fehlercode

Mögliche logische Fehler:

~ Mögliche physikalische Fehler:

- 0 - kein Fehler aufgetreten
- 1 - Schreib/Lesefehler
- 4 - Laufwerk nicht anwählbar
- 9 - Dateiname enthält Fragezeichen

Unterschiede zu CP/M plus:

- 1) Ein angegebenes Paßwort wird ignoriert.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

- 1) Automatische Suche in User 0 und Öffnen im Lesezugriff, falls vorhanden.
- 2) Eindeutiger Dateiname notwendig.
- 3) Übernahme eines Byte Counts.

~ Die durch DE bezeichnete Datei wird geöffnet. Dieser Öffnungsvorgang besteht im wesentlichen aus der Suche nach der Datei im Directory des vorgegebenen Laufwerks sowie der Kopie des gefundenen Eintrags in den FCB. Normalerweise sollten die Extent Bytes die Nummer 0 enthalten, es ist jedoch möglich, dort die Nummer des Extents einzusetzen, das gezielt geöffnet werden soll. Der Dateiname muß eindeutig sein, d. h. er darf keine Fragezeichen enthalten.

Das Current Record Byte des FCB (Byte Nummer 32) sollte auf 0 gesetzt werden, damit anschließend sequentielle Dateizugriffe möglich sind. Hat es jedoch den Wert 0FFH, wird der Byte Count (==> Funktion 31) in dieses Byte hineinkopiert.

Die Datei wird zuerst im lokalen Benutzerbereich gesucht. Ist sie dort nicht vorhanden, wird noch einmal im Benutzerbereich 0 nach der Datei gesucht. Wird sie dort gefunden, wird überprüft, ob das SYS - Attribut (t2') gesetzt ist. Ist dies der Fall, wird sie - allerdings nur zum Lesen - geöffnet. Dieser Status wird durch ein Setzen des Attributbits f8' markiert.

Dieser Mechanismus gestattet es, auf Dateien im Benutzerbereich 0 zuzugreifen, selbst wenn man sich in einem anderen Benutzerbereich befindet. Somit ist es unnötig, von jedem Programm in jedem Benutzerbereich Kopien zur Verfügung zu haben.

Falls im Directory Label die Zeitprotokollierung des Dateizugriffs gefordert wird, wird diese anschließend durchgeführt, falls das Extent 0 geöffnet wurde.

Wurde die Datei gefunden, enthalten A und H beide den Wert 0. Falls sie nicht gefunden wurde, enthält A den Wert 0FFH und H den Wert 0.

Falls ein physikalischer Fehler auftrat, erfolgt im normalen Fehlermodus eine Fehlermeldung und Programmabbruch. Falls der Fehlermodus durch den Aufruf 45 geändert wurde, enthält A den Wert 0FFH und H einen der o. a. physikalischen Fehler.

```

*****
*
*           16 - Datei schließen
*
*****

```

Versorgung: C = IOH
DE = FCB - Adresse

Rückkehr: A = Fehlercode
H = physikalischer Fehlercode

Mögliche logische Fehler:

Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 2 - Schreibschutz Laufwerk
- 3 - Schreibschutz Datei
- 4 - Laufwerk nicht anwählbar
- 9 - Dateiname enthält Fragezeichen

Unterschiede zu CP/M plus:

Der Partial Close wird als normaler Close durchgeführt.
Das Setzen von Zeitmarken für Dateiänderungen wurde von Funktion 21
hierher verlegt.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Der Dateiname muß eindeutig sein.

Die durch den FCB bezeichnete Datei wird geschlossen. Falls der FCB
durch Schreibzugriffe geändert wurde, wird dieser neu in das Directory
übernommen. Ein partielles Schließen der Datei, das durch das Setzen des
Datei - Attributs f5' angezeigt wird, hat ebenfalls das Schließen der
Datei zur Folge.

Falls das Setzen von Zeitmarken für Dateiänderungen im Directory La-
bel gefordert wird, wird die aktuelle Zeit in das entsprechende Feld des
Eintrags übernommen. Falls die Extent - Nummer ungleich 0 ist, wird vor-
her der Eintrag für Extent 0 gesucht. Dieser Vorgang wird allerdings nur
dann durchgeführt, wenn die Datei tatsächlich geändert wurde.

Falls beim Directory - Zugriff ein physikalischer Fehler auftrat, er-
folgt im Normalfall eine Fehlermeldung mit der Möglichkeit, den Vorgang
zu wiederholen. Falls jedoch durch Funktion 45 ein anderer Fehlermodus
eingestellt wurde, enthält A den Wert OFFH und H einen der o. a. physi-
kalischen Fehlercodes.

```
*****
*
*           17 - Datei suchen
*
*****
```

Versorgung: C = 11H
DE = FCB - Adresse

Rückkehr: A = Directory Code
H = Benutzernummer bzw. physikalischer Fehlercode

Mögliche logische Fehler:

Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 4 - Laufwerk nicht anwählbar

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Diese Funktion gestattet es, eine oder mehrere Dateien gezielt zu suchen. Es müssen das Drive - Byte, der Dateiname sowie das erste Extentbyte initialisiert werden. Das zweite Extentbyte wird automatisch auf 0 gesetzt. An allen Positionen sind Fragezeichen erlaubt.

Diese Funktion gibt allerdings nur den ersten gefundenen Eintrag einer Datei zurück. Alle anderen Einträge müssen mit Hilfe der Funktion 18 gesucht werden, wobei zwischen den einzelnen Funktionsaufrufen keine anderen Systemaufrufe, die Directory - Zugriffe zur Folge haben, gestattet sind.

Ist das Drive - Feld mit einem Fragezeichen besetzt, werden alle Directory - Einträge einer Platte zurückgeliefert, egal, ob sie besetzt, frei oder Zeitmarken sind. Diese Aufrufart gestattet es, den Inhalt des Directorys komplett einzulesen.

Normalerweise wird durch das Setzen des Extent - Feldes auf einen bestimmten Wert das Extent, was zu suchen ist, gezielt angegeben. Allerdings können mit dieser Aufrufart nur Extents bis 31 einschließlich gesucht werden. Falls jedoch das Extent - Feld mit einem Fragezeichen besetzt wird, werden alle Extents der zu suchenden Dateien übermittelt.

Ein Fragezeichen im Dateinamen bedeutet, daß das an dieser Position stehende Zeichen irrelevant ist und nicht mit zur Suche herangezogen wird. Eventuell gesetzte Datei - Attribute werden bei der Suche nicht mit herangezogen.

Der gefundene Eintrag wird nicht direkt im FCB zurückgegeben. Vielmehr wird der gesamte Directory - Block von 128 Bytes in den aktuellen DMA - Puffer hineinkopiert. A enthält dann die Nummer des Eintrags innerhalb dieses Blocks. Diese Nummer kann Werte von 0 bis 3 annehmen. Falls Zeiteinträge auf dem Laufwerk durchgeführt werden, enthält A Werte von 0 bis 2, da der letzte Eintrag in jedem Block vom "Zeit - FCB" belegt ist.

Falls die Datei nicht gefunden wurde oder ein physikalischer Fehler auftrat, enthält A den Wert OFFH. H enthält den Wert 0, falls die Datei nicht gefunden werden konnte, andernfalls eine der o.a. physikalischen Fehlercodes. Physikalische Fehlercodes werden nur dann zurückgeliefert, wenn diese Betriebsart durch die Funktion 45 eingestellt wurde.

```
*****
*
*           18 - Folgeeinträge suchen
*
*****
```

Versorgung: C = 12H
 DE = ---

Rückkehr: A = Directory Code
 H = physikalischer Fehlercode

Mögliche logische Fehler:

Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 4 - Laufwerk nicht anwählbar

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Falls mehrere Directory - Einträge durch die Funktion 17 gefunden werden sollen, müssen die Folgeeinträge mit Hilfe dieser Funktion gesucht werden. Dies bedeutet, daß zwischen einem Aufruf 17 und einem Aufruf 18 bzw. zwischen zwei aufeinanderfolgenden Aufrufen von Funktion 18 keine Funktionsaufrufe benutzt werden dürfen, die Directory - Zugriffe zur Folge haben können. Ebenso darf der FCB, der in Funktion 17 als Parameter benutzt wurde, nicht verändert werden.

```
*****
*
*           19 - Datei löschen
*
*****
```

Versorgung: C = 13H
 DE = FCB - Adresse

Rückkehr: A = Fehlercode
 H = physikalischer Fehlercode

Mögliche logische Fehler:

Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 2 - Schreibschutz Laufwerk
- 3 - Schreibschutz Datei
- 4 - Laufwerk nicht anwählbar

Unterschiede zu CP/M plus:

Paßworte sowie das Löschen von XFCBs werden nicht unterstützt.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Falls eine der zu löschenden Dateien schreibgeschützt ist, werden überhaupt keine Dateien gelöscht.

Alle Dateien, die dem angegebenen FCB entsprechen, werden gelöscht. Der Dateiname kann Fragezeichen enthalten. Wenn das Datei - Attribut f5' gesetzt ist, sollen nur XFCBs gelöscht werden. Da EOS keine XFCBs kennt, wird diese Aufrufart wie ein gelungener Löschvorgang behandelt, d.h. daß kein Löschvorgang stattfindet und A und H bei der Rückkehr 00H enthalten.

Falls eine der gefundenen Dateien schreibgeschützt ist, wird die Löschfunktion insgesamt nicht durchgeführt. Es werden überhaupt keine Dateien gelöscht, also auch keine Dateien, die nicht schreibgeschützt sind.

Falls keine zu löschende Datei gefunden werden konnte, enthält A den Wert 0FFH und H den Wert 0. Bei physikalischen Fehlern wird je nach der eingestellten Fehlerbehandlungsart verfahren.

```

*****
*
*           20 - Sequentielles Lesen
*
*****

```

Versorgung: C = 14H
DE = FCB - Adresse

Rückkehr: A = Fehlercode
H = Anzahl Blöcke bzw. physikalischer Fehlercode

Mögliche logische Fehler:

- 1 - Lesen hinter Dateiende
- 9 - FCB ungültig

Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 4 - Laufwerk nicht anwählbar
- 9 - Dateiname enthält Fragezeichen

Unterschiede zu CP/M plus:

Der Multisector Count wurde von 128 auf 256 erweitert.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Es können bis zu 256 Blöcke entsprechend 32 KBytes auf einmal eingelesen werden.

Es werden je nach eingestelltem Multisector Count 1 bis 256 Blöcke zu je 128 Bytes sequentiell in den durch Funktion 26 eingestellten Puffer eingelesen. Die Startposition des Lesevorgangs wird durch das aktuelle Extent sowie den Stand des Current Record Bytes (das 33. Byte des FCB) bestimmt. Ein Anwenderprogramm, das sequentiell lesen will, sollte daher das Current Record Byte nach dem Öffnen der Datei immer auf 0 zurücksetzen. Falls der Wert des Current Record Bytes 127 überschreitet, wird automatisch das nächste Extent geöffnet.

Falls der Lesevorgang einwandfrei beendet werden konnte, enthalten sowohl A als auch H den Wert 0. Wenn jedoch der Lesevorgang abgebrochen werden mußte, weil keine Daten mehr vorhanden waren (entweder bei Dateiende oder beim Auftreten einer Lücke in einer Random - Datei), enthält A den Wert 1 und H die Anzahl der bis zum Fehlerzeitpunkt eingelesenen Sektoren, die sich zwischen 0 und 255 bewegen kann.

Der Fehlercode 9 wird dann zurückgeliefert, wenn die Datei bereits einmal geschlossen wurde und dabei ein Fehler aufgetreten war, so daß der FCB für ungültig erklärt werden mußte.

Physikalische Fehler werden, falls der Fehlerbehandlungsmodus auf automatische Rückkehr eingestellt war, in H zurückgegeben; in diesem Falle enthält A den Wert OFFH.

```

*****
*
*           21 - Sequentiell schreiben
*
*****

```

Versorgung: C = 15H
 DE = FCB - Adresse

Rückkehr: A = Fehlercode
 H = Anzahl Blöcke bzw. physikalischer Fehlercode

Mögliche logische Fehler:

- 1 - Directory voll
- 2 - Speichermangel auf Platte
- 7 - höchstzulässige Dateigröße überschritten
- 9 - FCB ungültig

Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 2 - Schreibschutz Laufwerk
- 3 - Schreibschutz Datei
- 4 - Laufwerk nicht anwählbar
- 9 - Dateiname enthält Fragezeichen

Unterschiede zu CP/M plus:

- 1) Der Multisector Count wurde von 128 auf 256 erhöht.
- 2) Im Fehlerfall erfolgt ein automatisches Schließen der Datei, so daß keine Daten verloren gehen können.
- 3) Das Setzen der Zeitmarke für Dateiänderungen wird von Funktion 16 durchgeführt.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Es können bis zu 256 Blöcke zu je 128 Bytes auf einmal geschrieben werden.

In die durch den FCB bezeichnete Datei werden 1 bis 256 Blöcke zu je 128 Bytes, beginnend bei der eingestellten DMA - Adresse, geschrieben. Die Datei muß vorher geöffnet worden sein. Die Anzahl der zu schreibenden Blöcke wird durch den eingestellten Multisector Count bestimmt. Die Blocknummer, ab der geschrieben werden soll, wird durch das aktuelle Extent sowie durch den Wert des Current Record Bytes (das 33. Byte des FCB) bestimmt. Ein Programm, das sequentiell schreiben will, sollte das Current Record Byte nach dem Öffnen oder Einrichten der Datei immer auf 0 setzen.

Nach Ende der Schreib - Operation zeigen die Extent Bytes bzw. das Current Record Byte auf den nächsten zu schreibenden Block. Nach einem fehlerfreien Schreibvorgang enthalten sowohl A als auch H den Wert 0. Falls der Schreibvorgang aufgrund des Auftretens eines logischen Fehlers abgebrochen werden mußte, enthält A den Fehlercode und H die Anzahl der Blöcke, die bis zum Auftreten des Fehlers geschrieben werden konnte. Diese Zahl kann Werte zwischen 0 und 255 annehmen.

Falls ein Fehler auftrat, versucht EOS automatisch, die Datei zu schließen, so daß im Gegensatz zu CP/M plus keine Daten verloren gehen.

Bei physikalischen Fehlern richtet sich die Behandlung nach der eingestellten Betriebsart. Im Falle einer Rückkehr enthält A den Wert 0FFH und H den physikalischen Fehlercode.

```
*****
*
*           22 - Datei neu einrichten
*
*****
```

Versorgung: C = 16H
 DE = FCB - Adresse

Rückkehr: A = Fehlercode
 H = physikalischer Fehlercode

Mögliche logische Fehler:

5 - Directory voll

Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 2 - Schreibschutz Laufwerk
- 3 - Schreibschutz Datei
- 4 - Laufwerk nicht anwählbar
- 8 - Datei existiert bereits
- 9 - Dateiname enthält Fragezeichen

Unterschiede zu CP/M plus:

Das Attribut f6' (auch XFCB einrichten) wird ignoriert.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Einrichten von Doppeleinträgen ist nicht mehr möglich.

Die durch den FCB bezeichnete Datei wird neu eingerichtet. Das rufende Programm muß sowohl den Dateinamen als auch das Extent Byte vorher initialisieren, wobei der Dateiname keine Fragezeichen enthalten darf. Der FCB wird initialisiert und für Schreib/Lesezugriffe geöffnet. Das obere Extent Byte wird von EOS immer auf 0 gesetzt. Es ist daher möglich Extents zwischen 0 und 31 direkt einzurichten. Allerdings muß davor gewarnt werden, da EOS empfindlich auf unkorrekte Directory - Einträge reagieren kann.

Das Datei - Attribut f6' (XFCB einrichten) wird unter EOS ignoriert, da EOS keine Paßworte unterstützt.

Wenn Zeiteinträge für Neueinrichtungen gefordert werden, wird, sofern das Extent 0 eingerichtet wurde, die aktuelle Zeit im Zugriffs - Zeitfeld festgehalten. Falls auch das Festhalten von Änderungen gewünscht wird, wird die Zeit auch im Änderungs - Zeitfeld festgehalten.

Ehe die Datei eingerichtet wird, wird geprüft, ob auf dem Laufwerk und in dem aktuellen Benutzerbereich bereits eine Datei gleichen Namens existiert. In diesem Fall wird die Einrichtung abgelehnt.

Bei physikalischen Fehlern richtet sich die Behandlung nach der eingestellten Betriebsart. Im Falle einer Rückkehr enthält A den Wert OFFH und H den physikalischen Fehlercode.

```

*****
*
*           23 - Datei umbenennen
*
*****

```

Versorgung: C = 17H
DE = FCB - Adresse

Rückkehr: A = Fehlercode
H = physikalischer Fehlercode

Mögliche logische Fehler:

Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 2 - Schreibschutz Laufwerk
- 3 - Schreibschutz Datei
- 4 - Laufwerk nicht anwählbar
- 8 - Datei existiert bereits
- 9 - Dateiname enthält Fragezeichen

Unterschiede zu CP/M plus:

Paßworte werden ignoriert.
Die Attribute des alten Namens werden übernommen.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Beide Dateinamen müssen eindeutig sein.
Der neue Dateiname darf nicht bereits vergeben sein.

Die durch den FCB bezeichnete Datei wird umbenannt. Der FCB enthält in diesem Fall zwei Dateinamen. Der zweite Name liegt ab dem 17. Byte des FCB. Seine Struktur entspricht genau der der ersten 16 Bytes des FCB, mit dem Unterschied, daß das Drive - Feld nicht ausgewertet wird. Beide Dateinamen müssen eindeutig sein, d.h. sie dürfen keine Fragezeichen enthalten. Falls bereits eine Datei gleichen Namens existiert, wird die Umbenennung abgelehnt. Eventuell gesetzte Datei - Attribute werden in den neuen Namen mit übernommen.

Falls die Datei umbenannt werden konnte, haben sowohl A als auch H den Wert 0. Falls die Datei nicht gefunden wurde, hat A den Wert 0FFH und H den Wert 0.

Bei physikalischen Fehlern richtet sich die Behandlung nach der eingestellten Betriebsart. Im Falle einer Rückkehr enthält A den Wert 0FFH und H den physikalischen Fehlercode.

```
*****
*
*                24 - Definierte Laufwerke erfragen
*
*****
```

Versorgung: C = 18H
 DE = ---

Rückkehr: HL = Login - Vektor

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

In HL wird ein Bitvektor zurückgeliefert. Dieser Vektor bezeichnet alle dem System zu diesem Zeitpunkt bekannten Laufwerke, d. h. alle Laufwerke, auf die seit der letzten Normierung zugegriffen wurde. Das niederwertigste Bit (Register L, Bit 0) bezeichnet dabei das Laufwerk A, das höchstwertige (Register H, Bit 7) das Laufwerk P.

```
*****
*
*           25 - Aktuelles Laufwerk erfragen
*
*****
```

Versorgung: C = 19H
 DE = ---

Rückkehr: A = Laufwerks - Nummer

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Das durch die Funktionen 13 oder 14 eingestellte Bezugslaufwerk wird zurückgegeben, wobei eine 0 dem Laufwerk A, eine 1 dem Laufwerk B etc. entspricht.


```

*****
*
*           26 - DMA - Adresse setzen
*
*****

```

Versorgung: C = 1AH
 DE = DMA - Adresse

Rückkehr: ---

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Im DMA - Puffer liegende Paßworte werden ignoriert.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Die in DE liegende Adresse wird zukünftig als Transferadresse für sämtliche Platten - Operationen verwendet. Zusätzlich wird diese Adresse als Informations - Puffer für andere Funktionen benutzt. In Gegensatz zu CP/M plus wird der DMA - Puffer nicht für die Übergabe von Paßworten benutzt, da EOS Paßworte nicht benutzt.

Außer den üblichen Schreib/Lesefunktionen benutzen die folgenden Funktionen den DMA - Puffer für Informationsübergabe:

- 10 - Zeile von der Tastatur einlesen
- 17 - Datei suchen
- 18 - Folgeeintrag suchen
- 46 - Freien Speicherplatz erfragen

```

*****
*
*          27 - Allocation - Vektor erfragen
*
*****

```

Versorgung: C = 1BH
 DE = ---

Rückkehr: HL = Vektoradresse oder 0FFFFH

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Es können physikalische Fehler auftreten.

Diese Funktion liefert die Startadresse des Allocation Vectors des aktuellen Laufwerks zurück. Dies ist ein Bitvektor, in dem jedes Bit, ab dem höchstwertigen Bit des ersten Bytes gezählt, einen Allocation Cluster kennzeichnet. Ist das Bit gesetzt, ist der jeweilige Cluster belegt.

Der Allocation Vector sollte nicht dazu benutzt werden, um den freien Platz auf einem Laufwerk zu berechnen, da der Allocation Vector u. U. in einer getrennten Speicherbank liegen kann. Vielmehr ist dazu die Funktion 46 zu benutzen, die auch von allen EOS - Hilfsprogrammen benutzt wird.

Falls das aktuelle Laufwerk noch nicht definiert sein sollte, werden zuerst die Definitionen eingelesen. Sollte es dabei zu einem physikalischen Fehler kommen, enthält HL bei einer möglichen Rückkehr den Wert 0FFFFH.

```
*****
*
*                28 - Schreibschutz setzen
*
*****
```

Versorgung: C = 1CH
 DE = ---

Rückkehr: ---

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Dieser Aufruf setzt einen Schreibschutz auf das Bezugslaufwerk. Dieser Schreibschutz kann nur durch die Funktionen 13 bzw. 37 zurückgenommen werden.

```
*****
*
*                29 - Schreibschutzvektor erfragen
*
*****
```

Versorgung: C = 1DH
 DE = ---

Rückkehr: HL = Schreibschutzvektor

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Es wird ein Bitvektor zurückgeliefert, der alle zur Zeit schreibgeschützten Laufwerke angibt, wobei Laufwerk A dem niederwertigsten Bit (Register L, Bit 0) und P dem höchstwertigen Bit (Register H, Bit 7) entspricht.

```

*****
*
*          30 - Datei - Attribute setzen
*
*****

```

Versorgung: C = 1EH
DE = FCB - Adresse

Rückkehr: A = Fehlercode
H = physikalischer Fehlercode

Mögliche logische Fehler:

~ Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 2 - Schreibschutz Laufwerk
- 4 - Laufwerk nicht anwählbar
- 9 - Dateiname enthält Fragezeichen

Unterschiede zu CP/M plus:

Paßwort wird ignoriert.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Der Byte Count wurde neu eingeführt.
Das Backup - Attributbit wurde im Gegensatz zu DiCOS invertiert.

~ Die Attribute der durch den FCB bezeichneten Datei werden in das Directory übernommen. Die folgenden Attribute sind definiert:

```

f1' - f3' = frei definierbar
f4' - f8' = reserviert für EOS
t1'      = Schreibschutz
t2'      = Systemdatei
t3'      = Archiviert

```

Falls das Attributbit f6' gesetzt ist, wird das Current Record - Feld des FCB als Byte Count übernommen. Dieser Byte Count kann bei Öffnen der Datei wieder angefordert werden.

Falls die Attribute der angegebenen Datei neu gesetzt werden konnten, enthalten sowohl A als auch H den Wert 0. Wenn die angegebene Datei nicht gefunden werden konnte, enthält A den Wert OFFH und H den Wert 0.

Bei physikalischen Fehlern richtet sich die Behandlung nach der eingestellten Betriebsart. Im Falle einer Rückkehr enthält A den Wert OFFH und H den physikalischen Fehlercode.

```

*****
*
*           31 - Disk Parameter Block anfordern
*
*****

```

Versorgung: C = 1FH
 DE = ---

Rückkehr: HL = Adresse des Disk Parameter Blocks oder 0FFFFH

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Es können physikalische Fehler auftreten.

Es wird die Startadresse des aktuellen Disk Parameter Blocks zurückgeliefert.

Falls das aktuelle Laufwerk noch nicht definiert sein sollte, werden zuerst die Definitionen eingelesen. Sollte es dabei zu einem physikalischen Fehler kommen, enthält HL bei einer möglichen Rückkehr den Wert 0FFFFH.

```

*****
*
*           32 - Benutzerbereich setzen
*
*****

```

Versorgung: C = 20H
E = Benutzerbereich oder OFFH

Rückkehr: A = Benutzerbereich

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Es können Benutzerbereiche bis einschließlich 33 eingestellt werden.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Der DiCOS - Bereich von 0 bis 99 wurde eingeschränkt.

Der zur Zeit gültige Benutzerbereich kann durch diese Funktion abgefragt oder auch geändert werden. Wenn E den Wert OFFH enthält, wird kein neuer Benutzerbereich gesetzt, vielmehr wird dann in A der aktuelle Benutzerbereich zurückgeliefert. Wenn E andernfalls einen Wert kleiner als 34 enthält, wird der aktuelle Benutzerbereich auf diesen Wert umgestellt.

Der benutzbare Bereich wurde bewußt nicht auf den Bereich 0 bis 15 begrenzt, um interessierten Anwendern die Möglichkeit zu geben, XFCBs zu bearbeiten, die in den Benutzerbereichen 16 bis 31 liegen. Ferner können der Directory Label (Bereich 32) und die Zeit - FCB's (Bereich 33) direkt abgefragt werden.


```

*****
*
*           33 - Lesen im Randomzugriff
*
*****

```

Versorgung: C = 21H
DE = FCB - Adresse

Rückkehr: A = Fehlercode
H = Anzahl Blöcke bzw. physikalischer Fehlercode

Mögliche logische Fehler:

- 1 - Lesen hinter Dateiende
- 3 - logischer Fehler im Directory
- 4 - Random - Positionierung auf nicht existierenden Dateiteil
- 6 - Random - Satznummer ungültig

Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 4 - Laufwerk nicht anwählbar
- 9 - Dateiname enthält Fragezeichen

Unterschiede zu CP/M plus:

Die höchstzulässige Blocknummer beträgt 65535.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Dieser Funktionsaufruf verhält sich ähnlich dem Funktionsaufruf 20 (Sequentielles Lesen). Der Hauptunterschied ist der, daß die Startposition durch eine 3 Bytes lange Blocknummer bezeichnet wird, die im FCB hinter dem Current Record Byte, also ab dem 34. Byte liegt. Diese Blocknummer kann sich zwischen 0 und 65535 bewegen, wobei das niedrigste Byte auf Position 34, das mittlere Byte auf Position 35 und das höchste Byte auf Position 36 abgelegt wird. Die Datei muß vor dem Lesezugriff geöffnet sein.

Es können je nach eingestelltem Multisector Count zwischen 128 und 32768 Bytes auf einmal eingelesen werden. Die eingestellte Satzposition wird durch den Lesevorgang nicht verändert.

Es ist möglich, einen Block gezielt einzulesen und von diesem Punkt aus sequentiell weiterzuarbeiten. Allerdings wird dann beim ersten sequentiellen Zugriff der aktuelle Block noch einmal eingelesen (bzw. die ersten Blöcke bei einem Multisector Count größer als 1).

Falls ein logischer Fehler auftrat, enthält H die Anzahl der Blöcke, die erfolgreich eingelesen werden konnten. A enthält in diesem Fall die Fehlernummer.

Bei physikalischen Fehlern richtet sich die Behandlung nach der eingestellten Betriebsart. Im Falle einer Rückkehr enthält A den Wert OFFH und H den physikalischen Fehlercode.

```

*****
*
*           34 - Schreiben im Random - Zugriff
*
*****

```

Versorgung: C = 22H
 DE = FCB - Adresse

Rückkehr: A = Fehlercode
 H = Anzahl Blöcke bzw. physikalischer Fehlercode

Mögliche logische Fehler:

- 2 - Speichermangel
- 3 - logischer Fehler im Directory
- 5 - Directory voll
- 6 - Random - Satznummer ungültig
- 7 - höchstzulässige Dateigröße überschritten

Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 2 - Schreibschutz Laufwerk
- 3 - Schreibschutz Datei
- 4 - Laufwerk nicht anwählbar
- 9 - Dateiname enthält Fragezeichen

Unterschiede zu CP/M plus:

Bei einem Fehler wird die Datei automatisch geschlossen.
 Die höchstzulässige Blocknummer beträgt 65535.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Diese Funktion verhält sich ähnlich wie die Funktion 21 (sequentielles Schreiben). Der Hauptunterschied ist der, daß die Startposition durch eine Blocknummer bezeichnet wird, die hinter dem Current Record Byte des FCB ab der 34. Position abgelegt wird. Der Wert dieser Blocknummer bewegt sich zwischen 0 und 65535, wobei das niedrigste Byte auf Position 34, das mittlere Byte auf Position 35 und das höchste Byte auf Position 36 abgelegt wird.

Es werden je nach eingestelltem Multisector Count 128 bis 32768 Bytes geschrieben, wobei jedoch die eingestellte Satzposition nicht geändert wird. Es ist möglich, ab der eingestellten Satzposition sequentiell weiterzuarbeiten, wobei jedoch der erste Block noch einmal geschrieben wird (bzw. die ersten Blöcke bei einem Multisector Count größer als 1).

Falls ein logischer Fehler auftrat, enthält H die Anzahl der Blöcke, die erfolgreich geschrieben werden konnten. A enthält in diesem Fall die Fehlernummer.

Bei physikalischen Fehlern richtet sich die Behandlung nach der eingestellten Betriebsart. Im Falle einer Rückkehr enthält A den Wert OFFH und H den physikalischen Fehlercode.

```

*****
*
*           35 - Dateiende berechnen
*
*****

```

Versorgung: C = 23H
DE = FCB - Adresse

Rückkehr: A = Fehlercode
H = physikalischer Fehlercode

Mögliche logische Fehler:

— Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 4 - Laufwerk nicht anwählbar
- 9 - Dateiname enthält Fragezeichen

Unterschiede zu CP/M plus:

Die höchste Blocknummer beträgt 65536.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

— Das Dateiende wird gesucht und die Nummer des ersten freien Blocks in den Bytes 34 - 36 des FCB zurückgeliefert. Falls die höchste Blocknummer beispielsweise 65535 beträgt, wird die Nummer 65536 zurückgeliefert, selbst wenn die Datei physikalisch nur aus diesem einen Block bestehen sollte. Die Datei muß vorher nicht geöffnet werden. Sie muß sogar geschlossen sein, falls sie zum Schreiben geöffnet war, damit die Information im Directory auf dem neuesten Stand ist.

Falls die Datei gefunden wurde, enthalten sowohl A als auch H den Wert 0. Falls sie nicht gefunden werden konnte, enthält A den Wert 0FFH und H den Wert 0.

Bei physikalischen Fehlern richtet sich die Behandlung nach der eingestellten Betriebsart. Im Falle einer Rückkehr enthält A den Wert 0FFH und H den physikalischen Fehlercode.

```

*****
*
*           36 - Blocknummer setzen
*
*****

```

Versorgung: C = 24H
 DE = FCB - Adresse

Rückkehr: ---

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Die höchste Blocknummer beträgt 65535.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Diese Funktion berechnet die Nummer des nächsten zu bearbeitenden Blocks und legt sie in den Bytes 34 bis 36 des FCB ab. Damit ist es möglich, frei zwischen sequentieller und Random - Bearbeitung hin- und herzuschalten.

```

*****
*
*           37 - Laufwerk normieren
*
*****

```

Versorgung: C = 25H
DE = Bitvektor

Rückkehr: A = 0

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Auch fest installierte Laufwerke werden normiert.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

s.o.

DE enthält einen Bitvektor, wobei jedes Bit ein zu normierendes Laufwerk bezeichnet. Das niederwertigste Bit (Register L, Bit 0) entspricht hierbei dem Laufwerk A, das höchstwertige (Register H, Bit 7) dem Laufwerk P.

Die Normierung bewirkt, daß sämtliche Definitionen dieses Laufwerks vergessen werden. Der nächste Zugriff auf dieses Laufwerk hat eine komplette Neudefinition zur Folge. Das Format des Datenträgers wird hierbei neu bestimmt, und ein eventueller Schreibschutz des Datenträgers wird zurückgenommen.

Diese Funktion normiert auch fest installierte Laufwerke, die sonst vom Funktionsaufruf 13 nicht erfaßt werden.

```
*****
*
*              38 - Laufwerk sperren
*
*****
```

Versorgung: C = 26H
 DE = Bitvektor

Rückkehr: A = 0

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Diese Funktion ist nur unter einem Multiuser - Betriebssystem von Bedeutung. Sie wurde nur wegen der Kompatibilität zu MP/M definiert. Ein Aufruf dieser Funktion hat den o.a. Rückkehrwert zur Folge.


```
*****
*
*           39 - Laufwerk freigeben
*
*****
```

Versorgung: C = 27H
 DE = Bitvektor

Rückkehr: A = 0

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Diese Funktion ist nur unter einem Multiuser - Betriebssystem von Bedeutung. Sie wurde nur wegen der Kompatibilität zu MP/M definiert. Ein Aufruf dieser Funktion hat den o.a. Rückkehrwert zur Folge.

```

*****
*
*           40 - Schreiben im Randomzugriff mit Initialisierung
*
*****

```

Versorgung: C = 28H
 DE = FCB - Adresse

Rückkehr: A = Fehlercode
 H = Anzahl Blöcke bzw. physikalischer Fehlercode

Mögliche logische Fehler:

- 2 - Speichermangel
- 3 - logischer Fehler im Directory
- 4 - Random - Positionierung auf nicht existierenden Dateiteil
- 5 - Directory voll
- 6 - Random - Satznummer ungültig
- 7 - höchstzulässige Dateigröße überschritten
- 9 - FCB ungültig

Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 2 - Schreibschutz Laufwerk
- 3 - Schreibschutz Datei
- 4 - Laufwerk nicht anwählbar
- 9 - Dateiname enthält Fragezeichen

Unterschiede zu CP/M plus:

Die höchstzulässige Blocknummer beträgt 65535.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Diese Funktion verhält sich genau wie die Funktion 34. Der Unterschied zwischen beiden Funktionen besteht darin, daß vorher nicht zugewiesene Bereiche der Platte mit Nullen initialisiert werden, so daß sich ein genau definierter Dateiinhalt ergibt. Die Funktion 34 initialisiert neu zugewiesene Plattenbereiche nicht, so daß uninitialisierte Blöcke innerhalb der Datei auftreten können.

```
*****
*
*               41 - Schreiben Random mit Prüfllesen
*
*
*****
```

Versorgung: C = 28H
 DE = FCB - Adresse

Rückkehr: A = 0FFH
 H = 0

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2:

Neue Funktion.

Diese Funktion ist eine spezifische MP/M II - Funktion, die unter einem Einbenutzersystem wie EOS sinnlos ist und daher weggelassen wurde. Ein Aufruf hat die o. a. Rückkehrwerte zur Folge.

```
*****
*
*           42 - Block sperren
*
*****
```

Versorgung: C = 2AH
 DE = FCB - Adresse

Rückkehr: A = 0
 H = 0

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Diese Funktion ist eine spezifische MP/M II - Funktion, die unter einem Einbenutzersystem wie EOS sinnlos ist und daher weggelassen wurde. Ein Aufruf hat die o. a. Rückkehrwerte zur Folge.

```
*****
*
*           43 - Block freigeben
*
*****
```

Versorgung: C = 2BH
 DE = FCB - Adresse

Rückkehr: A = 0
 H = 0

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Diese Funktion ist eine spezifische MP/M II - Funktion, die unter einem Einbenutzersystem wie EOS sinnlos ist und daher weggelassen wurde. Ein Aufruf hat die o. a. Rückkehrwerte zur Folge.

```

*****
*
*          44 - Multisector Count setzen
*
*****

```

Versorgung: C = 2CH
E = Anzahl Blöcke

Rückkehr: A = Alter Multisector Count

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Der alte Multisector Count wird zurückgeliefert.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Das Setzen des Multisector Counts beeinflusst die folgenden Funktionen:

- 20 - Sequentielles Lesen
- 21 - Sequentielles Schreiben
- 33 - Lesen im Randomzugriff
- 34 - Schreiben im Randomzugriff
- 40 - Schreiben im Randomzugriff mit Initialisierung

Normalerweise wird bei diesen Funktionen ein Block zu 128 Bytes transferiert. Die Anzahl der pro Systemaufruf transferierten Blöcke kann mit Hilfe dieser Funktionen zwischen 1 und 256 eingestellt werden (eine 0 entspricht 256 Blöcken), so daß bei einem Aufruf der o. a. Funktionen ein Datentransfer von bis zu 32 KBytes stattfindet.

Tritt bei diesem Transfer ein logischer Fehler auf, so wird dieser in A zurückgemeldet. H enthält in diesem Fall die Anzahl der bis zu dem Auftreten des Fehlers korrekt übertragenen Blöcke. Diese Zahl kann sich zwischen 0 und 255 bewegen.

Ein physikalischer Fehler wird je nach der eingestellten Fehlerbehandlungsart behandelt. In diesem Falle enthält H den physikalischen Fehlercode.

```

*****
*
*           45 - Fehlerbehandlungsart setzen
*
*****

```

Versorgung: C = 2DH
 DE = Fehlercode

Rückkehr: ---

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Der in E übergebene Wert bestimmt das Fehlerverhalten von EOS beim Auftreten eines physikalischen Fehlers. E kann die folgenden Werte annehmen:

OFFH - ein Fehler wird nicht angezeigt. Der physikalische Fehlercode wird an das rufende Programm übergeben.

OFEH - ein Fehler wird angezeigt und der physikalische Fehlercode an das rufende Programm übergeben.

andere - ein Fehler wird angezeigt. Der Benutzer hat die Möglichkeit, interaktiv auf diesen Fehler zu reagieren.

..

```

*****
*
*           46 - Freien Speicherplatz erfragen
*
*****

```

Versorgung: C = 2EH
E = Laufwerk

Rückkehr: A = Fehlercode
H = physikalischer Fehlercode

Mögliche logische Fehler:

Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 2 - Schreibschutz Laufwerk
- 3 - Schreibschutz Datei
- 4 - Laufwerk nicht anwählbar
- 8 - Datei existiert bereits
- 9 - Dateiname enthält Fragezeichen

Unterschiede zu CP/M plus:

Das High Byte ist immer 0.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Der freie Platz auf dem angegebenen Laufwerk wird ermittelt und in die ersten drei Bytes des DMA - Puffers abgelegt. Das niederwertige Byte belegt das erste Byte des Puffers, das höhere Byte das zweite Byte des Puffers. Das dritte Byte des Puffers wird von EOS immer auf 0 gesetzt. Die derart abgelegte Zahl gibt den freien Platz in 128 - Bytes - Blöcken an. um den freien Platz in KBytes zu bekommen, ist diese Zahl durch 8 zu teilen.

Falls des Laufwerk vorher nicht definiert war, wird es physikalisch angewählt. Dabei eventuell auftretende physikalische Fehler werden je nach der eingestellten Fehlerbehandlungsart behandelt.


```

*****
*
*           47 - Folgeprogramm aufrufen
*
*****

```

Versorgung: C = 2FH
 E = Chain Flag

Rückkehr: ---

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Dieser Aufruf ist genaugenommen nichts anderes als ein definierter Start des Kommando - Interpreters. EOS erwartet im DMA - Puffer eine normale Kommandozeile, die durch ein Byte mit dem Wert 00H abgeschlossen sein muß. Diese Kommandozeile wird von Kommando - Interpreter wie eine normale Eingabe behandelt und entsprechend ausgeführt, wobei als Bezugs- laufwerk und aktueller Benutzerbereich die zuletzt im Kommando - Interpreter gesetzten Bezugsgrößen eingestellt werden. Ist das Purge - Flag (Register E) ungleich 0, werden jedoch das aktuelle Bezugslaufwerk sowie der aktuelle Benutzerbereich beim Programmstart übernommen.

Eventuell auftretende Fehler werden wie bei einer normalen Kommando- eingabe vom Kommando - Interpreter behandelt.

Mit Hilfe der Funktion 108 ist es möglich, dem Folgeprogramm einen Wert zu übermitteln.

..

```
*****
*
*           48 - Datenpuffer schreiben
*
*****
```

Versorgung: C = 30H
E = Purge Flag

Rückkehr: A = Fehlercode
H = physikalischer Fehlercode

Mögliche logische Fehler:

Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 2 - Schreibschutz Laufwerk
- 4 - Laufwerk ungültig

Unterschiede zu CP/M plus:

Das Purge Flag wird nicht interpretiert.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Durch die EOS - Speicherverwaltung bedingt, werden die Datenpuffer bei einem Schreibbefehl nicht unbedingt gleich auf die Platte übertragen. Diese Funktion veranlaßt das sofortige Schreiben aller noch ungeschriebenen Datenpuffer. Tritt bei diesem Schreibvorgang ein physikalischer Fehler auf, wird je nach der eingestellten Fehlerbehandlungsart verfahren.

```

*****
*
*           49 - Systemvariable setzen oder lesen
*
*****

```

Versorgung: C = 31H
DE = Zusatzversorgung

Rückkehr: HL = gelesener Wert

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Diese Funktion gestattet den Zugriff auf den internen EOS - Variablenbereich. DE zeigt auf die folgende Zusatzversorgung:

DB	OFFSET	; Offset vom Bereichsanfang
DB	CODE	; Schreib / Lesecode
DW	VALUE	; Byte- oder Wortwert zum Setzen

Der Schreib / Lesecode kann die folgenden Werte annehmen:

OFFH - Bytewert setzen
OFFH - Wortwert setzen
000H - Wert lesen

Alle anderen Werte sind reserviert.

Die zulässigen Werte für OFFSET sind dem Kapitel über den Systemvariablenbereich zu entnehmen.

WARNUNG: Die unbedachte Benutzung dieser Funktion führt unweigerlich zum Systemabsturz!

```

*****
*
*          50 - Hardware - Schnittstelle aufrufen
*
*****

```

Versorgung: C = 32H
 DE = Zusatzversorgung

Rückkehr: Register mit Rückkehrwerten besetzt

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Funktion 27 wird unterstützt.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Da EOS kein CP/M - ähnliches BIOS mehr im eigentlichen Sinne besitzt, wurde diese Funktion eingeführt, um Applikationsprogrammen die Möglichkeit zu geben, die ehemaligen CP/M - BIOS - Funktionen zu simulieren. EOS selbst verwaltet eine CP/M 2.2 - bzw. DiCOS - kompatible BIOS - Einsprungtabelle, die ebenfalls mit Hilfe dieser Funktion simuliert wurde.

DE zeigt auf die folgende Zusatzversorgung:

DB	FUNC	; Funktionsnummer
DB	AREG	; Wert für Register A
DW	BCREG	; Wert für Register BC
DW	DEREG	; Wert für Register DE
DW	HLREG	; Wert für Register HL

Die einzelnen Funktionen und deren Rückkehrwerte sind ausführlich im Abschnitt über die Hardware - Schnittstelle beschrieben.

```
*****
*
*           59 - Overlay laden
*
*****
```

Versorgung: C = 3BH
 DE = FCB - Adresse

Rückkehr: A = Fehlercode
 H = physikalischer Fehlercode

Mögliche logische Fehler:

~ Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 4 - Laufwerk nicht anwählbar
- 9 - Dateiname enthält Fragezeichen

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

~ Mit Hilfe dieser Funktion kann der EOS - Programmlader angesprochen werden. Dieser wird normalerweise nur vom Kommando - Interpreter benutzt und liegt daher auch nicht im Speicher. Um ihn im Speicher zu behalten, müssen entweder bereits Funktionsmodule geladen sein oder das rufende Programm muß mit einem leeren Programm - Header versehen sein.

In DE wird ein FCB erwartet. Dieser FCB muß aktiviert sein, d. h. die Datei muß geöffnet sein. Desweiteren erwartet EOS die Ladeadresse in den Bytes 33 und 34 (R0 und R1) des FCB. Diese Adresse darf nicht kleiner als 0100H sein. Ebenfalls muß genug Platz für die gesamte Datei vorhanden sein. Falls eine der beiden Bedingungen nicht erfüllt wurde, kehrt EOS mit A = OFFH und H = 0 zurück. Alle anderen möglichen Fehler entsprechen denen, die bei der Funktion 20 (Sequentielles Lesen) auftreten können.

Falls die Datei vom Typ "PRL" ist, wird nach dem Ladevorgang eine Relocation durchgeführt. Eventuelle in der Datei enthaltene Funktionsmodule werden aktiviert.

```
*****
*
*           60 - Funktionsmodul ansprechen
*
*****
```

Versorgung: C = 3CH
DE = Zusatzversorgung

Rückkehr: HL = freier Fehlercode

Mögliche logische Fehler:

Funktionsabhängig.

Mögliche physikalische Fehler:

Funktionsabhängig.

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Diese Funktion gestattet es, aktive Funktionsmodule zwecks Initialisierung, Parameterübergabe etc. gezielt anzusprechen. DE zeigt auf die folgende Zusatzversorgung:

DB	FUNC	; Sonderfunktion
DW	NPARAM	; Anzahl Parameter
DW	PARAM1	; Parameter 1
....		
DW	PARAMn	; Parameter n

Die Sonderfunktionen 128 bis 255 sind für EOS reserviert.

Falls sich keines der geladenen Module angesprochen fühlte, liefert EOS den Wert OFFH in A und 00H in H zurück.

```
*****
*
*           64 - Fehlerbehandlungsart setzen
*
*****
```

Versorgung: C = 40H
 E = 0 oder OFFH

Rückkehr: ---

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Neue Funktion.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Alte DiCOS - Funktion.

Diese Funktion hat den selben Effekt wie die Funktion 45, außer, daß alle anderen Werte als OFFH die Rückkehr zur normalen Fehlerbehandlungsart zur Folge haben.

Aus Kompatibilitätsgründen zu DiCOS wurde diese Funktion beibehalten.

```
*****
*
*          65 - Prozeßnummer wechseln
*
*****
```

Versorgung: C = 41H
 E = Prozeßnummer

Rückkehr: A = 0

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Neue Funktion.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Alte DiCOS - Funktion.

Aus Kompatibilitätsgründen zu DiCOS wurde diese Funktion beibehalten.
Sie hat jedoch keinen Effekt mehr.


```
*****
*
*           66 - Speicherbereiche ausgeben
*
*****
```

Versorgung: C = 42H
 DE = Zusatzversorgung

Rückkehr: A = 0

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Neue Funktion.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Alte DiCOS - Funktion.

Aus Kompatibilitätsgründen zu DiCOS wurde diese Funktion beibehalten.
Sie hat jedoch keinen Effekt mehr.

```
*****
*
*           67 - Test auf Stopzeichen
*
*****
```

Versorgung: C = 43H
 E = Zeichen oder OFFH

Rückkehr: A = 0 oder OFFH

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Neue Funktion.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Alte DiCOS - Funktion.

Das in E enthaltene Zeichen wird überprüft, ob es eines der beiden Stopzeichen (Control-C oder das frei definierbare) entspricht. Ist dies der Fall, enthält A den Wert OFFH, andernfalls wird 00H zurückgeliefert.

Enthält E den Wert OFFH, wird vorher die Tastatur überprüft. Liegt ein Zeichen an, wird dieses überprüft.

Wurde durch Funktion 109 der Check auf Stopzeichen unterdrückt, wird immer A = 00H zurückgeliefert.

```
*****
*
*           68 - Datei ausführen
*
*****
```

Versorgung: C = 44H
DE = FCB - Adresse

Rückkehr: A = Fehlercode

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Neue Funktion.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Alte DiCOS - Funktion. DE = 0 nicht mehr zulässig.

Die durch den FCB bezeichnete Datei wird ausgeführt, d. h. die Tastatur wird auf diese Datei umgelegt. Die Datei kann verschiedene Steuerzeichen enthalten, die näher in der Beschreibung des Programms \$D0.COM beschrieben sind.

Konnte die Datei nicht ausgeführt werden, weil sie nicht gefunden werden konnte oder weil ein physikalischer Fehler auftrat, enthält A den Wert 0FFH, andernfalls enthält A den Wert 0. Falls bei der Abarbeitung der Datei ein Fehler auftritt, wird ohne Kommentar wieder auf die Tastatur umgeschaltet. Dieses ist ebenfalls beim Auftreten des Dateiendes der Fall.

Falls das Attributbit f3' im FCB gesetzt ist, wird die Datei nach der Abarbeitung gelöscht.

Diese Funktion wird unter EOS mit Hilfe eines Funktionsmoduls realisiert. Dieses Modul, \$D0.RSX, kann mit Hilfe des Dienstprogramms \$ATTACH an ein Programm angebunden werden, das diese Funktion benötigt.

```

*****
*
*               69 - Datei im Hintergrund drucken
*
*
*****

```

Versorgung: C = 45H
 DE = FCB - Adresse

Rückkehr: A = Fehlercode

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Neue Funktion.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Alte DiCOS - Funktion. DE = 0 nicht mehr zulässig.

Die durch den FCB bezeichnete Datei wird im Hintergrund ausgedruckt.

Konnte die Datei nicht geöffnet werden, weil sie nicht gefunden werden konnte oder weil ein physikalischer Fehler auftrat, enthält A den Wert OFFH, andernfalls enthält A den Wert 0. Falls beim Druck der Datei ein Fehler auftritt, wird der Ausdruck abgebrochen. Dieses ist ebenfalls beim Auftreten des Dateiendes der Fall.

Falls das Attributbit f3' im FCB gesetzt ist, wird die Datei nach der Abarbeitung gelöscht.

Diese Funktion wird unter EOS mit Hilfe eines Funktionsmoduls realisiert. Dieses Modul, \$D0.RSX, kann mit Hilfe des Dienstprogramms \$ATTACH an ein Programm angebunden werden, das diese Funktion benötigt.

```

*****
*
*           70 - Hintergrunddruck steuern
*
*****

```

Versorgung: C = 46H
 E = Steuercode

Rückkehr: A = 0 oder 0FFH

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Neue Funktion.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Alte DiCOS - Funktion.

Ein durch die Funktion 69 gestarteter Hintergrunddruck kann durch diese Funktion gesteuert werden. Die folgenden Steuercodes sind definiert:

- 0 - Überprüfen, ob der Hintergrunddruck aktiv ist.
- 1 - Hintergrund - Druck neu starten. Die Datei wird neu ausgegeben.
- 2 - Hintergrund - Druck abbrechen. Die Datei wird gelöscht, falls sie das Attributbit f3' gesetzt hatte.
- 3 - Hintergrunddruck freigeben. Ein durch Code 4 angehaltener Ausdruck wird fortgesetzt.
- 4 - Hintergrunddruck anhalten. Der Druck wird so lange angehalten, bis er durch einen Aufruf mit den Codes 1, 2, oder 3 aktiviert bzw. abgebrochen wurde.

Diese Funktion wird unter EOS mit Hilfe eines Funktionsmoduls realisiert. Dieses Modul, \$DO.RSX, kann mit Hilfe des Dienstprogramms \$ATTACH an ein Programm angebunden werden, das diese Funktion benötigt.

```

*****
*
*           71 - Laufwerk auf Bereitschaft prüfen
*
*****

```

Versorgung: C = 47H
 E = Laufwerk

Rückkehr: A = 0 oder OFFH
 H = Bitvektor

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Neue Funktion.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Das in E bezeichnete Laufwerk wird überprüft, ob es für Datentransfers bereit ist, wobei eine 0 dem Laufwerk A, eine 1 dem Laufwerk B etc. entspricht. Ist das Laufwerk bereit, enthält A den Wert OFFH und H eine Bitleiste mit den folgenden Werten:

Bit 7 gesetzt - Laufwerk ist nicht wechselbar
 Bit 6 gesetzt - Laufwerk enthält doppelseitige Diskette

Ist das Laufwerk nicht bereit, enthält A den Wert 0. Ist es nicht definiert, enthält A den Wert 7.

Dieser Aufruf führt keine physikalischen Plattenzugriffe durch. Insbesondere erfolgt keine Formatbestimmung, so daß nachfolgende Schreib / Leseoperationen durchaus noch fehlerhaft verlaufen können.

```

*****
*
*          72 - Versionsnummer setzen oder lesen
*
*****

```

Versorgung: C = 48H
 DE = Versionsnummer oder OFFFFH

Rückkehr: A = Versionsnummer (falls angefordert)

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Neue Funktion.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Die durch die Funktion 12 zu erfragende Versionsnummer wird neu gesetzt. Der Wert in E wird als Versionsnummer übernommen. Dadurch ist es beispielsweise möglich, Programme zum Laufen zu bringen, die eine spezifische Versionsnummer erwarten.

Enthält DE den Wert OFFFFH, wird die aktuelle Versionsnummer in A zurückgeliefert.

```
*****
*
*           73 - Drucker - Wartezeit setzen oder lesen
*
*****
```

Versorgung: C = 49H
 DE = Wartezeit oder 0FFFFH

Rückkehr: A = Wartezeit (falls angefordert)

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Neue Funktion.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Die Wartezeit für die Nicht - Bereit - Fehlermeldung eines Ausgabege-
räts wird neu bestimmt. E kann Werte von 0 bis 255 enthalten, wobei eine
0 einer 256 entspricht.

Enthält DE den Wert 0FFFFH, wird die aktuelle Wartezeit in A zu-
rückgeliefert.

Die Funktion der Wartezeit ist im Kapitel über die Gerätezuord-
nung nachzulesen.


```

*****
*
*           74 - Tab Stop setzen oder lesen
*
*****

```

Versorgung: C = 4AH
 DE = Tab Stop oder OFFFFH

Rückkehr: A = Tab Stop (falls angefordert)

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Neue Funktion.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Der gültige Tab Stop für die Funktionen 2, 9 und 111 wird neu gesetzt. Er darf nur die Werte 2, 4, 8, 16 oder 32 enthalten. Alle anderen Werte führen zur Konfusion auf dem Bildschirm.

Enthält DE den Wert OFFFFH, wird der aktuelle Tab Stop in A zurückgeliefert.

Beim Start des Systems ist der Tab Stop auf 8 eingestellt.

```

*****
*
*
*
*
*****
        
```

Versorgung: C = 4BH
 DC = Stopzeichen oder OFFFH
 Rückkehr: A = Stopzeichen (falls angefordert)

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Neue Funktion.

Unterschiede zu CP/M 2.2 bzw. DICS:

Neue Funktion.

Das frei definierbare zweite Stopzeichen wird neu gesetzt. E enthält
 das zu setzende Stopzeichen. Hat E den Wert 0, wird das bis dahin gül-
 tige zweite Stopzeichen gelöscht.
 Enthält DE den Wert OFFFH, wird das aktuelle zweite Stopzeichen in
 A zurückgeliefert.
 Beim Start des Systems ist das zweite Stopzeichen deaktiviert.

```
*****
*
*           76 - EOS - Flags setzen oder lesen
*
*****
```

Versorgung: C = 4CH
DE = Flags oder OFFFFH

Rückkehr: A = Flags (falls angefordert)

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Neue Funktion.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

EOS enthält ein Flag Byte, das durch diese Funktion gesetzt bzw. abgefragt werden kann. Das Flag Byte enthält die folgenden Flags:

Bit 7 - reserviert zwecks Kompatibilität zu CP/M plus.
Bit 6 - reserviert zwecks Kompatibilität zu CP/M plus.

Bit 5 EIN - Funktion 10 benutzt den Screen Editor.
AUS - Funktion 10 benutzt den EOS - Mini - Editor.
Voreinstellung: AUS

Bit 4 EIN - Bei Programmabbruch durch Stopzeicheneingabe wird noch einmal Rückfrage gehalten.
AUS - Bei Stopzeicheneingabe erfolgt Programmabbruch.
Voreinstellung: AUS

Bit 3 EIN - Bei jedem Systemaufruf wird auf Stopzeicheneingabe geprüft. Ist ein Stopzeichen erkannt, wird das Programm unterbrochen.
AUS - EOS verhält sich normal.
Voreinstellung: AUS

Die Bits 2 bis 0 sind reserviert. Eine genauere Beschreibung des Flag Bytes ist im entsprechenden Kapitel zu finden.

Enthält DE den Wert OFFFFH, wird das aktuelle Flag Byte in A zurückgeliefert.

```
*****
*
*          98 - Datenblöcke freigeben
*
*****
```

Versorgung: C = 62H

Rückkehr: A = 0

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Nicht implementiert.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Diese Funktion ist unter EOS nicht implementiert, da EOS eine andere Speicherverwaltung als CP/M besitzt. Ein Aufruf dieser Funktion hat die Rückgabe von 0 in A und HL zur Folge.

```

*****
*
*          99 - Datei kürzen
*
*****

```

Versorgung: C = 63H
DE = FCB - Adresse

Rückkehr: A = Fehlercode
H = physikalischer Fehlercode

Mögliche logische Fehler:

Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 2 - Schreibschutz Laufwerk
- 3 - Schreibschutz Datei
- 4 - Laufwerk nicht anwählbar
- 9 - Dateiname enthält Fragezeichen

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

DE zeigt auf den FCB einer Datei. Diese Datei muß geschlossen sein. Die Bytes 34 bis 36 enthalten eine Blocknummer, wobei das niederwertigste Byte in Byte 34, das mittlere Byte in Byte 35 und das höchstwertigste Byte (unter EOS immer 0) in Byte 36 abgelegt wird. Dieser Block wird zum letzten Block der Datei erklärt. Alle Information, die sich in Bereichen mit Blocknummern befindet, die größer sind als die angegebene Blocknummer, ist verloren. Der freigewordene Speicherplatz steht sofort wieder zur Verfügung.

Falls die Funktion erfolgreich abgeschlossen werden konnte, enthalten sowohl A als auch H den Wert 0. Bei einem logischen Fehler enthält A den Wert 0FFH und H den Wert 0. Diese Fehler sind:

- die Datei wurde nicht gefunden
- die angegebene Blocknummer liegt hinter dem Dateiende
- die angegebene Blocknummer liegt in einem Dateibereich, der in Folge von Randomzugriffen noch nicht beschrieben wurde

Physikalische Fehler werden je nach der eingestellten Fehlerart behandelt.

```

*****
*
*           100 - Directory Label schreiben
*
*****

```

Versorgung: C = 64H
DE = FCB - Adresse

Rückkehr: A = Fehlercode
H = physikalischer Fehlercode

Mögliche logische Fehler:

Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 2 - Schreibschutz Laufwerk
- 4 - Laufwerk nicht anwählbar

Unterschiede zu CP/M plus:

Das Paßwort wird nicht mit übernommen.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Der Directory Label auf dem im FCB bezeichneten Laufwerk wird neu geschrieben. Als Name der Platte wird der Dateiname des FCB übernommen. Das 13. Byte (das Extent Byte) des FCB enthält einen Bitvektor mit den folgenden Daten:

- Bit 6 - Zeitmarken für Dateizugriffe schreiben
- Bit 5 - Zeitmarken für Dateiänderungen schreiben
- Bit 4 - Zeitmarken für Neueinrichtungen von Dateien schreiben

Alle anderen Bits sind reserviert.

Das Setzen des Bits 6 schließt das Setzen von Bit 4 aus und umgekehrt, da für beide Zeiteinträge nur ein Datenfeld zur Verfügung steht. Die Einrichtung eines derartigen Labels wird abgelehnt.

Falls das Schreiben von Zeitmarken gefordert wird, muß das Directory mit Hilfe des Dienstprogramms \$INITDIR auf die Aufnahme von Zeitmarken vorbereitet worden sein, da andernfalls die Einrichtung eines derartigen Labels abgelehnt wird.

Wenn der Label korrekt eingerichtet bzw. geändert werden konnte, enthalten sowohl A als auch H den Wert 0. Wurde das Einrichten wegen fehlender Zeitmarken - Einträgen oder wegen Unverträglichkeiten der geforderten Zeiteinträge miteinander abgelehnt, enthält A den Wert OFFH und H den Wert 0. Bei physikalischen Fehlern wird nach der eingestellten Fehlerbehandlungsart verfahren.

```

*****
*
*           101 - Datenbyte des Directory Labels erfragen
*
*****

```

Versorgung: C = 65H
E = Laufwerk

Rückkehr: A = Datenbyte oder Fehlercode
H = physikalischer Fehlercode

Mögliche logische Fehler:

Mögliche physikalische Fehler:

1 - Schreib/Lesefehler
4 - Laufwerk nicht anwählbar

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Das Datenbyte des Directory Labels des durch E bezeichneten Laufwerks wird zurückgeliefert. E enthält eine 0 für Laufwerk A, eine 1 für Laufwerk B etc. Das Datenbyte selbst entspricht dem Byte 13 des Directory Label - Eintrags. Es enthält die folgenden Bits:

Bit 6 - Zeitmarken für Dateizugriffe schreiben
Bit 5 - Zeitmarken für Dateiänderungen schreiben
Bit 4 - Zeitmarken für Neueinrichtungen von Dateien schreiben
Bit 0 - Directory Label existiert

Falls kein Directory Label existiert, enthält A den Wert 0, ansonsten das Datenbyte. H enthält in beiden Fällen den Wert 0. Physikalische Fehler werden wie üblich in A und H zurückgeliefert, falls der dort auftretende Fehler entsprechend behandelt wurde.


```
*****
*
*           102 - Zeitmarken lesen
*
*****
```

Versorgung: C = 66H
DE = FCB - Adresse

Rückkehr: A = Fehlercode
H = physikalischer Fehlercode

Mögliche logische Fehler:

Mögliche physikalische Fehler:

- 1 - Schreib/Lesefehler
- 4 - Laufwerk nicht anwählbar
- 9 - Dateiname enthält Fragezeichen

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Die durch den FCB bezeichnete Datei wird auf dem angegebenen Laufwerk gesucht. Die dazugehörigen Zeitmarken werden in der folgenden Weise in den FCB eingetragen:

- Byte 13 - Paßwortbyte, unter EOS immer 0.
- Byte 25-28 - Zeitmarke für Neueinrichtung oder Zugriff
- Byte 29-32 - Zeitmarke für Änderung

Falls keine Zeitmarken existieren, werden diese Felder mit 0 aufgefüllt. Das Format der Zeitmarken ist unter dem entsprechenden Kapitel nachzulesen.

Wurde die Datei gefunden, enthalten sowohl A als auch H den Wert 0. Konnte die Datei nicht gefunden werden, enthält A den Wert OFFH und H den Wert 0. Bei physikalischen Fehlern wird je nach der eingestellten Fehlerbehandlungsart verfahren.

```
*****
*
*          103 - XFCB schreiben
*
*****
```

Versorgung: C = 67H
DE = XFCB - Adresse

Rückkehr: A = 0FFH
H = 0

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Nicht implementiert.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Diese Funktion dient unter CP/M zum Einrichten von Paßworten für eine Datei. Da EOS keine Paßworte unterstützt, wird dieser Aufruf ignoriert. Es werden die o.a. Werte zurückgegeben.

```
*****
*
*      104 - Datum setzen
*
*****
```

Versorgung: C = 68H
DE = Adresse der Datenstruktur

Rückkehr: ---

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DICO5:

Neue Funktion.

Das Systemdatum kann mit Hilfe dieser Funktion neu gesetzt werden. DE zeigt auf die folgende Struktur:

DW	DATUM	: Julianisches Datum. Tag 1 entspricht dem
DB	HOUR	: Stunde im BCD - Format
DW	MINUTE	: Minute im BCD - Format

Die Sekunde wird beim Aufruf immer auf 0 gesetzt.

```

*****
*
*           105 - Datum lesen
*
*****

```

Versorgung: C = 69H
 DE = Adresse der Datumsstruktur

Rückkehr: A = aktuelle Sekunde im BCD - Format

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Das aktuelle Systemdatum wird abgefragt und in die durch DE bezeichnete Datumsstruktur abgelegt. Diese Struktur hat das folgende Aussehen:

DW	DATUM	; Julianisches Datum. Tag 1 entspricht dem
		; 1. Januar 1978.
DB	HOURL	; Stunde im BCD - Format
DW	MINUTE	; Minute im BCD - Format

Die aktuelle Sekunde wird im BCD - Format im Register A zurückgeliefert.

```
*****
*
*          106 - Paßwort setzen
*
*****
```

Versorgung: C = 6AH
 DE = Paßwortadresse

Rückkehr: ---

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Nicht implementiert.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Unter CP/M dient diese Funktion zum Setzen eines Standard - Paßworts.
Da EOS keine Paßworte unterstützt, wird dieser Aufruf ignoriert.

```

*****
*
*               107 - Seriennummer erfragen
*
*****

```

Versorgung: C = 6BH
 DE = Adresse der Seriennummer

Rückkehr: Seriennummer eingetragen.

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Die Seriennummer des Systemkerns wird ab der durch DE bezeichneten Adresse wie folgt eingetragen:

DB	OEM	; Seriennummer des Herstellers
DW	USER	; Seriennummer des Endbenutzers

Der Wert 0FFFFH veranlaßt EOS, den aktuellen Return Code zurückzuliefern.

- | | |
|-------------|---|
| 0000 - FFFF | - erfolgreiches Programmende |
| FF00 - FF7F | - fehlerhaftes Programmende |
| FF80 - FFFC | - reserviert |
| FFFD | - Programmabbruch durch einen physikalischen Fehler |
| FFFF | - Programmabbruch durch Eingabe eines Stopzeichens |

Dieser Aufruf gestattet eine kleine, begrenzte Kommunikation zwischen einzelnen Programmen. Diese Programme müssen allerdings durch den Aufruf 47 geladen werden, da der Kommando - Interpreter den Return Code immer auf 0 zurücksetzt. Einzelne Return Codes werden auch vom Systemkern selbst gesetzt. Im Einzelnen sind die folgenden Werte zulässig:

Neue Funktion.

Unterschiede zu CP/M 2.2 bzw. DICS:

Unterschiede zu CP/M plus:

Mögliche physikalische Fehler:

Mögliche logische Fehler:

Rückkehr: HL = Rückkehrcode

Versorgung: C = 6CH
DE = Rückkehrcode oder 0FFFFH

```
*****
*
* 108 - Programm - Rückkehrcode setzen oder lesen
*
*****
```

```

*****
*
*           109 - Console Mode setzen oder lesen
*
*****

```

Versorgung: C = 6DH
DE = OFFFFH (Lesen) oder Bitvektor

Rückkehr: HL = Console Mode

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Die Reaktion von EOS kann mit dieser Funktion in weiten Teilen beeinflusst werden. Diese Funktion geht mit der Funktion 76 einher, die weitere Steuerungsmöglichkeiten von EOS erschließt. Der übergebene Bitvektor hat die folgende Bedeutung, wobei Bit 0 dem Bit 0 in Register I entspricht:

Bit 0 - EIN: Die Funktion 11 liefert einen Wert ungleich 0 nur dann zurück, falls ein Stopzeichen eingegeben wurde.

AUS: Die Funktion 11 liefert immer dann einen Wert ungleich 0 zurück, wenn ein beliebiges Zeichen eingegeben wurde.

Bit 1 - EIN: Die Bildschirmausgabe kann nicht durch Control-S angehalten werden. Control-S wird als Eingabe erkannt.

AUS: Die Bildschirmausgabe kann mit Control-S angehalten und mit Control-Q oder einem erneuten Control-S fortgesetzt werden.

Bit 2 - EIN: Die Interpretation von Tab Stops unterbleibt. Der Drucker kann nicht mehr durch Control-P ein- bzw. ausgeschaltet werden.

AUS: Tab Stops werden je nach eingestelltem Tab Stop interpretiert. Der Drucker kann durch Eingabe von Control-P ein- bzw. ausgeschaltet werden.

Bit 3 - EIN: Das Stopzeichen wird ignoriert. Programme können nicht mehr durch Eingabe des Stopzeichens abgebrochen werden.

AUS: Das Stopzeichen wird erkannt.

Die Bits 8 und 9 sind für die Redirection der Tastatur bestimmt. Sie steuern den Status der Tastatur:

00 - Status Requests werden ignoriert, außer, wenn zwei Status Requests hintereinander kommen. In diesem Fall liefert der zweite Request den Wert 1 zurück.

01 - Status Requests liefern immer 00H zurück.

10 - Status Requests liefern immer 01H zurück.

11 - Redirection ignorieren.

```
*****
*
*          110 - Stringende setzen oder lesen
*
*****
```

Versorgung: C = 6EH
 DE = 0FFFFH (lesen) oder E = Stringende

Rückkehr: A = Stringende

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Das Stringende - Zeichen für Funktion 9, normalerweise das Zeichen "\$", kann mit dieser Funktion beliebig umdefiniert werden. Falls DE den Wert 0FFFFH enthält, wird das aktuelle Stringende - Zeichen in A zurückgeliefert.

```

*****
*
*           111 - Block auf dem Bildschirm ausgeben
*
*****

```

Versorgung: C = 6FH
 DE = Blockwerte

Rückkehr: ---

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

DE zeigt auf die folgende Struktur:

DW	START	; Anfang des Blocks
DW	SIZE	; Länge des Blocks

Der komplette Block wird auf dem Bildschirm ausgegeben. Da EOS dabei intern die Funktion 2 benutzt, sind alle dort beschriebenen Features - abhängig vom Console Mode - aktiv.

```
*****
*
*               112 - Block auf dem Drucker ausgeben
*
*****
```

Versorgung: C = 7DH
 DE = Blockwerte

Rückkehr: ---

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

DE zeigt auf die folgende Struktur:

DW	START	; Anfang des Blocks
DW	SIZE	; Länge des Blocks

Der komplette Block wird auf dem Drucker ausgegeben.

```
*****
*
*           115 - Grafik aufrufen
*
*****
```

Versorgung: C = 73H
DE = Zusatzversorgung

Rückkehr: ---

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Neue Funktion.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

Mit Hilfe dieser Funktion kann die eingebaute Grafik direkt angesprochen werden. DE zeigt auf die folgende Zusatzversorgung:

DB	OPCODE	; Befehlscode
DB	0	; reserviert
DW	PARAM1	; Parameter 1
...		
DW	PARAMn	; Parameter n

Die Anzahl der Parameter kann von Aufruf zu Aufruf unterschiedlich sein. Eine nähere Beschreibung des Opcodes ist dem Abschnitt über die Grafik zu entnehmen.

```

*****
*
*          152 - FCB aufbauen
*
*****

```

Versorgung: C = 98H
DE = Zusatzversorgung

Rückkehr: HL = Pointer oder 0

Mögliche logische Fehler:

Mögliche physikalische Fehler:

Unterschiede zu CP/M plus:

Es können keine Fehler auftreten.

Unterschiede zu CP/M 2.2 bzw. DiCOS:

Neue Funktion.

DE zeigt auf die folgende Struktur:

DW	STRING	; Startadresse eines Textstrings
DW	FCB	; Adresse eines FCB's

Der angegebene String wird nach einer gültigen Dateibezeichnung durchsucht und diese in den angegebenen FCB eingetragen. Die maximale Länge des Eingabestrings beträgt 128 Bytes. Der FCB muß 36 Bytes lang sein. Bei der Konstruktion des FCB wird wie folgt vorgegangen:

Es werden Blanks und Tabs übersprungen. Falls im nachfolgenden String ein Doppelpunkt enthalten ist, wird der vorangehende Buchstabe als Laufwerksindikator übernommen. Dann werden Name und Typ übernommen, wobei der Name vom Typ durch einen Punkt getrennt ist. Falls sich an den Dateinamen ein Semikolon anschließt, wird die folgende Zeichenkette als Paßwort übernommen. Falls Name oder Paßwort länger als 8 Zeichen bzw. der Typ länger als 3 Zeichen sind, wird der Rest des Teilstrings ignoriert. Alle Zeichen werden in Großbuchstaben umgewandelt.

Der FCB wird wie folgt besetzt:

Byte 0 - Laufwerksindikator oder 0
Bytes 1 bis 8 - Dateiname, mit Blanks aufgefüllt
Bytes 9 bis 11 - Dateityp, mit Blanks aufgefüllt
Bytes 12 bis 16 - Mit Nullen gefüllt
Bytes 17 bis 24 - Paßwort, mit Blanks gefüllt
Bytes 25 bis 32 - Mit Nullen gefüllt

Falls Name, Type oder Paßwort nicht vorhanden sind, werden stattdessen Blanks eingesetzt.

Die folgenden Zeichen werden als Separator erkannt:

Control Characters

Blank

Null (00H)

; außer zwischen Typ und Paßwort

=

<

>

. (Punkt) außer zwischen Name und Typ

: außer zwischen Laufwerk und Name

, (Komma)

ö

Ä

Ü diese Zeichen (leider) wegen der CP/M plus - Kompatibilität

Nach dem Ende der Funktion überprüft EOS noch den nachfolgenden Text. Dazu werden Blanks und Tabs übersprungen. EOS kehrt dann mit den folgenden Werten in HL zurück:

bei 00H oder CR: HL = 0, um das Ende anzuzeigen

bei einem Separator: HL enthält die Adresse des Separators

sonst: HL enthält die Adresse des ersten Zeichens
hinter dem erkannten FCB.

Abschnitt B
die Hardware - Schnittstelle

B.1. Einleitung

EOS benötigt eine (einigermaßen) normierte Schnittstelle, um die Hardware des MZ-3500 anzusprechen. Diese Schnittstelle setzt sich aus einer ganzen Zahl von Funktionsaufrufen, ähnlich den EOS - Funktionsaufrufen, zusammen. Mit Hilfe dieser Funktionsaufrufe werden Zeichen ausgegeben und empfangen, die Plattenspeicher gelesen und geschrieben und vieles andere mehr. Ein Großteil dieser Aufrufe ist für Benutzer, die hardwarenahe Programme schreiben wollen, von Interesse und daher in diesem Abschnitt dokumentiert. Die Ansteuerung des Bildschirms sowie der Grafik wurde zu eigenen Abschnitten zusammengefaßt, da die Beschreibung dieser Geräte den Rahmen dieses Abschnitts gesprengt hätten.

Im Einzelnen enthält dieser Abschnitt die folgenden Beschreibungen:

- die Struktur des Disk Parameter Blocks
- die Hardware - Funktionsaufrufe
- eine Liste der Fehlermeldungen der Hardware - Treiber

B.2. Aufruf der Hardware - Funktionen

Alle Funktionsaufrufe an die Hardware - Schnittstelle werden über den EOS - Funktionsaufruf 50 geleitet. Die Register werden beim EOS - Aufruf wie folgt besetzt:

Register C: 32H

Register DE: Zeiger auf die folgende Zusatzversorgung:

DB	FUNC	; Hardware - Funktionscode
DB	AREG	; Wert für Register A
DW	BCREG	; Wert für Register BC
DW	DEREG	; Wert für Register DE
DW	HLREG	; Wert für Register HL

EOS besetzt die CPU - Register vor dem Aufruf der jeweiligen Hardware - Funktion mit den mitgelieferten Werten vor. Alle Ergebniswerte werden im Doppelregister HL zurückgeliefert. Zusätzlich erscheint der Wert des Registers L im Register A. Eine Ausnahme bildet der Funktionsaufruf 25 (Daten verschieben), der die Register BC, DE und HL besetzt zurückliefert.

Nehmen wir an, Sie wollen ein Zeichen direkt unter Umgehung von EOS auf dem Bildschirm ausgeben. Der zu schreibende Code sähe dann so aus:

	LD	C,50	; EOS - Funktionscode
	LD	DE,ZUSATZ	; Zusatzversorgung
	CALL	5	; Ausführung
	...		
ZUSATZ:	DB	4	; Hardware - Funktionscode
	DB	0	; A - Register
	DW	ZEICHEN	; auszugebendes Zeichen in BC
	DW	0	; DE - Register
	DW	0	; HL - Register

B.3. Die Hardware - Einsprungtabelle

Die Betriebssysteme CP/M und DiCOS sind ebenfalls mit einem Mechanismus versehen, der den Zugriff auf die Hardware - Schnittstelle ermöglicht. Diese Schnittstelle ist in Form einer Einsprungtabelle realisiert. Auf der Adresse 0 ist ein Sprung auf den Warmstartpunkt abgelegt, der es gestattet, die Lage dieser Einsprungtabelle zu berechnen. Unter EOS ist diese Tabelle ebenfalls vorhanden, um die Kompatibilität zu CP/M 2.2 und DiCOS zu gewährleisten. Die Tabelle wurde über den EOS - Aufruf 50 simuliert. Sie liegt beim MZ-3500 ab der Adresse 0F500H. Es sind die folgenden Vektoren implementiert:

	JP	FUNC0	; Kaltstart, NICHT VERWENDEN!
WARM:	JP	FUNC1	; Warmstart
	JP	FUNC2	; Tastatur - Status
	JP	FUNC3	; Tastatur - Eingabe
	JP	FUNC4	; Bildschirmausgabe
	JP	FUNC5	; Druckerausgabe
	JP	FUNC6	; Ausgabe Zusatzgerät
	JP	FUNC7	; Eingabe Zusatzgerät
	JP	FUNC8	; Lesekopf auf Spur 0
	JP	FUNC9	; Laufwerk anwählen
	JP	FUNC10	; Spur setzen
	JP	FUNC11	; Sektor setzen
	JP	FUNC12	; DMA - Adresse setzen
	JP	FUNC13	; Block lesen
	JP	FUNC14	; Block schreiben
	JP	FUNC15	; Status Drucker
	JP	FUNC16	; Sektor - Umrechnung

Es sei noch erwähnt, daß der Sprung auf den Warmstartpunkt das Nachladen des Kommando - Interpreters sowie des Programmladers zur Folge hat.

Auf der Adresse 0 ist der folgende Befehl abgelegt:

JP	WARM	; entsprechend 0F503H beim MZ-3500
----	------	------------------------------------

Als Beispiel wollen wir wiederum ein Zeichen auf dem Bildschirm ausgeben:

LD	C,ZEICHEN	; auszugebendes Zeichen
LD	HL,(1)	; Zieladresse des Sprungbefehls
LD	L,12	; Offset für Bildschirmausgabe
JP	(HL)	; und Ansprung der Funktion

B.4. Fehlermeldungen der Hardware - Aufrufe

Es gibt zwei Arten von Fehlermeldungen: Die erste umfaßt alle Schreib/Lesevorgänge auf magnetischen Datenträgern. Diese Meldung hat das folgende Format:

(ttssm) Meldung

wobei die einzelnen Zeichen die folgende Bedeutung haben:

- tt - Spur des Zugriffs
- ss - Sektor des Zugriffs (beide Werte in Sedezimal - Darstellung)
- m - Modus, R = Lesen, W = Schreiben

Es können die folgenden Meldungen auftreten:

- 1) Sektoradresse nicht gefunden
- 2) Sektor nicht gefunden
- 3) Laufwerk nicht bereit
- 4) Prüfsummenfehler
- 5) Spurende erreicht
- 6) Fehlerhafte Versorgung
- 7) Systemfehler
- 8) Interner Fehler

Zur Bedeutung der Fehlermeldungen:

1) Es konnten keine Adressen - Marken auf dem Datenträger gefunden werden, weil sich zum Beispiel das Aufzeichnungsformat geändert hat.

2) Der gewünschte Sektor konnte auf der eingestellten Spur nicht gefunden werden.

3) Das Laufwerk ist nicht bereit, Daten aufzunehmen oder zu senden.

4) Die Prüfsumme des eingelesenen Sektors stimmt nicht. Der Sektor enthält fehlerhafte Daten.

5) Ein Lesebefehl über mehrere Sektoren hinweg hat das Spurende erreicht. Dieser Fehler kommt praktisch nur bei einer fehlerhaften Implementation des Disk Parameter Blocks vor.

6) Fehlerhafte Versorgung. Tritt beispielsweise bei einer unzulässig hohen Spurnummer auf.

7) Systemfehler. Dieser Fehler umfaßt alle anderen, nicht näher definierten Fehler beim Schreib/Lesevorgang.

8) Dies ist meist ein tödlicher Fehler. EOS ist durch irgendwelche Umstände außer Rand und Band geraten. Da hilft nur ein erneuter Systemstart. Dieser Fehler sollte nie auftreten.

Kommen wir zur zweiten Fehlermeldung: diese kommt bei nicht bereiten Ausgabegeräten vor und sieht wie folgt aus:

Gerät XXXXXX nicht bereit

wobei XXXXXX den Namen des Geräts, wie er in der Gerätetabelle definiert ist, bezeichnet. Diese Meldung tritt nach dem Ablauf des Time Outs (der durch die EOS - Funktion 73 zu beeinflussen ist) auf. Falls zu diesem Zeitpunkt an der Tastatur eine Eingabe vorliegt, wird daraufhin der Ausgabevorgang abgebrochen und ein Systemrestart durchgeführt. Wenn keine Eingabe anliegt, wird der Time Out - Vorgang wiederholt. Der Bildschirm ist von diesem Vorgang ausgeschlossen.

B.5. Fest installierte Laufwerke

Mit diesem Begriff werden alle Laufwerke bezeichnet, die im System fest installiert sind. Dies sind vor allem Hard Disks und die RAM Disk. Mit diesen Laufwerken hat es eine besondere Bewandnis: Der EOS - Funktionsaufruf 13 bewirkt keine Normierung dieser Laufwerke. Diese Normierung ist im Grunde nur bei einem Wechsel des Aufzeichnungsmediums vonnöten. Der dazu erforderliche Zeitaufwand kann natürlich bei fest installierten Geräten entfallen. Ein derartiges Gerät wird beim Erstzugriff einmal normiert und bleibt dann fest im System definiert. Sollte sich einmal die Notwendigkeit ergeben, auch ein derartiges Laufwerk zu normieren, kann dazu die EOS - Funktion 37 (Laufwerke normieren) benutzt werden.

Ein derartiges Laufwerk wird daran erkannt, daß das höchstwertige Bit im Skew Byte des Disk Parameter Blocks (das Byte Nummer 18) gesetzt ist.

B.6. Der Skew Factor

Ein Skew Factor bestimmt die Zuordnung zwischen logischen und physikalischen Sektoren. Um eine möglichst hohe Geschwindigkeit beim Plattenzugriff zu erzielen, greift EOS nicht auf hintereinanderliegende Sektoren zu. Wenn vielmehr ein Sektor eingelesen wurde, wird der nächste Sektor (oder die nächsten Sektoren) beim Lesen übersprungen, um EOS die Gelegenheit zu geben, den eingelesenen Sektor zu verarbeiten. Der Abstand von zwei hintereinander zu lesenden Sektoren wird Skew Factor genannt. Bei einem Skew Factor von 2 wird beispielsweise erst Sektor 1, dann Sektor 3, dann Sektor 5 etc. eingelesen.

B.7. Der Disk Parameter Block

Für jedes angeschlossene Laufwerk ist ein sogenannter Disk Parameter Block definiert. Diese Datenstruktur enthält alle relevanten Angaben über das Format der verwendeten Platte. Es ist beispielsweise für jedes Diskettenformat einer 8 - Zoll - Diskette ein derartiger Disk Parameter Block vonnöten. Die Startadresse kann mit Hilfe der EOS - Funktion 31 angefordert werden. Desweiteren ist es möglich, den Disk Parameter Block des aktuellen Laufwerks direkt aus dem Systemvariablen - Bereich auszu-lesen. Von dieser Möglichkeit sollte allerdings kein Gebrauch gemacht werden, da EOS nicht garantiert, daß der dort abgelegte Disk Parameter Block auch mit dem des aktuellen Laufwerks übereinstimmt.

Im Nachfolgenden der Aufbau des Disk Parameter Blocks:

Bytes 1 - 2 : Die Anzahl der Blöcke pro Spur. Diese Zahl gibt an, wie viele Blöcke zu je 128 Bytes auf eine Spur passen. Eine Spur muß nicht notwendigerweise mit einer physikalischen Spur übereinstimmen. Bei doppelseitigen Diskettenlaufwerken werden beispielsweise die beiden Spuren der Vorderseite und der Rückseite zu einer Spur zusammengefaßt.

Byte 3 : Block Shift Factor. Diese Zahl gibt an, um wie viele Bits eine logische Blocknummer (die ja 128 Bytes adressiert) nach rechts geschiftet werden muß, um die Nummer des Allocation Clusters zu erhalten. Es gelten dabei die folgenden Zusammenhänge:

Größe eines Clusters	Shift Factor
1 KBytes	3
2 KBytes	4
4 KBytes	5
8 KBytes	6
16 KBytes	7

Byte 4 : Block Mask. Dies ist das Pendant zum Block Shift Factor, nämlich die Maske, mit der die Nummer des gewünschten 128 - Bytes - Blocks maskiert werden muß, um den Block - Offset innerhalb des Allocation Clusters zu erhalten. Diese maskierten Bits stellen genau die Bits dar, die beim Shiftvorgang durch den Block Shift Factor verloren gehen. Es gelten die folgenden Zusammenhänge:

Größe eines Clusters	Shift Mask
1 KBytes	7
2 KBytes	15
4 KBytes	31
8 KBytes	63
16 KBytes	127

Byte 5 : Extent Mask. Diese Maske wird bei Suchen im Directory über das Extent Byte gelegt, um den richtigen Directory - Eintrag für das gewünschte Extent zu finden. Diese Maske muß auf die Anzahl der definierten Clusters Rücksicht nehmen. Sind mehr als 256 Clusters definiert, müssen pro Cluster jeweils 2 Bytes in der Allocation List des Eintrags verwendet werden. Dadurch passen weniger Clusters in die Allocation List, und die Extent Mask fällt daher kleiner aus. Es gelten folgende Zusammenhänge:

Clustergröße	kleine Platten	große Platten
1 KBytes	0	nicht definiert
2 KBytes	1	0
4 KBytes	3	1
8 KBytes	7	3
16 KBytes	15	7

Bytes 6 - 7 : Nummer des höchsten Clusters. Dieser Wortwert stellt die Anzahl Clusters auf der Platte minus eins dar.

Bytes 8 - 9 : Nummer des letzten Directory Entries. Dieser Wortwert stellt die Anzahl der definierten Entries minus eins dar. Ein Directory Entry belegt 32 Bytes, d. h. es passen 4 Entries in einen Block zu 128 Bytes, 32 Entries in einen Cluster zu 1 KBytes, 64 Entries in einen Cluster zu 2 KBytes etc.

Bytes 10 - 11 : Pre - Allocation Map. Diese beiden Bytes enthalten einen Bitvektor, der zum Initialisieren des Allocation Vectors verwendet wird. Mit diesem Mechanismus ist es möglich (und notwendig), die vom Directory belegten Clusters fest zu reservieren.

Bytes 12 - 13 : Checksum Vector Size. Dieser Wortwert gibt die Länge des Checksum Vectors an. Dies ist eine Liste von Prüfsummenwerten, die EDS bei jedem Zugriff auf das Directory anlegt, wobei jeder Wert die Prüfsumme eines 128 - Bytes - Blocks ausmacht. Ändert sich die Prüfsumme bei einem Directory - Zugriff, wird ein Datenträgerwechsel erkannt und automatisch der Schreibschutz gesetzt. Dieser Vektor ist daher nur für wechselbare Datenträger von Interesse. Fest installierte Datenträger haben hier den Wert 0.

Bytes 14 - 15 : Anzahl der reservierten Spuren. Dieser Wortwert gibt die Startspur des Datenbereichs an. Somit ist es möglich, für Systemzwecke die äußeren Spuren eines Datenträgers zu reservieren.

Bytes 16 : Sektorgröße. Dieser Wert gibt die Größe eines physikalischen Sektors des Datenträgers an. Dieser Wert muß nicht unbedingt mit der tatsächlichen physikalischen Sektorgröße übereinstimmen. Der Wert stellt den Betrag dar, um den ein logischer Sektor (der 128 Bytes adressiert) nach rechts geschiftet werden muß, um die physikalische Sektorgröße zu erhalten. Es gelten dabei die folgenden Zusammenhänge:

Größe eines Sektors	Shift Factor
128 Bytes	0
256 Bytes	1
512 Bytes	2
1024 Bytes	3
2048 Bytes	4
4096 Bytes	5

Byte 17 : Sektormaske. Dies ist wiederum eine Maske, mit der der logische Sektor zu 128 Bytes maskiert wird, um die Position dieses Sektors innerhalb eines physikalischen Sektors zu bestimmen. Tatsächlich maskiert diese Maske genau die Bits, die beim Shiftvorgang des Sektors verloren gehen. Es gelten die folgenden Zusammenhänge:

Größe eines Sektors	Shift Mask
128 Bytes	0
256 Bytes	1
512 Bytes	3
1024 Bytes	7
2048 Bytes	15
4096 Bytes	31

Bytes 18 : Skew Factor. Wenn dieses Byte den Wert 1 hat, hat die Platte keinen Skew Factor. Hat dieses Byte den Wert 0, hat die Platte einen Skew Factor, der über eine Tabelle bestimmt wird. Diese Tabelle entzieht sich allerdings dem Direktzugriff des Programmierers.

Das höchstwertige Bit dieses Bytes hat eine besondere Bedeutung: ist dieses gesetzt, wird das Laufwerk als fest installiert betrachtet (siehe dazu auch Kapitel 5).

B.8. Tabelle der Hardware - Funktionsaufrufe

Aufruf	Versorgung	Rückkehr
0 System - Initialisierung	---	---
1 System - Reinitialisierung	---	---
2 Eingabestatus Tastatur	---	AL = 0 / Zeichen
3 Tastatur - Eingabe	---	AL = Zeichen
4 Bildschirm - Ausgabe	C = Zeichen	---
5 Drucker - Ausgabe	C = Zeichen	---
6 Zusatzgeräte - Ausgabe	C = Zeichen	---
7 Zusatzgeräte - Eingabe	---	AL = Zeichen
8 Kopf auf Spur 0 positionieren	---	---
9 Laufwerk anwählen	C = LW, E = Bit	HL = 0 / nicht 0
10 Spur anwählen	BC = Spur	---
11 Sektor anwählen	BC = Sektor	---
12 DMA - Adresse setzen	BC = Adresse	---
13 Datenblock lesen	---	AL = 0 / 1
14 Datenblock schreiben	C = Schreibcode	AL = 0 / 1 / 2
15 Druckerstatus	---	AL = 0 / OFFH
16 Sektor - Umsetzung	BC = Sektor	HL = Sektor
17 Ausgabestatus Bildschirm	---	AL = 0 / OFFH
18 Eingabestatus Zusatzgerät	---	AL = 0 / Zeichen
19 Ausgabestatus Zusatzgerät	---	AL = 0 / OFFH
20 Adresse Gerätetabelle	---	HL = ADRESSE
21 Gerät initialisieren	C = Gerätenummer	---
22 Laufwerksbereitschaft prüfen	C = Laufwerk	HL = Prüfergebnis
23 Bildschirmeditor aktivieren	C = Col, DE = Addr	---
24 Puffer schreiben	---	AL = 0 / 1 / 2
25 Daten verschieben	Register vorbesetzt	neue Register
26 Datum / Uhrzeit	C = Flag, DE = Addr	---
27 Grafik ansprechen	BC = Versorgung	---
28 Hex - Ausgabe	BC = Hexwert	---

```
*****
*                                     *
*                                     *
*                                     *
*                                     *
*                                     *
*****
```

Versorgung: ---

Rückkehr: --

Dieser Aufruf dient dazu, alle nötigen Kaltstart - Initialisierungen durchzuführen. Insbesondere werden dabei die interne Speicherwaltung sowie die Gerätetreiber neu initialisiert. Ferner wird der Kommando - Interpreter auf die RAM - Disk kopiert und der System Prompt ausgegeben. Die Verwendung dieser Funktion wird ausdrücklich nicht empfohlen, da einige dieser Initialisierungs - Routinen nur einmal laufen dürfen. Als Beispiel sei erwähnt, daß der Kommando - Interpreter nicht noch einmal auf die RAM - Disk kopiert werden kann, da dieser dort bereits existiert.

```

*****
*
*           1 - System - Reinitialisierung
*
*****

```

Versorgung: ---

Rückkehr: ---

Bei jedem Warmstart (d.h. Aufruf der EOS - Funktion 0 oder Sprung auf die Adresse 0000H) wird u. a. dieser Teil aufgerufen. Dadurch werden noch evtl. anstehende Datenpuffer rausgeschrieben und die Gerätetreiber in einen definierten Zustand versetzt.

Diese Funktion kann von einem Benutzerprogramm aus aufgerufen werden (obwohl der Nutzen zweifelhaft ist).

Beispiel:

```

LD      C,50      ; EOS - Funktionscode
LD      DE,ZUSATZ ; Zusatzversorgung
CALL    5         ; EOS aufrufen
...
ZUSATZ: DB      1      ; Hardware - Funktion
        DB      0
        DW      0
        DW      0
        DW      0

```

Hardware - Schnittstelle

```

*****
*
* 2 - Lastatur - Status prüfen
*
*****

```

Versorgung: ---

Rückkehr: AL = Statuscode

Die über den Bitvektor CONIN: angeschlossenen Eingabegeräte werden auf eine anstehende Eingabe überprüft. Falls eines der Geräte ein Zeichen zur Verfügung stellen kann, enthalten A und L das Zeichen, was eingelesen werden würde, andernfalls den Wert 00H.

Beispiel:

LD	C,50	; EOS - Funktionscode
LD	DE,ZUSATZ	; Zusatzversorgung
CALL	5	; EOS aufrufen
OR	A	; liegt ein Zeichen an?
JP	NZ,BEREIT	
...		
DB	2	; Hardware - Funktion
DB		
DB		
DB		
DB		
DB		

```

*****
*
*           3 - Zeichen von Tastatur einlesen
*
*****

```

Versorgung: --

Rückkehr: AL = Zeichen

Die über den Bitvektor CONIN: angeschlossenen Eingabegeräte werden auf eine anstehende Eingabe überprüft. Das erste Gerät, was ein Zeichen anzubieten hat, kommt dann zum Zuge. Dieses Zeichen wird eingelesen und in A und L zurückgegeben. Das Zeichen wird unkonvertiert übergeben, d.h. es werden alle 8 Bits übergeben.

Es wird so lange auf eine Eingabe gewartet, bis eines der Eingabegeräte bereit ist.

Beispiel:

```

LD      C,50          ; EOS - Funktionscode
LD      DE,ZUSATZ      ; Zusatzversorgung
CALL    5             ; EOS aufrufen
LD      (ZEICHEN),A    ; Zeichen ablegen
...
ZUSATZ: DB      3      ; Hardware - Funktion
        DB      0
        DW      0
        DW      0
        DW      0

```

```

LD      C,50      ; EOS - Funktionscode
LD      DE,ZUSATZ ; Zusatzversorgung
CALL    5         ; EOS aufrufen
...
DB      4         ; Hardware - Funktion
DB      0         ; Zeichen
DW      '*'
DW      0
DW      0
DW      0

```

Ausgabe des Zeichens '*' auf dem Bildschirm

Beispiel:

Das Ausgabegerät 0 ist von diesem Vorgang ausgeschlossen, da dieses unter EOS fest als Bildschirm definiert ist.

Das in C übergebene Zeichen wird auf allen über den Bitvektor CONOUT angeschlossenen Ausgabegeräten ausgegeben. Dieses Zeichen wird dabei nicht maskiert oder konvertiert. Es wird so lange gewartet, bis das Zeichen auf allen angeschlossenen Geräten ausgegeben werden konnte.

Ist eines der Geräte nicht bereit, wird eine durch die EOS - Funktion 73 einstellbare Zeitspanne gewartet. Ist diese abgelaufen, wird eine Fehlermeldung ausgegeben, die u. A. den physikalischen Namen des Gerätes enthält. Liefert zu diesem Zeitpunkt die Hardware - Funktion 2 einen Wert ungleich 0 zurück, wird das anstehende Zeichen eingelesen und der Ausgabevorgang abgebrochen, d.h. es findet ein Systemreset statt. Falls die Hardware - Funktion 2 den Wert 0 zurückliefert, wird der Wartezyklus wiederholt.

Rückkehr: ---

Versorgung: C = Zeichen

```

*****
*                                     *
*      4 - Zeichen auf Bildschirm ausgeben      *
*                                     *
*****

```

Hardware - Schnittstelle

Abschnitt B:


```
*****
*
*           5 - Zeichen auf Drucker ausgeben
*
*****
```

Versorgung: C = Zeichen

Rückkehr: ---

Das in C übergebene Zeichen wird auf allen über den Bitvektor LSTOUT angeschlossenen Ausgabegeräten ausgegeben. Dieses Zeichen wird dabei nicht maskiert oder konvertiert. Es wird so lange gewartet, bis das Zeichen auf allen angeschlossenen Geräten ausgegeben werden konnte.

Ist eines der Geräte nicht bereit, wird eine durch die EOS - Funktion 73 einstellbare Zeitspanne gewartet. Ist diese abgelaufen, wird eine Fehlermeldung ausgegeben, die u. A. den physikalischen Namen des Gerätes enthält. Liefert zu diesem Zeitpunkt die Hardware - Funktion 2 einen Wert ungleich 0 zurück, wird das anstehende Zeichen eingelesen und der Ausgabevorgang abgebrochen, d.h. es findet ein Systemrestart statt. Falls die Hardware - Funktion 2 den Wert 0 zurückliefert, wird der Wartezyklus wiederholt.

Das Ausgabegerät 0 ist von diesem Vorgang ausgeschlossen, da dieses unter EOS fest als Bildschirm definiert ist.

Beispiel:

Ausgabe eines Seitenvorschubs auf dem Drucker

	LD	C,50	; EOS - Funktionscode
	LD	DE,ZUSATZ	; Zusatzversorgung
	CALL	5	; EOS aufrufen
	...		
ZUSATZ:	DB	5	; Hardware - Funktion
	DB	0	
	DW	OCH	; ASCII-Steuerzeichen FF (Form Feed)
	DW	0	
	DW	0	

```

*****
*
*           6 - Zeichen auf Zusatzgerät ausgeben
*
*****

```

Versorgung: C = Zeichen

Rückkehr: ---

Das in C übergebene Zeichen wird auf allen über den Bitvektor AUXOUT: angeschlossenen Ausgabegeräten ausgegeben. Dieses Zeichen wird dabei nicht maskiert oder konvertiert. Es wird so lange gewartet, bis das Zeichen auf allen angeschlossenen Geräten ausgegeben werden konnte.

Ist eines der Geräte nicht bereit, wird eine durch die EOS - Funktion 73 einstellbare Zeitspanne gewartet. Ist diese abgelaufen, wird eine Fehlermeldung ausgegeben, die u. A. den physikalischen Namen des Gerätes enthält. Liefert zu diesem Zeitpunkt die Hardware - Funktion 2 einen Wert ungleich 0 zurück, wird das anstehende Zeichen eingelesen und der Ausgabevorgang abgebrochen, d.h. es findet ein Systemrestart statt. Falls die Hardware - Funktion 2 den Wert 0 zurückliefert, wird der Wartezyklus wiederholt.

Das Ausgabegerät 0 ist von diesem Vorgang ausgeschlossen, da dieses unter EOS fest als Bildschirm definiert ist.

Beispiel:

Ausgabe des Zeichens '?' auf dem Zusatzgerät

	LD	C,50	; EOS - Funktionscode
	LD	DE,ZUSATZ	; Zusatzversorgung
	CALL	5	; EOS aufrufen
	...		
ZUSATZ:	DB	6	; Hardware - Funktion
	DB	0	
	DW	'?'	; Zeichen
	DW	0	
	DW	0	

```

*****
*
*           7 - Zeichen von Zusatzgerät einlesen
*
*****

```

Versorgung: --

Rückkehr: AL = Zeichen

Die über den Bitvektor AUXIN: angeschlossenen Eingabegeräte werden auf eine anstehende Eingabe überprüft. Das erste Gerät, was ein Zeichen anzubieten hat, kommt dann zum Zuge. Dieses Zeichen wird eingelesen und in A zurückgegeben. Das Zeichen wird unkonvertiert übergeben, d.h. es werden alle 8 Bits übergeben.

Es wird so lange auf eine Eingabe gewartet, bis eines der Eingabegeräte bereit ist.

Beispiel:

```

LD      C,50          ; EOS - Funktionscode
LD      DE,ZUSATZ      ; Zusatzversorgung
CALL    5             ; EOS aufrufen
LD      (ZEICHEN),A    ; Zeichen ablegen
...
ZUSATZ: DB      7      ; Hardware - Funktion
        DB      0
        DW      0
        DW      0
        DW      0

```

```

*****
*
*           8 - Kopf auf Spur 0 positionieren
*
*****

```

Versorgung: ---

Rückkehr: ---

Der Schreib / Lesekopf des aktuellen Laufwerks wird auf Spur 0 positioniert. Diese Funktion hat lediglich das Setzen interner Variablen zur Folge.

Beispiel:

```

          LD      C,50          ; EOS - Funktionscode
          LD      DE,ZUSATZ     ; Zusatzversorgung
          CALL    5             ; EOS aufrufen
          ...
ZUSATZ:   DB      8             ; Hardware - Funktion
          DB      0
          DW      0
          DW      0
          DW      0

```

```
*****
*
*          9 - Laufwerk anwählen
*
*****
```

Versorgung: C = Laufwerk
E = Erstzugriffsbit

Rückkehr: HL = Fehlercode

Das in C übergebene Laufwerk wird für alle nachfolgenden Schreib/Lesezugriffe als Bezugslaufwerk definiert. C enthält dabei eine 0 für das Laufwerk A, eine 1 für das Laufwerk B usw. Das niederwertigste Bit in Register E (Bit 0) gibt an, ob die Anwahl ein Erstzugriff ist oder nicht; ist das Bit gelöscht, müssen die Parameter des Laufwerks neu bestimmt werden. In diesem Fall wird unter Umständen auch auf die Platte zugegriffen, um das Aufzeichnungsformat zu bestimmen. Ist Bit 0 in Register E gesetzt, werden lediglich die internen Variablen umgesetzt. Ein Plattenzugriff erfolgt dabei nicht.

Konnte die Laufwerksanwahl korrekt durchgeführt werden, enthält HL einen Wert ungleich 0. Ist hingegen das gewünschte Laufwerk nicht definiert oder trat bei der Bestimmung der Laufwerksparameter ein Fehler auf, enthält HL den Wert 0.

Das Erstzugriffs - Bit in Register E ist eine Erweiterung gegenüber CP/M 2.2 bzw. DiCOS.

Beispiel:

Anwahl des Laufwerks B ohne Neubestimmung der Parameter

```
LD      C,50          ; EOS - Funktionsnummer
LD      DE,ZUSATZ      ; Zusatzversorgung
CALL    5             ; Funktion ausführen
LD      A,H
OR      L             ; Laufwerk angewählt?
JP      Z,IOERROR      ; nein, Sprung
...
ZUSATZ: DB      9      ; Hardware - Funktionscode
        DB      0
        DW      'B'-'A' ; Laufwerk
        DW      1      ; Neubestimmung verhindern
        DW      0
```

```

*****
*
*           10 - Spur setzen
*
*****

```

Versorgung: BC = Spurnummer, ab 0 zählend

Rückkehr: ---

Die in BC enthaltene Zahl wird als Spur für den nächsten Plattenzugriff gesetzt. Diese Zahl beginnt bei 0.

Beispiel:

Setzen der Spur 3 für den nächsten Zugriff

	LD	C,50	; EOS - Funktionsnummer
	LD	DE,ZUSATZ	; Zusatzversorgung
	CALL	5	; Funktion ausführen
	...		
ZUSATZ:	DB	10	; Hardware - Funktionscode
	DB	0	
	DW	3	; Spur
	DW	0	
	DW	0	

```

*****
*
*           11 ~ Sektor setzen
*
*****

```

Versorgung: BC = Sektor, ab 0

Rückkehr: ---

Die in BC enthaltene Zahl wird als Sektornummer für den nächsten Plattenzugriff gesetzt. Diese Zahl beginnt bei 0. Sie kennzeichnet in den meisten Fällen nicht den physikalischen Sektor, sondern vielmehr einen logischen Block von 128 Bytes innerhalb der durch die Hardware - Funktion 10 gesetzten Spur. Unter Umständen wird der dazugehörige Sektor intern gepuffert.

EOS ruft vor dieser Funktion die Hardware - Funktion 16 auf, um eine eventuelle Umsetzung zwischen logischen und physikalischen Sektornummern zu erreichen.

Beispiel:

Setzen des Sektors 20 für den nächsten Zugriff

	LD	C,50	; EOS - Funktionsnummer
	LD	DE,ZUSATZ	; Zusatzversorgung
	CALL	5	; Funktion ausführen
	...		
ZUSATZ:	DB	11	; Hardware - Funktionscode
	DB	0	
	DW	20	; Sektor
	DW	0	
	DW	0	

```

LD      C,50      ; EOS - Funktionsnummer
LD      DE,ZUSATZ ; Zusatzversorgung
CALL    5         ; Funktion ausführen
...
ZUSATZ: DB 12     ; Hardware - Funktionscode
        DB 0      ; DMA - Adresse
        DM 0
        DM 0
        DM 80H
        DM 0

```

Setzen der DMA - Adresse 80H für Datentransfers

Beispiel:

Der in BC enthaltene Wert wird als Pufferadresse für alle zukünftigen Datentransfers definiert. Dieser Puffer muß eine Länge von 128 Bytes haben, um jeweils einen Datenblock aufnehmen zu können. Ist Multisector I/O gefordert, wird diese Funktion wiederholt von EOS aufgerufen.

Rückkehr: ---

Versorgung: BC = DMA - Adresse

```

*****
*                                     *
*      12 - DMA - Adresse setzen      *
*                                     *
*****

```

Hardware - Schnittstelle

Abschnitt B:


```

*****
*                                     *
*                               13 - Block lesen                               *
*                                     *
*****

```

Versorgung: ---

Rückkehr: AL = 0 oder 1

Der durch die Funktionen 9, 10 und 11 bestimmte Block von 128 Bytes Länge wird von der Platte in den durch Funktion 12 gesetzten Datenblock eingelesen. Dieser Vorgang hat nicht immer einen Plattenzugriff zur Folge, da EOS u. U. mehrere physikalische Sektoren intern puffert.

Wird der Block korrekt eingelesen, enthalten A und L den Wert 0. Verläuft der Plattenzugriff fehlerhaft, enthalten A und L den Wert 1. Dabei wird auch eine Fehlermeldung auf dem Bildschirm ausgegeben.

Beispiel:

```

LD      C,50          ; EOS - Funktionsnummer
LD      DE,ZUSATZ     ; Zusatzversorgung
CALL    5             ; Funktion ausführen
OR      A             ; fehlerfrei?
JP      NZ,IOERROR    ; Sprung bei Fehler
...
ZUSATZ: DB      13     ; Hardware - Funktionscode
        DB      0
        DW      0
        DW      0
        DW      0

```

```

*****
*
*           14 - Block schreiben
*
*****

```

Versorgung: C = 0, 1 oder 2

Rückkehr: AL = 0, 1 oder 2

Der durch die Funktionen 9, 10 und 11 bestimmte Block zu 128 Bytes wird aus dem durch die Funktion 12 eingestellten Datenpuffer auf die Platte geschrieben. Die Art des Schreibvorgangs kann dabei durch den Parameter in C beeinflusst werden:

- 0 - Allocated Write. Der interne Sektorpuffer muß vor dem Einfügen des neuen Blocks von der Platte eingelesen werden, da er bereits gültige Daten enthält. Der so gefüllte Sektorpuffer wird noch nicht auf die Platte zurückgeschrieben, sondern erst, wenn der Puffer anderweitig benötigt wird.
- 1 - Immediate Write. Dieser Modus verhält sich wie Modus 0, mit dem Unterschied, daß der Sektorpuffer sofort wieder auf die Platte zurückgeschrieben wird.
- 2 - Unallocated Write. Der Sektorpuffer muß vor dem Einfügen des Datenblocks nicht eingelesen werden, da er noch keine definierte Daten enthält.

Konnte der Schreibvorgang erfolgreich beendet werden, enthalten A und L den Wert 0. Trat ein Schreibfehler auf, enthalten A und L den Wert 1. Konnte der Sektorpuffer nicht geschrieben werden, weil der Datenträger schreibgeschützt war, enthalten A und L den Wert 2. Im Fehlerfall wird auch auf dem Bildschirm eine Meldung ausgegeben.

Beispiel:

Schreiben des eingestellten Sektors mit Buffer Flush

LD	C,50	; EOS - Funktionsnummer
LD	DE,ZUSATZ	; Zusatzversorgung
CALL	5	; Funktion ausführen
OR	A	; fehlerfrei?
JP	NZ,IOERROR	; Sprung bei Fehler
...		
ZUSATZ: DB	14	; Hardware - Funktionscode
DB	0	
DW	1	; mit sofortigem Buffer Flush
DW	0	
DW	0	

```

*****
*                                     *
*                               15 - Druckerstatus prüfen                *
*                                     *
*****

```

Versorgung: ---

Rückkehr: AL = 0 oder OFFH

Wenn alle über den Bitvektor LSTOUT: angeschlossenen physikalischen Ausgabegeräte bereit sind, enthalten A und L den Wert OFFH, andernfalls den Wert 0.

Beispiel:

```

LD      C,50          ; EOS - Funktionsnummer
LD      DE,ZUSATZ     ; Zusatzversorgung
CALL    5             ; Funktion ausführen
OR      A             ; bereit?
JP      NZ,BEREIT
...
ZUSATZ: DB      15          ; Hardware - Funktionscode
        DB      0
        DW      0
        DW      0
        DW      0

```

```

*****
*
*           16 - Blocknummer umrechnen
*
*****

```

Versorgung: BC = Blocknummer

Rückkehr: HL = umgerechnete Blocknummer

Die in BC übergebene Blocknummer wird nach den im DPB des aktuellen Laufwerks angegebenen Kriterien in eine physikalische Blocknummer umgerechnet. Diese Blocknummer wird in HL zurückgeliefert. EOS ruft diese Funktion unmittelbar vor der Funktion 11 (Sektor setzen) auf.

Die Beschreibung des DPB ist in dem entsprechenden Kapitel nachzulesen.

Beispiel:

Umrechnen des Sektors 3

	LD	C,50	; EOS - Funktionsnummer
	LD	DE,ZUSATZ	; Zusatzversorgung
	CALL	5	; Funktion ausführen
	LD	(SEKTOR),HL	; neuen Sektor ablegen
	...		
ZUSATZ:	DB	16	; Hardware - Funktionscode
	DB	0	
	DW	3	; Sektor - Nummer
	DW	0	
	DW	0	

```

*****
*
*           17 - Status der Bildschirmausgabe
*
*****

```

Versorgung: ---

Rückkehr: AL = 0 oder OFFH

Wenn alle über den Bitvektor CONOUT: angeschlossenen physikalischen Ausgabegeräte bereit sind, enthalten A und L den Wert OFFH, andernfalls den Wert 0.

Beispiel:

```

LD      C,50          ; EOS - Funktionsnummer
LD      DE,ZUSATZ     ; Zusatzversorgung
CALL    5             ; Funktion ausführen
OR      A             ; bereit?
JP      NZ,BEREIT
...
ZUSATZ: DB      17          ; Hardware - Funktionscode
        DB      0
        DW      0
        DW      0
        DW      0

```

```

*****
*
*               18 - Status des zusätzlichen Eingabekanals
*
*****

```

Versorgung: ---

Rückkehr: AL = 0 oder OFFH

Wenn eines der über den Bitvektor AUXIN: angeschlossenen physikalischen Eingabegeräte bereit ist, enthalten A und L den Wert OFFH, andernfalls den Wert 0.

Beispiel:

```

LD      C,50      ; EOS - Funktionsnummer
LD      DE,ZUSATZ ; Zusatzversorgung
CALL    5         ; Funktion ausführen
OR      A         ; bereit?
JP      NZ,BEREIT
...
ZUSATZ: DB      18      ; Hardware - Funktionscode
        DB      0
        DW      0
        DW      0
        DW      0

```

```

*****
*
*           19 - Status des zusätzlichen Ausgabekanals
*
*
*****

```

Versorgung: ---

Rückkehr: AL = 0 oder OFFH

Wenn alle über den Bitvektor AUXOUT: angeschlossenen physikalischen Ausgabegeräte bereit sind, enthalten A und L den Wert OFFH, andernfalls den Wert 0.

Beispiel:

```

LD      C,50          ; EOS - Funktionsnummer
LD      DE,ZUSATZ     ; Zusatzversorgung
CALL    5             ; Funktion ausführen
OR      A             ; bereit?
JP      NZ,BEREIT
...
ZUSATZ: DB      19          ; Hardware - Funktionscode
        DB      0
        DW      0
        DW      0
        DW      0

```

```

*****
*
*                20 - Adresse der Gerätetabelle
*
*****

```

Versorgung: ---

Rückkehr: HL = Tabellenadresse

Die Startadresse der Tabelle der physikalischen Geräte wird in HL zurückgeliefert. In der Implementation auf dem MZ-3500 ist das die Adresse 0FE00H.

Beispiel:

```

LD      C,50          ; EOS - Funktionsnummer
LD      DE,ZUSATZ     ; Zusatzversorgung
CALL    5             ; Funktion ausführen
LD      A,(HL)        ; erstes Zeichen aus Tabelle holen
...
ZUSATZ: DB      20          ; Hardware - Funktionscode
        DB      0
        DW      0
        DW      0
        DW      0

```



```

*****
*
*                21 - physikalisches Gerät initialisieren
*
*
*****

```

Versorgung: C = Gerätenummer (0 bis 15)

Rückkehr: ---

Das physikalische Gerät, dessen Position in der Gerätetabelle der Nummer in C entspricht, wird neu initialisiert. Diese Funktion dient dazu, Änderungen innerhalb der Gerätetabelle auch dem entsprechenden Gerät mitzuteilen. Das EOS - Hilfsprogramm \$DEVICE benutzt diese Funktion.

Beispiel:

Initialisieren des Geräts Nr. 2(RS232)

	LD	C,50	; EOS - Funktionsnummer
	LD	DE,ZUSATZ	; Zusatzversorgung
	CALL	5	; Funktion ausführen
	...		
ZUSATZ:	DB	21	; Hardware - Funktionscode
	DB	0	
	DW	2	; Gerätenummer
	DW	0	
	DW	0	

```

*****
*
*                22 - Laufwerk auf Bereitschaft prüfen
*
*****

```

Versorgung: C = Laufwerksnummer (0 bis 15)

Rückkehr: H = Bitvektor
L = 0 oder OFFH

Das Laufwerk, dessen Nummer in C übergeben wird, wird überprüft, ob es bereit ist, Daten zu senden bzw. zu empfangen. Eine 0 entspricht dabei dem Laufwerk A, eine 1 dem Laufwerk B etc.

Ist das Laufwerk bereit, enthält L den Wert OFFH und H einen Bitvektor mit der folgenden Bedeutung:

Bit 7 - Laufwerk ist ein nicht auswechselbares Speichermedium
Bit 6 - Laufwerk enthält eine doppelseitige Diskette

Ist das Laufwerk nicht bereit, enthält L den Wert 0. H ist dabei undefiniert.

Es erfolgt bei diesem Aufruf kein Plattenzugriff. Insbesondere erfolgt keine Formatbestimmung.

Beispiel:

Prüfen, ob das Laufwerk B bereit ist

	LD	C,50	; EOS - Funktionsnummer
	LD	DE,ZUSATZ	; Zusatzversorgung
	CALL	S	; Funktion ausführen
	OR	A	
	JP	NZ,BEREIT	
	...		
ZUSATZ:	DB	22	; Hardware - Funktionscode
	DB	0	
	DW	'B'-'A'	; Laufwerk
	DW	0	
	DW	0	

```

*****
*                                     *
*               23 - Bildschirmditor aktivieren               *
*                                     *
*****

```

Versorgung: C = Anfangsspalte
DE = Pufferadresse

Rückkehr: ---

Der zusätzliche Bildschirm - Editor wird bei diesem Aufruf angesprochen. DE enthält die Adresse eines Zeilenpuffers, der wie bei der EOS - Funktion 10 beschrieben, aufgebaut sein muß. C enthält die erste gültige Spalte des Bildschirms, beginnend ab 0. Die Funktion geht in den Edit Mode des Bildschirms, empfängt eine Zeile und trägt diese in den übergebenen Puffer ein.

Beispiel:

Einlesen einer Zeile ab der Bildschirmspalte 10

```

LD      C,50          ; EOS - Funktionsnummer
LD      DE,ZUSATZ     ; Zusatzversorgung
CALL    5             ; Funktion ausführen
...
ZUSATZ: DB      23     ; Hardware - Funktionscode
        DB      0
        DW      BUFFER ; Zeilenpuffer
        DW      9     ; ab Spalte 10 (zählt ab 0)
        DW      0
...
BUFFER: DB      80     ; Puffergröße
        DS      1     ; Anzahl empfangener Zeichen
        DS      80    ; Zeilenpuffer

```

```

LD      C,50
LD      DE,ZUSATZ
CALL    ...
ZUSATZ: DB      24
        DB      0
        DB      0
        DB      0
        DB      0
        ; EOS - Funktionsnummer
        ; Zusatzversorgung
        ; Funktion ausführen
        ; Hardware - Funktionscode

```

Beispiel:

Alle noch nicht geschriebenen Sektorpuffer werden sofort auf die Platte übertragen. Treten dabei Fehler auf, wird wie unter Funktion 14 (Block schreiben) verfahren.

Rückkehr: AL = 0, 1 oder 2

Versorgung: ---

```

*****
*
*      24 - Sektorpuffer schreiben
*
*****

```

Hardware - Schnittstelle

Abschnitt B:

```

*****
*
*                25 - Daten verschieben
*
*****

```

Versorgung: BC = Anzahl Bytes
 DE = Quelladresse
 HL = Zieladresse

Rückkehr: BC = 0
 DE = Quelladresse + Länge
 HL = Zieladresse + Länge

Ein Speicherbereich der Länge <BC> wird von der Adresse <DE> zur Adresse <HL> verschoben. Diese Funktion funktioniert genau wie der Z80 - Maschinenbefehl LDIR, mit dem Unterschied, daß die Inhalte von DE und HL vertauscht sind. Desweiteren findet die Verschiebung ausschließlich in Speicherbank 0 statt.

EOS benutzt diese Funktion, um seine Daten in den internen Arbeitsbereich hinein- und herauszuschieben.

Beispiel:

Verschieben von 500 Bytes von Adresse 4000H nach Adresse 5000H

	LD	C,50	; EOS - Funktionsnummer
	LD	DE,ZUSATZ	; Zusatzversorgung
	CALL	5	; Funktion ausführen
	...		
ZUSATZ:	DB	25	; Hardware - Funktionscode
	DB	0	
	DW	500	; Anzahl Bytes
	DW	4000H	; Quelladresse
	DW	5000H	; Zieladresse

```

*****
*
*               26 - Datum und Uhrzeit setzen und lesen
*
*****

```

Versorgung: C = 0 oder OFFH
DE = Zeitpuffer

Rückkehr: ---

Wenn C den Wert 0 enthält, wird die aktuelle Tageszeit in den durch DE bezeichneten Puffer eingetragen. Enthält C den Wert OFFH, wird die aktuelle Tageszeit aus dem Zeitpuffer übernommen. Der Zeitpuffer hat die folgende Struktur:

DW	DATE	; Julianisches Datum, Tag 1 = 1. Januar 1978
DB	HOURL	; Stunde im BCD - Format
DB	MINUTE	; Minute im BCD - Format
DB	SECOND	; Sekunde im BCD - Format

Beispiel:

Setzen des Datums 1. Juni 1983, 15 Uhr 30 Minuten und 55 Sekunden

	LD	C,50	; EOS - Funktionsnummer
	LD	DE,ZUSATZ	; Zusatzversorgung
	CALL	5	; Funktion ausführen
	...		
ZUSATZ:	DB	26	; Hardware - Funktionscode
	DB	0	
	DW	OFFH	; Setzen der Zeit
	DW	DATUM	; Zeitpuffer
	DW	0	
	...		
DATUM:	DW	1978	; 1. Juni 1983
	DB	15H	; Stunde
	DB	30H	; Minute
	DB	55H	; Sekunde

```
*****
*
*           27 - Grafik ansprechen
*
*****
```

Versorgung: BC = Grafikparameter

Rückkehr: ---

Dieser Funktionsaufruf verhält sich genau so wie die EOS - Funktion 115. Der dort übergebene Parameter wandert nach BC. Es wird allerdings keine Überprüfung des Opcodes durchgeführt, so daß zu empfehlen ist, für grafische Ein/Ausgabe die EOS - Funktion 115 zu benutzen.

Eine genaue Beschreibung der Grafikbefehle ist im Abschnitt C nachzulesen.

```

*****
*
*           28 - Daten sedezimal ausgeben
*
*****

```

Versorgung: BC = Wert

Rückkehr: ---

Der in BC übergebene Wert wird vierstellig sedezimal auf dem Bildschirm ausgegeben.

Beispiel:

Ausgabe der Zahl 10000 im Hexformat auf dem Bildschirm

```

LD      C,50          ; EOS - Funktionsnummer
LD      DE,ZUSATZ     ; Zusatzversorgung
CALL    5             ; Funktion ausführen
...
ZUSATZ: DB      28     ; Hardware - Funktionscode
        DB      0
        DW      10000  ; Zahl
        DW      0
        DW      0

```


Abschnitt C
die Grafik - Schnittstelle
(EOS - Funktion 115)

C.1. Einleitung

Der MZ-3500 hat eine eingebaute Grafik, die sich sehen lassen kann. Die Bedienung dieser Grafik ist unter EOS auf zwei Arten möglich: entweder über den Grafik - Interpreter oder über die EOS - Funktion 115. Die letztere Grafikschnittstelle ist Gegenstand dieses Abschnitts.

Die EOS - Funktion 115 erwartet in DE einen Zeiger auf die Grafik - Zusatzversorgung. Das erste Byte dieses Versorgungsblocks enthält dabei immer den Grafik - Befehlscode. Dieser Code kann sich zwischen 0 und 21 bewegen. Hinter diesem Befehlscode folgen die Parameter, die (bis auf das Definieren von Bit - Images) immer im Wortformat vorliegen müssen.

Schauen wir uns ein Beispiel an: Um den Bildpunkt mit den Koordinaten X = 100 und Y = 200 zu setzen, schreiben Sie:

```
LD      C,115          ; EOS - Funktion 115
LD      DE,SET          ; Zusatzversorgung
CALL    5              ; Funktion ausführen
...

SET:    DB      9        ; Befehlscode
        DW      100      ; X - Koordinate
        DW      200      ; Y - Koordinate
```

C.2. Die Definition des Bildschirms

Der Bildschirm hat eine Größe von 640 mal 400 Punkten, wobei sich X zwischen 0 und 639 und Y zwischen 0 und 399 bewegen darf. Die Koordinaten 0,0 bezeichnen dabei die linke obere Bildschirmecke. Es gibt zwei Video - Ausgänge, die getrennt bedient werden können. Es ist durchaus möglich, zwei verschiedene Grafiken zu zeichnen, da jeder Bildschirm - Ausgang so programmiert werden kann, daß er nur bestimmte Farbcodes wiedergibt. Man könnte den einen Bildschirm nur auf die Wiedergabe von Rot und den anderen Bildschirm nur auf die Wiedergabe von Grün programmieren, so daß mit den beiden Farben die beiden Bildschirme angesprochen werden können.

Das Zeichnen ist in 8 Grundfarben möglich, wobei jedoch durch geschicktes Zusammenstellen von Grundfarben beliebige Farbtöne erzeugt werden können. Die wählbaren Farben sind:

- 0 - schwarz
- 1 - blau
- 2 - rot
- 3 - magenta
- 4 - grün
- 5 - cyan
- 6 - gelb
- 7 - weiß

C.3. Die Definition von Bit Images

Es gibt eine Möglichkeit, Zeichen softwaremäßig zu definieren (die Funktion 20). Dort geben Sie ein Bitraster von 8 x 8 Bits an, das bei Aufruf des entsprechenden Zeichens ausgegeben wird. Diese Raster werden auch zum definierten Füllen von Bereichen verwendet. Die Bitmuster sind zum Teil schon vordefiniert, um das Erzeugen von Balkengrafiken zu vereinfachen. Die Definition sieht beispielsweise so aus:

```
Byte 1 :    0 0 0 1 1 0 0 0
Byte 2 :    0 0 1 0 0 1 0 0
Byte 3 :    0 1 0 0 0 0 1 0
Byte 4 :    0 1 1 1 1 1 1 0
Byte 5 :    0 1 0 0 0 0 1 0
Byte 6 :    0 1 0 0 0 0 1 0
Byte 7 :    0 1 0 0 0 0 1 0
Byte 8 :    0 0 0 0 0 0 0 0
```

Dies entspräche der Definition des Buchstabens A.

Eine andere Art der Definition kann für das Füllen von Flächen verwendet werden:

```
Byte 1 :    1 0 0 0 1 0 0 0
Byte 2 :    0 1 0 0 0 1 0 0
Byte 3 :    0 0 1 0 0 0 1 0
Byte 4 :    0 0 0 1 0 0 0 1
Byte 5 :    1 0 0 0 1 0 0 0
Byte 6 :    0 1 0 0 0 1 0 0
Byte 7 :    0 0 1 0 0 0 1 0
Byte 8 :    0 0 0 1 0 0 0 1
```

Bei einem Füllbefehl mit diesem Muster wird die Fläche schraffiert.

Für Markierungen, Schraffuren etc. sind die Bit - Images 0 bis 12 bereits fest vordefiniert. Sie können allerdings beliebig umdefiniert werden:

- 0 - 100 prozentige Füllung
- 1 - 75 prozentige Füllung
- 2 - 50 prozentige Füllung
- 3 - 25 prozentige Füllung
- 4 - 10 prozentige Füllung
- 5 - Schraffur schräg nach rechts
- 6 - Schraffur schräg nach links
- 7 - gekreuzte Schraffur
- 8 - senkrechte Striche
- 9 - waagerechte Striche
- 10 - ein stilisiertes "D&Z" (naja!)
- 11 - senkrechte Schlangenlinien
- 12 - waagerechte Schlangenlinien

C.4. Tabelle der Grafikfunktionen

Funktion	Ergebnis
0	Grafik initialisieren
1	Grafik löschen
2	Grafikteile löschen
3	Bildschirm 1 definieren
4	Bildschirm 2 definieren
5	Hintergrundfarbe setzen
6	Zeichenfarbe setzen
7	Zeichenmodus setzen
8	Schreibmaske setzen
9	Bildpunkt schreiben
10	Bildpunkt löschen
11	Linie zeichnen
12	Linie fortschreiben
13	Rechteck zeichnen
14	Kreis zeichnen
15	Kreisbogen zeichnen
16	Rechteckfläche füllen
17	reserviert
18	Zeichen schreiben
19	Zeichen fortschreiben
20	Bitmuster definieren
21	Hard Copy durchführen
22	Definieren eines Bildschirmaschnitts
23	Auslesen eines Bildschirmausschnitts
24	Laden eines Bildschirmausschnitts

Es sei noch darauf hingewiesen, daß durch EOS keine Prüfung der Parameter erfolgt. Unsinnige Parameter haben ebenso unsinnige Ergebnisse zur Folge.

```
*****
*
*           0 - Grafik initialisieren
*
*****
```

Versorgung: DB 0 ; Opcode

Der Grafiktreiber wird neu initialisiert.

```
*****
*
*           1 - Löschen der Grafik
*
*****
```

Versorgung: DB 1 ; Opcode

Die gesamte Grafik wird gelöscht.

```
*****
*
*           2 - Löschen von Grafikteilen
*
*****
```

Versorgung: DB 2 ; Opcode
 Dw X1,Y1 ; obere linke Koordinaten
 Dw X2,Y2 ; untere rechte Koordinaten
 Dw COLOR ; Farbe

Dieser Aufruf gestattet das selektive Löschen von Grafiken. Innerhalb des durch die Diagonal - Koordinaten X1,Y1 und X2,Y2 bezeichneten Rechtecks wird in allen Objekten die Farbkombination COLOR gelöscht, wobei COLOR als Kombination von drei Bits zu sehen ist mit der Bedeutung:

Bit 0 - Blauanteile, Bit 1 - Rotanteile, Bit 2 - Grünanteile

Die folgenden Wertebereiche sind definiert:

X - 0 bis 639
 Y - 0 bis 399
 COLOR - 0 bis 7 (als Bitmuster)

```

*****
*
*           3 - Definieren von Bildschirm 1
*
*
*****

```

```

Versorgung:      DB      3      ; Opcode
                  DW      BITS   ; Bitvektor

```

Der Bildschirm 1 wird für alle weiteren Zeichenoperationen definiert.
Die mitgelieferte Bitleiste enthält die folgenden Bits:

```

Bit 0 : Blau - Anteile der Grafik anzeigen.
Bit 1 : Rot - Anteile der Grafik anzeigen.
Bit 2 : Grün - Anteile der Grafik anzeigen.
Bit 3 : wenn gesetzt, wird ein Farbbildschirm benutzt, sonst ein
        Schwarzweiß - Bildschirm.
Bit 4 : wenn gesetzt, soll die normale Zeichenausgabe eingeschaltet
        bleiben.

```

```

*****
*
*           4 - Definieren von Bildschirm 2
*
*
*****

```

```

Versorgung:      DB      4      ; Opcode
                  DW      BITS   ; Bitvektor

```

Der Bildschirm 2 wird für alle weiteren Zeichenoperationen definiert.
Die mitgelieferte Bitleiste enthält die folgenden Bits:

```

Bit 0 : Blau - Anteile der Grafik anzeigen.
Bit 1 : Rot - Anteile der Grafik anzeigen.
Bit 2 : Grün - Anteile der Grafik anzeigen.
Bit 3 : wenn gesetzt, wird ein Farbbildschirm benutzt, sonst ein
        Schwarzweiß - Bildschirm.
Bit 4 : wenn gesetzt, soll die normale Zeichenausgabe eingeschaltet
        bleiben.
Bit 5 : wenn gesetzt, sollen auch die Zeichenattribute, wie Blinken,
        Inversdarstellung etc. dargestellt werden.

```

```
*****
*
*                               5 - Hintergrundfarbe wählen
*
*
*****
```

```
Versorgung:      DB      5      ; Opcode
                  DW      COLOR  ; Farbe
```

Die Hintergrundfarbe wird neu gesetzt. Diese Farbe ist unabhängig von der Grafik. Sie kann auch bei der normalen benutzung des Bildschirms eingeschaltet werden, wenn dieser im Farb - Modus betrieben wird (siehe Abschnitt D).

```
*****
*
*                               6 - Zeichenfarbe wählen
*
*
*****
```

```
Versorgung:      DB      6      ; Opcode
                  DW      COLOR  ; Farbe
```

Die Farbe für alle nachfolgenden grafischen Operationen wird neu gesetzt. Der Wert darf sich zwischen 0 und 7 bewegen.


```
*****
*
*           7 - Zeichenmodus einstellen
*
*****
```

```
Versorgung:      DB      7      ; Opcode
                  DW      MODE   ; Modus
```

Der Zeichenmodus wird definiert. Es gelten die folgenden Zuweisungen:

- 0 - überschreiben
- 1 - Farbe invertieren
- 2 - Farbe löschen (subtraktives Zeichnen)
- 3 - additive Farbmischung

```
*****
*
*           8 - Strichmaske definieren
*
*****
```

```
Versorgung:      DB      8      ; Opcode
                  DW      MASK   ; Bitvektor
```

Mit diesem Aufruf wird die Strichart für alle nachfolgenden Zeichenbefehle definiert. Die Grafikbefehle setzen ihre Vektoren aus einem Punktraster zusammen, das mit Hilfe dieser Funktion definiert werden kann. Es ist beispielsweise möglich, eine gestrichelte Linie durch das Bitmuster

1100110011001100 (binär)

zu definieren.

Der durch X, Y bezeichnete Bildpunkt wird gelöscht. Für X und Y gelten die folgenden Wertebereiche:

X - 0 bis 639
Y - 0 bis 399

Versorgung: DB 10 X, Y ; Opcode ; Koordinaten

* 10 - Bildpunkt löschen *

An den Koordinaten X, Y wird ein Punkt in der Zeichenfarbe gesetzt. Für X und Y gelten die folgenden Wertebereiche:

X - 0 bis 639
Y - 0 bis 399

Versorgung: DB 9 X, Y ; Opcode ; Koordinaten

* 9 - Bildpunkt setzen *

Dieser Aufruf kann zum Zeichnen von Polygonzügen verwendet werden.

X

- 0 bis 639

Y

- 0 bis 399

Es wird vom jetzigen Standort des (unsichtbaren) Grafik - Cursors bis zu den angegebenen Zielkoordinaten eine Linie gezogen. Die Farbe wird durch den Aufruf 6, die Art der Linie durch den Aufruf 8 definiert. Für X und Y gelten die folgenden Wertebereiche:

Versorgung:

DB

12

; Dpcode

DW

X,Y

; Zielkoordinaten

*

12 - Linie fortzuschreiben

*

*

X

- 0 bis 639

Y

- 0 bis 399

Vom Koordinatenpunkt X1, Y1 bis zum Koordinatenpunkt X2, Y2 wird eine Linie gezeichnet. Die Farbe wird durch den Aufruf 6, die Art der Linie durch den Aufruf 8 definiert. Für X und Y gelten die folgenden Wertebereiche:

Versorgung:

DB

11

; Dpcode

DW

X1,Y1

; Startkoordinaten

DW

X2,Y2

; Zielkoordinaten

*

11 - Linie zeichnen

*

*

```
*****
*
*           13 - Rechteck zeichnen
*
*****
```

Versorgung:	DB	13	; Opcode
	DW	X1,Y1	; obere linke Ecke
	DW	X2,Y2	; untere rechte Ecke

Die beiden Koordinatenpaare geben die obere linke bzw. die untere rechte Ecke eines zu zeichnenden Rechtecks an. Die Farbe wird durch den Aufruf 6, die Art der Linie durch den Aufruf 8 definiert. Für X und Y gelten die folgenden Wertebereiche:

X	- 0 bis 639
Y	- 0 bis 399

```
*****
*
*           14 - Kreis zeichnen
*
*****
```

Versorgung:	DB	14	; Opcode
	DW	X,Y	; Kreismittelpunkt
	DW	R	; Radius

Es wird ein Kreis mit dem Mittelpunkt X, Y und dem Radius R gezeichnet. Die Farbe wird durch den Aufruf 6, die Art der Linie durch den Aufruf 8 definiert. Für X und Y gelten die folgenden Wertebereiche:

X	- 0 bis 639
Y	- 0 bis 399

Es sollte darauf geachtet werden, daß sich der Kreis vollständig auf dem Bildschirm unterbringen läßt.

```
*****
*
*           15 - Kreisbogen zeichnen
*
*****
```

```
Versorgung:      DB      15      ; Opcode
                  DW      X,Y     ; Mittelpunkt
                  DW      R       ; Radius
                  DW      PHI1    ; Startwinkel
                  DW      PHI2    ; Endwinkel
```

Es wird ein Kreisbogen mit dem Mittelpunkt X,Y und dem Radius R gezeichnet. Der Start- sowie der Endwinkel des Bogens werden durch PHI1 und PHI2 bezeichnet. Die Werte für PHI1 und PHI2 werden in Grad angegeben, wobei eine 0 am weitesten in X-Richtung befindlichen Punkt des Kreises auf dem Bildschirm, d.h. den rechten Rand des Kreises, bezeichnet. Die Farbe wird durch den Aufruf 6, die Art der Linie durch den Aufruf 8 definiert. Der Winkel wird gegen den Uhrzeigersinn gerechnet, so daß ein Winkel von 90 Grad den oberen Rand des Kreises bezeichnet.

Für X und Y gelten die folgenden Wertebereiche:

```
X      - 0 bis 639
Y      - 0 bis 399
PHI    - 0 bis 359
```

```
*****
*
*           16 - Rechteck füllen
*
*****
```

```
Versorgung:      DB      16      ; Opcode
                  DW      X1,Y1   ; obere linke Ecke
                  DW      X2,Y2   ; untere rechte Ecke
                  DW      COLOR    ; Farbe
                  DW      IMAGE    ; Image - Nummer
```

Durch X1,Y1 und X2,Y2 werden die diagonalen Eckpunkte eines Rechtecks angegeben, das mit einem Muster gefüllt wird. Dieses Muster setzt sich aus einem Vielfachen eines durch die Funktion 20 definierten 8x8 - Bitmusters zusammen. Die Zeichenfarbe wird durch COLOR angegeben. Es gelten dabei die folgenden Wertebereiche:

```
X      - 0 bis 639
Y      - 0 bis 399
COLOR  - 0 bis 7
IMAGE  - 0 bis 126
```

```
*****
*
*           18 - Zeichen schreiben
*
*****
```

Versorgung:	DB	18	; Opcode
	DW	X,Y	; Startkoordinaten
	DW	CHAR	; Zeichen
	DW	ROT	; Schreibrichtung
	DW	FACTOR	; Vergrößerungsfaktor

Es wird das Zeichen CHAR ausgegeben. Der linke untere Eckpunkt des Zeichens wird durch X,Y bestimmt. Die Schreibrichtung des Zeichens wird durch ROT bestimmt. Der Vergrößerungsfaktor wird durch FACTOR bestimmt, wobei eine 0 dem Maßstab 1:1 entspricht, eine 1 dem Maßstab 1:2, eine 2 dem Maßstab 1:3 etc. Das Zeichen ist im ASCII - Code anzugeben.

ROT kann die folgenden Werte annehmen:

- 0 - normale Schreibrichtung
- 1 - um 45 Grad nach oben geneigt
- 2 - senkrecht nach oben (90 Grad)
- 3 - 135 Grad
- 4 - waagerecht nach links, auf dem Kopf (180 Grad)
- 5 - 45 Grad nach unten links, auf dem Kopf (225 Grad)
- 6 - senkrecht nach unten (270 Grad)
- 7 - 45 Grad schräg nach unten rechts (315 Grad)

Es gelten die folgenden Wertebereiche:

- X - 0 bis 639
- Y - 0 bis 399
- CHAR - 0 bis 127
- ROT - 0 bis 7
- FACTOR - 0 bis 15

```
*****
*
*          19 - Zeichen fortschreiben
*
*****
```

Versorgung:	DB	19	; Opcode
	DW	CHAR	; Zeichen

An der aktuellen Position des (unsichtbaren) Grafik - Cursors wird ein Zeichen ausgegeben. Das Format dieses Zeichens richtet sich nach den in der Funktion 18 vorgegebenen Werten.

- Diese Funktion darf nur als Folgeaufruf der Funktion 18 verwendet werden und dient dazu, komplette Strings auszugeben.

```

*****
*
*                20 - Bit - Image definieren
*
*
*****

```

```

Versorgung:      DB      20      ; Opcode
                  DW      CHAR    ; Zeichen
                  DB      BIT1    ; Bitmuster 1
                  ...
                  DB      BIT8    ; Bitmuster 8

```

Mit Hilfe dieser Funktion können softwaremäßig Zeichen definiert werden. Unter der Zeichenposition CHAR wird ein Bit - Image von 8 mal 8 Bildpunkten definiert, das mittels der Funktionen 18 und 19 ausgegeben werden kann. Desweiteren kann mit dieser Funktion das Füllmuster für die Funktion 16, ebenfalls in Form eines 8 mal 8 - Rasters, vorgegeben werden. CHAR kann Werte zwischen 0 und 126 annehmen.

Die Bit - Images 0 bis 12 sind bereits mit Füllmustern vordefiniert, um das Erstellen von Blockgrafiken zu erleichtern. Die Bit - Images 32 bis 126 beziehen sich auf das ASCII - Alphabet, das auch jederzeit umdefiniert werden kann.


```
*****
*
*           21 - Hard Copy durchführen
*
*****
```

```
Versorgung:      DB      21      ; Opcode
                  DW      COLOR   ; Farbe
```

Der Grafik - Bildschirm wird als Hard Copy auf dem SHARP - Ink Jet - Drucker ausgegeben. COLOR ist ein Bitvektor mit der folgenden Bedeutung:

```
Bit 0 - Blau - Anteile ausgeben
Bit 1 - Rot   - Anteile ausgeben
Bit 2 - Grün  - Anteile ausgeben
```

Somit kann auch eine nach Farben selektierte Hard Copy erreicht werden.

```
*****
*
*           22 - Bildschirmausschnitt definieren
*
*****
```

```
Versorgung:      DB      22      ; Opcode
                  DW      X1,Y1   ; obere linke Ecke
                  DW      X2,Y2   ; untere rechte Ecke
                  DW      COLOR   ; Farbe
```

Es wird ein Bildschirmausschnitt definiert, der anschließend in einer Datei abgelegt bzw. aus einer Datei eingelesen werden kann. X1,Y1 und X2,Y2 bezeichnen dabei die Eck - Koordinaten eines rechteckigen Bildschirmausschnitts, auf den sich die nachfolgenden Funktionsaufrufe beziehen. COLOR ist ein Bitvektor mit den folgenden Bitwerten:

```
Bit 0 - Blau - Anteile mit erfassen
Bit 1 - Rot   - Anteile erfassen
Bit 2 - Grün  - Anteile erfassen
```

```

*****
*
*                23 - Bildschirmausschnitt laden
*
*****

```

Versorgung: DB 23 ; Opcode

Dies ist ein Folgeaufruf der Funktion 22. Er lädt einen Grafik - Datenblock von 128 Bytes Länge aus dem aktuellen DMA - Puffer heraus in den Grafik - Bildschirm. In HL wird ein Wert zurückgeliefert: ist HL gleich 0, so ist der Transfer in den durch die Funktion 21 definierten Bildschirmausschnitt beendet. Ist HL ungleich 0, wird noch ein weiterer Block erwartet.

Zwischen dem Aufruf von Funktion 21 und dem wiederholten Aufruf dieser Funktion darf kein anderer Aufruf der EOS - Funktion 115 erfolgen. Wird diese Regel nicht befolgt, wird die Übertragung abgebrochen.

```

*****
*
*                24 - Bildschirmausschnitt auslesen
*
*****

```

Versorgung: DB 24 ; Opcode

Mit Hilfe dieser Funktion wird ein Datenblock von 128 Bytes Länge aus dem Grafikspeicher ausgelesen und in dem aktuellen DMA - Puffer abgelegt. In HL wird ein Wert zurückgeliefert; ist HL gleich 0, ist das Auslesen aus dem durch Funktion 22 bestimmten Bildschirmausschnitt beendet. Ist HL ungleich 0, sind noch weitere Datenblöcke zu übertragen.

Zwischen dem Aufruf von Funktion 21 und dem wiederholten Aufruf dieser Funktion darf kein anderer Aufruf der EOS - Funktion 115 erfolgen. Wird diese Regel nicht befolgt, wird die Übertragung abgebrochen.

Abschnitt D
die Bildschirm - Schnittstelle

D.1. Einleitung

Im Nachfolgenden wird die Schnittstelle zum Bildschirm beschrieben. Diese Schnittstelle "versteh" eine Reihe von Steuercodes. Fast alle diese Steuercodes fangen mit dem Sonderzeichen "ESCAPE" (dezimal 27) an, gefolgt von den entsprechenden Steuerzeichen. Ist im nachfolgenden Text daher beispielsweise von der Zeichenfolge "ESC A" die Rede, versteht sich dies als die Ausgabe des Steuerzeichens "ESCAPE", gefolgt von der Ausgabe des Buchstabens "A".

D.2. Die implementierten Normen

Um eine möglichst einfache Verwendung bereits bestehender Software zu ermöglichen, werden vom MZ-3500 die Steuercodes von zwei weit verbreiteten Standard-Terminals verarbeitet. Zwischen den beiden Möglichkeiten wird durch folgende Steuercodes umgeschaltet:

ESC 'A' = TeleVideo 950 Terminal (TV950)

ESC 'B' = Implementation des Lear Siegler ADM3A Terminals im CP/M auf dem MZ-80B.

Nach dem Kaltstart und dem Beenden von Programmen wird standardmäßig der TeleVideo-Mode eingeschaltet.

Alle Systemprogramme sind für den TeleVideo-Mode ausgelegt. Soll ein Anwenderprogramm im ADM3A-Mode arbeiten, dann muß dieser Mode im Programm jedesmal neu gesetzt werden (z.B. im Text der Startmeldung des Programms). Beim Setzen und späteren Rücksetzen des ADM3A-Modes wird der Bildschirm gelöscht. Wir empfehlen die Umstellung aller Programme auf den wesentlich komfortableren TeleVideo-Mode.

Spezifikationen:	TV950	ADM3A
=====	=====	=====
Zeilen pro Bildschirm	24	25
Zeichen pro Zeile	80	80
Status Zeile =>	ja (25. Zeile)	nein

D.3. Tabelle der vorhandenen Steuercodes

Abweichungen vom Standard sind mit einem Stern * gekennzeichnet. Nähere Erklärungen (=>) finden sich weiter unten im Text.

Der Einleitungscode (Leadin) ist fest als <ESC> (1BH) definiert.

Alle Angaben <x> und <y> sind jeweils als <x+20h> und <y+20h> zu verstehen; d.h. <blank> bezeichnet die Position 0.

Cursor-Funktionen:	TV950	ADM3A
-----	-----	-----
Linefeed	Ctrl-J	Ctrl-J / ESC '*'
Carriage Return	Ctrl-M	Ctrl-M / ESC '-'
neue Zeile (CR+LF)	-----	Ctrl-I / ESC '4'
Tabulator	Ctrl-I	Ctrl-I
Cursor home (0,0)	Ctrl-^	Ctrl-^ / ESC '>'
Cursor up	Ctrl-K	Ctrl-K / ESC '+'
Cursor down	Ctrl-V	* Ctrl-V / ESC '6'
Cursor left	Ctrl-H	Ctrl-H / ESC '('
Cursor right	Ctrl-L	Ctrl-L / ESC ''
Cursor Position setzen =>	ESC '=' <y> <x>	Ctrl-Ü / ESC '='
Cursor Position lesen =>	ESC '?'	Ctrl-Ö / ESC '<'

Erase-Funktionen:	TV950	ADM3A
-----	-----	-----
ganzen Bildschirm	Ctrl-Z / ESC '*'	Ctrl-Z / ESC ':'
Cursor bis Bildschirmende	ESC 'Y' / ESC 'y'	Ctrl-S / ESC '3'
Cursor bis Zeilenende	ESC 'T' / ESC 't'	Ctrl-R / ESC '2'
Cursor bis Seitenposition	* ESC '6' <y> <x>	-----
Cursor bis Zeilenposition	* ESC '5' <x>	-----

Insert und Delete:	TV950	ADM3A
-----	-----	-----
Zeichen einfügen	ESC 'Q'	Ctrl-N / ESC '.'
Zeile einfügen	ESC 'E'	Ctrl-O / ESC '/'
Zeichen löschen	ESC 'W'	Ctrl-P / ESC '0'
Zeile löschen	ESC 'R'	Ctrl-Q / ESC '1'

Attribute:	TV950	ADM3A
-----	-----	-----
inverse Einzel-Zeichen	ESC ')'	Ctrl-C / ESC '#'
normale Einzel-Zeichen	ESC '('	Ctrl-D / ESC '\$'
Zeichen Attribute =>	ESC 'G' <n>	* Ctrl-E / ESC '%'
Cursor Attribute =>	ESC '.' <n>	* Ctrl-F / ESC '&'

Bildschirm Modes:	TV950		ADM3A
-----	-----		-----
ganzer Bildschirm invers	ESC 'b'		Ctrl-A / ESC '!'
ganzer Bildschirm normal	ESC 'd'		Ctrl-B / ESC '"'
Bildschirm einblenden	* ESC 'l'	=>	* Ctrl-U / ESC '5'
Bildschirm ausblenden	* ESC 'O'	=>	* Ctrl-U / ESC '5'

Spezielle Funktionen:	TV950		ADM3A
-----	-----		-----
Glocke (Beep)	Ctrl-G		Ctrl-G / ESC '''
TV950 Mode initialisieren	* ESC 'A'		* ESC 'A'
ADM3A Mode initialisieren	* ESC 'B'		* ESC 'B'
Grafik-Interpreter einsch.	* ESC 'I'		* ESC 'I'
Tastatur u. Bildschirm init.	-----		ESC 'Ä'
CRT Farbmode setzen =>	* ESC 'C' <n>		* ESC 'C' <n>
CRT ASCII-mode setzen =>	* ESC '3' <n>		* ESC 'D' <n>
Tastatur Caps lock setzen=>	* ESC '4' <n>		* ESC 'E' <n>
Bildschirm Editor ein =>	* Ctrl-Y / ESC 'N'		* Ctrl-Y / ESC '9'
Monitor Mode ein/aus =>	* Ctrl-underline		* Ctrl-underline
Bildschirm Hardcopy =>	* ESC 'P'		* Ctrl-W / ESC '7'
Tasten umdefinieren =>	* ESC 'ö' <....>		* Ctrl-X / ESC '8'
Tastendefinition lesen =>	* ESC '\$'		* ESC '\$'
Tastatur sperren	ESC '#'		* ESC 'F'
Tastatur freigeben	ESC '\$'		* ESC 'G'
Uhr-Anzeige ein/aus =>	* ESC '2' <n>		-----

Status-Zeile (nur TV950):	TV950
-----	-----
System-Status-Zeile anzeigen	ESC 'h'
Benutzer-Status-Zeile anzeig.	ESC 'g'
System-Status-Zeile setzen	* ESC 'e' <n> <Text> <CR> =>
Benutzer-Status-Zeile setz.	* ESC 'f' (ESC 'G' <att>) <Text> <CR>

D.4. Nähere Beschreibung der einzelnen Sequenzen

D.4.1. Cursorposition setzen

TV950: <ESC> <'='> <y+20h> <x+20h>
y = 0...23 / x = 0...79

ADM3A: <Ctrl-Ü> <y+20h> <x+20h> oder <ESC> <'='> <y+20h> <x+20h>
y = 0...24 / x = 0...79

Die Positionsangaben y (Zeile) und x (Spalte) werden jeweils mit einem Offset von 20H angegeben.

Beispiel: <ESC> <'='> <Space> <Space> positioniert den Cursor auf die Home-Position (0,0).

D.4.2. Cursorposition lesen

TV950: <ESC> <'?'>

ADM3A: <Ctrl-Ü> oder <ESC> <'<'>

Es wird die Cursor-Position in gleicher Form als Tastencodes zurückgeliefert, wie sie auch gesetzt wird. Die Übertragung der Position wird durch <CR> abgeschlossen.

Beispiel: Der Cursor steht auf Position y = 5 und x = 4.
Folgende Codes werden zurückgeliefert:
<'%'> <'\$'> <CR>

D.4.3. Attribute für Zeichen setzen

Ein gesetztes Attribut hat so lange Gültigkeit, bis ein neues Attribut gesetzt wird, oder bis der Bildschirm neu initialisiert wird.

TV950: <ESC> <'G'> <n>

ADM3A: <Ctrl-E> <n> oder <ESC> <'%'> <n>

Von <n> werden nur die unteren 4 Bits mit folgender Bedeutung ausgewertet:

- Bit 0 = Begrenzungsstrich rechts vom Zeichen
- 1 = blinkende Zeichen
- 2 = inverse Zeichen
- 3 = unterstrichene Zeichen

Beispiel: <ESC> <'G'> <'0'> = keine Attribute
<ESC> <'G'> <'4'> = inverse Zeichen
<ESC> <'G'> <'<'> = invers blinkende Zeichen

Es sei noch darauf hingewiesen, daß TeleVideo die Attribute anders handhabt. Bei diesem Bildschirm bewirkt das Setzen eines Attributs, daß an der Cursorposition eine Leerstelle geschrieben wird und der Bildschirm bis zum nächsten Attribut bzw, bis zum Schirmende mit diesem Attribut dargestellt wird. Aus organisatorischen Gründen werden nach dem Setzen eines Attributs nur die danach geschriebenen Zeichen mit diesem versehen; der Bildschirm wird nicht automatisch mit dem Attribut versehen. Zusätzlich fällt das Schreiben der Leerstelle fort.

D.4.4. Cursor - Attribute setzen

Ein gesetztes Attribut hat so lange Gültigkeit, bis ein neues Attribut gesetzt wird, oder bis der Bildschirm neu initialisiert wird.

Nach einem Warmstart oder einer Bildschirm-Initialisierung erscheint der Cursor immer als stehender Block. Dieses Erscheinungsbild kann mit der folgenden Sequenz geändert werden:

TV950: <ESC> <'.'> <n>

ADM3A: <Ctrl-F> <n> oder <ESC> <'&'> <n>

n kann die folgenden (ASCII-) Werte annehmen:

- 0 = Cursor ausgeschaltet (nicht angezeigt)
- 1 = Cursor als blinkender Block
- 2 = Cursor als stehender Block
- 3 = Cursor als blinkender Strich (underline)
- 4 = Cursor als stehender Strich (underline)

Beispiel: <ESC> <'.'> <'1'> bewirkt die Anzeige des Cursors als blinkender Block.

D.4.5. Bildschirm ein- und ausblenden

Mit dieser Funktion kann die Zeichen-Darstellung unterdrückt werden. Alle Bildschirmfunktionen bleiben voll erhalten, der Bildschirminhalt wird aber erst angezeigt, wenn er wieder eingeblendet wird.

TV950: einblenden = <ESC> <'1'>
ausblenden = <ESC> <'0'>

ADM3A: einblenden = <Ctrl-U> <'1'> oder <ESC> <'5'> <'1'>
ausblenden = <Ctrl-U> <'0'> oder <ESC> <'5'> <'0'>

D.4.6. Bildschirm-Editor einschalten

Der Bildschirm-Editor ermöglicht ein Editieren über den gesamten Bildschirmbereich. Alle Cursorfunktionen und sonstige Steuer-Zeichen können verwendet werden. Der Edit Mode wird durch Eingabe von <CR> beendet, wodurch gleichzeitig die Zeile, in der der Cursor gerade steht, vollständig als Tasteneingabe, gefolgt von <CR>, zurückgeliefert wird. Der folgende Teil des Bildschirms wird gelöscht, damit Ausgaben der aufgerufenen Funktion übersichtlich bleiben.

TV950: <Ctrl-Y> <n+20H> oder <ESC> <'N'> <n+20H>

ADM3A: <Ctrl-Y> <n+20H> oder <ESC> <'9'> <n+20H>

n bezeichnet die Anzahl der Zeichen, die am Zeilenanfang ignoriert werden sollen, beispielsweise eine vorhergehende Ausgabe.

Diese Funktion erhöht den Bedienungskomfort besonders beim 'Line Input' von EOS.

D.4.7. Monitor Mode

Im Monitor Mode werden vom Bildschirm die Steuercodes nicht verarbeitet, sondern direkt als Control-Zeichen (umgekippte Zeichen) dargestellt. Diese Funktion ist besonders beim Testen von Software sehr nützlich.

Durch <Ctrl-Underline> wird der Monitor Mode ein- und auch wieder ausgeschaltet.

D.4.8. Hardcopy des Bildschirms

Der Bildschirminhalt wird direkt auf den Drucker übertragen. Im TV950-Mode wird die Statuszeile nicht mit ausgegeben.

TV950: <ESC> <'P'>

ADM3A: Ctrl-W / <ESC> <'7'>

D.4.9. Uhrzeit ein- und ausschalten

Im TeleVideo-Mode kann das Datum und die Uhrzeit in der Status-Zeile angezeigt werden. Für dieses Einblenden der Uhrzeit können verschiedene Modes gesetzt werden:

TV950: <ESC> <'§'> <n>

mit <n> = '0' - Uhr eingeschaltet
 = '1' - Uhr ausschalten durch <Ctrl-3> möglich
 > '1' - Uhr ausschalten durch <Ctrl-3> gesperrt

Das Einblenden der Uhrzeit kann auch durch die => Kontroll-Funktion <Ctrl-3> ein- und ausgeschaltet werden, wenn der angegebene Parameter '0' oder '1' ist.

D.5. Die Statuszeile im TV950-Mode

Die 25. Zeile ist im TV950-Mode als Informations-Zeile definiert und von den normalen Bildschirm-Funktionen abgetrennt. In dieser Zeile kann entweder eine System-Statuszeile oder eine frei definierbare Benutzer-Statuszeile angezeigt werden. Es kann jederzeit zwischen der System- und der Benutzer-Statuszeile umgeschaltet werden. Beide Zeilen sind völlig voneinander unabhängig.

D.5.1 Die System - Statuszeile

TV950: <ESC> <'h'> - System-Status-Zeile anzeigen

Die System-Status-Zeile (invers dargestellt) beinhaltet 8 Felder mit Informationen über den momentanen Zustand von EOS:

```
-----
!  0  !  1  !  2  !  3  !  4  !  5  !  6  !  7  !
-----
```

Feld '0' - Name des laufenden Programms
(des zuletzt vom Kommandointerpreter gestarteten Programms)

Feld '1' - Anzeige der standardmäßig angewählten Speichereinheit.

Feld '2' - Anzeige des angewählten Benutzerbereichs.

Feld '3' - Zustand der Tastatur und des Bildschirms:

```
'gesperrt' - alle Tastatur-Eingaben gesperrt
'Monitor'  - Bildschirm im => Monitor-Mode
'Key Def'  - Eingabe von => Tasten-Definitionen
'BenStat'  - Setzen der => Benutzer-Status-Zeile
'SysStat'  - Setzen der => System-Status-Zeile
'Graphik'  - Bildschirm im => Pseudo-Grafik-Mode
'Editieren' - spezieller => Editier-Mode aktiv
'Hard Copy' - => Bildschirm Hardcopy ausdrucken
```

Feld '4' - Hier erfolgt eine Anzeige, wenn EOS gerade eine Datei abarbeitet.

Feld '5' - Datum (abschaltbar)

Feld '6' - frei für Anwenderprogramme

Feld '7' - Uhrzeit (abschaltbar)

Alle Felder der System-Status-Zeile können auch von Programmen mit eigenen Meldungen beschrieben werden:

TV950: <ESC> <'e'> <n> <..Text..> <CR>

<n> = Nr. des Feldes ('0'..'7' / größer = System-Meldung)

<..Text..> = max. 9 Zeichen

<CR> = Ende des Textes

Solange die Uhr eingeschaltet ist, sollten die Felder '6' und '7' nicht vom Anwender benutzt werden. Wird für <n> die Nummer <'8'> angegeben, dann wird wieder die Original-Zeile des Systems erzeugt.

Beispiel: <ESC> <'e'> <'8'> <CR> = Status-Zeile von EOS generieren lassen und schließlich durch <ESC> <'h'> ausgeben.

D.5.2. Die Benutzer - Statuszeile

TV950: <ESC> <'g'> - Benutzer-Status-Zeile ausgeben

TV950: <ESC> <'f'> (<ESC> <'G'> <a>) <..Text..> <CR> - setzen

<a> = Attribut wie für Bildschirm (z.B. '2' = Blinken),
gilt aber für die gesamte Zeile

<..Text..> = max. 80 Zeichen Text

<CR> = Ende des Textes

Das Setzen eines Attributs ist optional. Wird die Zeichenfolge (<ESC> <'G'> <a>) nicht gesendet, dann wird automatisch invers als Attribut verwendet.

D.5.3. Unterdrückung der Statuszeile

Die Anzeige der Statuszeile kann völlig unterdrückt werden, indem folgende Codes gesendet werden:

<ESC> <'f'> <CR> <ESC> <'g'>

D.6. Die Tastatur

Die Tastatur ist in drei Funktions-Blöcke aufgeteilt:

1. alphanumerische Gesamt-Tastatur mit einigen Funktions-Tasten
2. numerische Zehner-Tastatur
3. frei definierbare Funktions-Tasten

Mit dem <PRO>-<OP>-Schalter kann die Tastatur gesperrt werden. Der Zustand der Tastatur wird im Status-Feld Nr.2 angezeigt.

In der Stellung <PRO> (programm) ist die Tastatur freigegeben.
In der Stellung <OP> (operation) ist die Tastatur gesperrt.

D.6.1. Die alphanumerische Gesamt-Tastatur

Die Tastatur entspricht der Norm DIN 2137 Teil 2. Die hier angeordneten Funktions-Tasten (TAB und Cursor-Steuerung) sind wie fast alle Funktions-Tasten frei definierbar. Fest vorgegeben sind nur die Tasten

<ENTER> = <CR> = 0Dh
<CMD> = <ESC> = 1Bh

Hinweis: Wundern Sie sich nicht, wenn Sie einige Control-Zeichen nicht finden; einige Tasten sind hardwaremäßig abweichend von der Beschriftung belegt:

<div> <div><</div> <div>^</div> <div>ü</div> </div>			
<div> <div>ü</div> <div>+</div> </div>		<div> <div>§</div> <div>Ä</div> </div>	
<div> <div>#</div> <div>ENTER</div> </div>		<div> <div>ü</div> <div>_</div> <div>ENTER</div> </div>	
Beschriftung und normale Belegung		Belegung auf der Control-Ebene	
(Ctrl-§)	ist auf	(Ctrl-ü)	
(Ctrl-Ä)	ist auf	(Ctrl-+)	
(Ctrl-ü)	ist auf	(Ctrl-<)	
(Ctrl-ü)	ist auf	(Ctrl-#)	
(Ctrl-^)	ist auf	(Ctrl-')	
(Ctrl-Underline)	ist auf	(Ctrl-Klammer)	

D.6.2. Die numerische Zehner-Tastatur

In diesem Block sind die Tasten <00> und <.> frei definierbar.

Die Tasten <0> bis <9> lösen, wenn sie gleichzeitig mit der <CTRL>-Taste gedrückt werden, die 'Kontroll-Funktionen' (dargestellt durch <Ctrl-0> bis <Ctrl-9>) aus. Folgende Funktionen sind verfügbar:

- <Ctrl-0> = Der Tastatur-Eingabepuffer, der bis zu 140 Zeichen zwischenspeichern kann, wird vollständig gelöscht.
- <Ctrl-1> = Wirkung der <SHIFT>- und <LOCK>-Tasten umdrehen. (Unabhängige Umschaltung zwischen Groß- und Kleinschreibung.)
- <Ctrl-2> = Umschaltung zwischen dem DIN-Zeichensatz mit Umlauten und dem ASCII-Zeichensatz mit eckigen Klammern (der ASCII-Zeichensatz ist nur mit dem speziellen EOS-Zeichengenerator-ROM verfügbar !)
- <Ctrl-3> = Einblendung der Uhr in der Status-Zeile ein- bzw. ausschalten.
- <Ctrl-4> = Es wird eine Hardcopy des Bildschirminhalts auf einen Drucker über die parallele Schnittstelle ausgeführt.
- <Ctrl-5> = Hardcopy der Graphik auf den SHARP Ink-Jet-Printer IO-700.
- <Ctrl-6> = Anzeige der Graphik ein- bzw. ausschalten.
- <Ctrl-7> = Start der direkten Tasten-Definition.

Alle Eingaben werden am Bildschirm protokolliert. Als erstes muß die gewünschte Taste gedrückt werden, die dann im Klartext angezeigt wird. Wird <ESC> (= <CMD>) gedrückt, dann wird die vom System vorgegebene Definition wieder hergestellt. Es kann ein Text mit bis zu 128 Zeichen eingegeben werden. Insgesamt steht ein Speicher für 300 Zeichen zur Verfügung.

Wird während der Definition <Ctrl-7> gedrückt, dann wird die Definition abgebrochen.

- <Ctrl-8> = Direkte Tasten-Definition abschließen. Der neue Text wird für die definierte Taste übernommen.
- <Ctrl-9> = Re-Initialisierung von Bildschirm und Tastatur.

D.6.3. Frei definierbare Funktionstasten

Alle Funktionstasten, bis auf <CMD> und <ENTER>, können vom Benutzer und seinen Programmen einzeln mit beliebigen Texten belegt werden. Dies geschieht entweder mit den => Kontroll-Funktionen <Ctrl-7> und <Ctrl-8> oder über Steuersequenzen, die vom Programm an den Bildschirm gesendet werden.

TV950: <ESC> <'ö'> <n> <.Text.> <Ctrl-Y> <Ctrl-Y> <Ctrl-Y>

ADM3A: <Ctrl-X> <n> <.Text.> <n> <.Text.> <Ctrl-Y> <Ctrl-Y>

<n> = Code der zu definierenden Taste.

<Text> = maximal 128 Zeichen Text pro Taste. Alle Control-Codes sind möglich. Den Codes <Ctrl-P> und <Ctrl-Y> muß ein zusätzliches <Ctrl-P> vorausgehen.

Es können mehrere Tasten nacheinander in einem Aufruf definiert werden. Jede Definition muß mit <Ctrl-Y> beendet werden. Die gesamte Definitionssequenz muß mit zwei aufeinanderfolgenden <Ctrl-Y> beendet werden. Für alle Tasten zusammen können max. 300 Zeichen gespeichert werden.

D.6.3.1. Steuer-Codes für die Tastendefinition

30H = '0' : alle Definitionen löschen.
 31H = '1' : momentane Definition merken.
 32H = '2' : gemerkte Definition wieder aktivieren.
 33H = '3' : Definition durch das System wieder herstellen.

ASCII Codes	Tasten		
-----	-----		
41H = 'A' :	<F1>	60h = `` :	<Ctrl-CMD>
42H = 'B' :	<F2>	61h = 'a' :	<Ctrl-F1>
43H = 'C' :	<F3>	62h = 'b' :	<Ctrl-F2>
44H = 'D' :	<F4>	63h = 'c' :	<Ctrl-F3>
45H = 'E' :	<F5>	64h = 'd' :	<Ctrl-F4>
46H = 'F' :	<F6>	65h = 'e' :	<Ctrl-F5>
47H = 'G' :	<F7>	66h = 'f' :	<Ctrl-F6>
48H = 'H' :	<F8>	67h = 'g' :	<Ctrl-F7>
49H = 'I' :	<F9>	68h = 'h' :	<Ctrl-F8>
4AH = 'J' :	<F10>	69h = 'i' :	<Ctrl-F9>
		6Ah = 'j' :	<Ctrl-F10>
4BH = 'K' :	<HOME>	68h = 'k' :	<Ctrl-HOME>
4CH = 'L' :		6Ch = 'l' :	<UP>
4DH = 'M' :	<INS>	6Dh = 'm' :	<DOWN>
4Eh = 'N' :	<-ENTER>	6Eh = 'n' :	<RIGHT>
4Fh = 'O' :	<CL>	6Fh = 'o' :	<LEFT>
50h = 'P' :	<RUN>		
51h = 'Q' :	<EDIT>		
52h = 'R' :	<DEB>		
53h = 'S' :	<Ctrl-DEB>		
54h = 'T' :	<BRK>		
55h = 'U' :	<Ctrl-BRK>		
56h = 'V' :	<OO>		
57h = 'W' :	<.> im numerischen Tastenfeld		
58h = 'X' :	<TAB>		

6.4. Vordefinition der Funktionstasten

Die meisten Funktions-Tasten werden vom System vordefiniert:

Taste	Definition	Funktion
fest definiert:		
<CMD>	1Bh = <ESC>	Leadin für Bildschirmcodes
<ENTER>	0Dh = <CR>	Zeile an EOS übergeben
umprogrammierbar:		
<TAB>	09h = <TAB>	horizontaler Tabulator
<INS>	0Fh = <Ctrl-O>	Zeichen einfügen
	7Fh = <RUBOUT>	Zeichen löschen
<DEB>	1Fh = <Ctrl-Underln>	Monitor Mode ein/aus
<BRK>	13h = <Ctrl-S>	Bildschirmausgabe anhalten
<Ctrl-BRK>	03h = <Ctrl-C>	Programm abbrechen
<UP>	08h = <Ctrl-K>	Cursor eine Zeile rauf
<DOWN>	16h = <Ctrl-V>	Cursor eine Zeile runter
<LEFT>	08h = <Ctrl-H>	Cursor ein Zeichen links
<RIGHT>	0Ch = <Ctrl-L>	Cursor ein Zeichen rechts
<HOME>	1Eh = <Ctrl-^>	Cursor auf 0,0 (Homeposition)
<Ctrl-HOME>	1Ah = <Ctrl-Z>	Bildschirm löschen
<CL>	18h = <Ctrl-X>	Eingabezeile löschen
<-ENTER>	<ESC> <'I'>	Bis Zeilenende löschen
<OO>	<'O'> <'O'>	Doppel-Null
<.>	<'>'>	Dezimalzeichen

D.6.4. Beispiel für eine Tastenprogrammierung

Die Programmierung für einen Texteditor könnte z.B. so aussehen:

```

EOS      EQU      5           ; EOS - Systemaufruf
ESC      EQU      27          ; ASCII für <ESC>
CTLY     EQU      25          ; <Ctrl-Y> = Trennzeichen
CTLP     EQU      16          ; <Ctrl-P> = Leadin-Zeichen

        ORG      100H         ; Start durch EOS
        JP      INIT          ; init keydefs vor dem Start
        JP      RESET         ; reset keydefs vor dem Ende

        ORG      MEMORY       ; freier Speicherraum

INIT:    LD      DE,INITBL     ; Tabelle mit neuen Definitionen
        CALL    SETDEF        ; Tasten umdefinieren
        JP      START         ; normale Startadresse

RESET:   LD      DE,RESTBL     ; Code für Re-Initialisierung
        CALL    SETDEF        ; Tastendefinition zurücksetzen
        JMP     0             ; und zurück zu EOS

SETDEF:  PUSH    DE           ; Tabellen-Adresse retten
        LD      C,109         ; EOS: Console Mode
        LD      DE,0FFFFH     ; alten Wert
        CALL    EOS           ; lesen
        LD      (OLDMOD),HL    ; und retten
        LD      C,109         ; EOS: Console Mode
        LD      DE,0FH        ; keine Umwandlungen
        CALL    EOS           ; setzen
        POP     DE           ; Adresse zurückholen
        LD      C,111         ; EOS: Print block
        CALL    EOS           ; Tabelle an Bildschirmtreiber
        LD      DE,(OLDMOD)    ; alten Console Mode
        LD      C,109         ; EOS: Console Mode
        CALL    EOS           ; wieder setzen
        RET

```

```

OLDMOD: DW      0                      ; alten Console Mode retten

INITBL: DW      INIMSG,INILEN          ; Adresse und Länge
INIMSG: DB      ESC,'ö'                ; start keydef
        DB      '1',CTLY               ; alte Def. retten
        DB      '0',CTLY               ; alles löschen
        DB      'o',8,CTLY              ; <- = ^H
        DB      'n',4,CTLY              ; -> = ^D
        DB      'l',5,CTLY              ; UP = ^E
        DB      'm',24,CTLY             ; DOWN = ^X
        DB      'K',ESC,'8V',13,CTLY    ; HOME = top screen
        DB      'k',ESC,'-8V',13,CTLY   ; ^HOME = bottom screen
        DB      'O',CTLP,CTLY,CTLY      ; CL = ^Y
        DB      'L',127,CTLY            ; DEL = RUBOUT
        DB      'M',15,CTLY             ; INS = ^O
        DB      'N',14,CTLY             ; -ENTER = insert line
        DB      'W',',',CTLY            ; DECIMAL POINT = Komma
        DB      'X',9,CTLY              ; TAB
        DB      'V','00',CTLY           ; 00
        DB      CTLY                   ; Tabellenende
INILEN EQU      $-INIMSG

RESTBL: DW      RESMSG,RESLEN
RESMSG: DB      ESC,'ö'
        DB      '2',CTLY,CTLY          ; alte Definition setzen
RESLEN EQU      $-RESMSG

END

```