

Bedienungshandbuch

STAR-MON

**Ein professioneller Monitor
für alle CPC's**

© by STAR-DIVISION 1986

Cassettenversion mit Diskversion, AMSDOS + CP/M
Zeilenassembler, Disassembler

**STAR-DIVISION · Uelzener Straße 12 · 2120 Lüneburg
Telefon (041 31) 40 25 50 u. 4 71 22**

Wichtige Hinweise zur Version V1.2Laden der Cassette

Cassette einlegen, Starttaste am Recorder drücken, und mit dem Basic-Befehl RUN starten. Der Monitor wird automatisch geladen und startet sich selbst. Ab Zählwerk Nr. 90 ist das Programm nochmals abgespeichert. Hier aber mit der hohen Schreibgeschwindigkeit (2000Bd.).

Memory wird automatisch auf die Adresse &7000 gelegt um den Monitor vor Überschreiben durch Basicprogramme zu schützen.

Speicherbelegung

Vom Monitor wird der Adressbereich &7000 &9800 belegt. Dieser Bereich muß immer geschützt bleiben.

Rückkehr zu BASIC

Vom Monitor kann mit dem B Befehl wieder zu Basic zurückgekehrt werden, und von Basic mit MON zum Monitor.

Selbstschutz des Monitors

Bei allen Funktionen die vom Monitor aus benützt werden, z. B. FillMemory,, schützt sich der Monitor selbst gegen Überschreiben. Dies ist jedoch nicht für eigene Z80Programme gegeben. Der Zweitregistersatz des Z80 sollte nicht benützt oder mit dem Registerbefehl verändert werden, da dies einen Systemabsturz zur Folge haben könnte.

Druckerbetrieb

Ein Parallelschalten des Druckers zum Bildschirm ist mit dem gleichzeitigen Druck auf die Taste CTRL und P möglich. Abgeschaltet wird in gleicher Weise. Einige Fehlermeldungen die auf dem Bildschirm erscheinen werden jedoch nicht mitgedruckt (z.B. Syntax Error).

Diskettenversion

Es gelten sinngemäß die Vorbemerkungen der Cassettenversion. Auf der Diskette befindet sich eine CP/M Version, sowie eine AMSDOS-Version des Monitors. Bei der CP/M Version kann über die Parameterfunktion auch auf Kassette umgeschaltet werden. Somit ist ein Überspielen von Kassette zu Disk und umgekehrt möglich. Die CP/M Version setzt das CP/M Betriebssystem auf der Diskette voraus, dieses befindet sich nicht auf der DISK !. Starten des MONITORS von Basic aus unter AMSDOS mit RUN MON ENIER. Für die CP/M Version gilt: erst Sytem starten, dann Programm mit CP/MON ENIER aufrufen. Es empfiehlt sich als erstes eine Sicherungskopie anzulegen.

Einleitung

Der Ihnen vorliegende Monitor, ist nicht nur ein Monitorprogramm, sondern beinhaltet auch einen zeilenorientierten Assembler, sowie einen Disassembler. Der Einsatzbereich liegt im Auslesen von eigenen Programmen, oder dem Analysieren und Untersuchen von fremden Programmen. Mit Hilfe des Zeilen-Assemblers ist natürlich auch ein Schreiben von Programmen möglich, wenn auch in begrenztem Umfang. Für größere Programme ist es sinnvoller mit einem Disk-Assembler zu arbeiten. Trace-Funktion, Disassembler und das Setzen von Breakpoint's sind natürlich exzellente Möglichkeiten, das Verhalten von Programmen zu untersuchen. Hingewiesen werden muß noch, daß alle Zugriffe auf den Monitor selbst verhindert sind. Wundern Sie sich nicht, wenn z. B. die Listingfunktion mit den Adressen des Monitors nicht funktioniert. Auch ist ein zweiter Kaltstart des Monitors nicht möglich. Es muß über die BSI Adresse 0030H wieder eingeschrieben werden. Bei allen Funktionen die vom Monitor aus benützt werden, schützt sich der Monitor selbst gegen ein Überschreiben.

Der Assembler

Der Z80, wie auch alle anderen Mikroprozessoren, verarbeitet Anweisungen, die in binärer Form in dem Speicherbereich vorliegen müssen, den er als Programmbereich adressiert. Die Erstellung des binären Programms kann auf vielfältige Weise erfolgen, die maschinennächste Form ist die Eingabe im hexadezimalen Format. z. B. über die Memory-Funktion dieses Monitorprogrammes. Da es nun aber nicht jedermanns Sache ist, die hexadezimale Codierung eines Befehles im Kopf zu haben, gibt es für jeden Befehl ein Merkwort. Dieses Merkwort wird als Mnemonic bezeichnet. Die Aufgabe eines Assemblers ist es diese Mnemonic in den Maschinencode zu übersetzen. Der im Monitorprogramm enthaltene Assembler ist ein zeilenorientierter Assembler, der jede Anweisung sofort, nach Abschluß mit der Enter-Taste, in den Maschinencode übersetzt. Innerhalb einer Zeile können Sie mit der 'BS'-Taste den Text noch korrigieren. Dieser Assembler assembliert Mnemonic's in der ZILD0 / MOSTEK - Notation. Es gibt noch eine andere Z80 Schreibweise, die auf den Mnemonic's für den Intel 8080 aufbaut und nur für die erweiterten Z80 Befehle Original-Mnemonic's verwendet. Diese Form wird nicht unterstützt.

Der Disassembler

Ein Disassembler ist ein Programm, welches einen vorhandenen Maschinencode (Objektcode) Byte für Byte nach zulässigen Befehlscodes des Z80 Prozessors absucht und daraus einen äquivalenten Programm mit symbolischen Befehlen (Mnemonic's) erzeugt. Dies ist also gerade die Umkehrung der Arbeitsweise des Assemblers, der die Mnemonic in ein ablauffähiges Programm übersetzt. Die einzige Information, die der Disassembler aus dem Maschinencode entnehmen kann, ist der hexadezimale Wert jedes Bytes. Daraus muß er selbstständig alle weiteren, zum Aufbau der Mnemonic erforderlichen, Informationen ableiten. Dabei ist es wichtig, an welcher Stelle des Maschinencodes mit

der Bearbeitung begonnen wird. Dies soll mit folgendem Beispiel belegt werden:

Maschinencode Z80:

a b
3E C3 CD 33 01 1E 0A

Beginnt der Disassembler an den mit a bzw. b markierten Stellen, so liefert er folgende Mnemonic:

Bei a:		bei b:
0100 3E C3	LD A,0C3	0101 C3 CD 33 JP 33DD
0102 CD 33 01	CALL 0133	0104 01 1E 0A LD BC,0A1E
0105 1E 0A	LD B,0A	.

Die beiden erzeugten Mnemonic's sind also völlig verschieden. Der Grund dafür ist, daß der Disassembler das bei der Adresse a und bzw. b vorgefundene Byte als Operationscode eines Befehls interpretiert und in einer Tabelle nachsieht, welcher Befehl es ist und ob dieser aus 1, 2, oder 3 Byte besteht. Hierdurch werden die folgenden Byte des nächsten Befehls als 1-Byte- oder 2-Byte- Argument interpretiert. Es ist also sehr wichtig, daß der Benutzer die Startadresse für die Disassemblierung richtig angibt. Außerdem muß bedacht werden, daß innerhalb eines Programms auch Datenbereiche liegen können die keinen Maschinencode darstellen. Ein Disassemblieren in diesen Bereichen ergibt natürlich völligen Unsinn.

Der Monitor

Ist ein Programm erst einmal geschrieben, dann muß es ausgiebig getestet werden. Kaum ein Programm ist jemals völlig fehlerfrei. Nun ist es nicht unbedingt empfehlenswert den neuen Objektcode "blind" zu laden und auf Verdacht laufen zu lassen. Die Gefahr, daß durch Fehlfunktionen etwas Wichtiges zerstört werden könnte ist zu groß. Auch besteht bei einem Systemabsturz kein Hinweis darauf, was eigentlich schiefgelaufen ist. Die Frage nach der fehlerhaften Stelle läßt sich ohne ein geeignetes Softwarewerkzeug nur sehr schlecht beantworten. Ein solches Werkzeug ist dieses Monitorprogramm.

Grundaufgaben:

- ein Programm laden (von Cassette oder Disk)
- das Programmverhalten untersuchen.
- den Speicherinhalt anzeigen und ändern
- das Programm sichern (auf Cassette oder Disk)

Nach dem Laden des zu testenden Programmes gibt es zwei Möglichkeiten das Programmverhalten zu untersuchen.

- Abarbeitung in Einzelschritten (Trace)
- Abarbeitung bis zu einem Haltepunkt (Breakpoint)

Die erste ist die sicherste, aber auch die langsamste Methode. Die Zweite hat den Nachteil, daß ein Programmabsturz vor dem Break vorkommen kann. Sinnvollerweise wird man beide Möglichkeiten, je nach Situation verwenden. Im Trace-Betrieb, in unserem Monitor auch Next genannt, wird jeder Befehl einzeln abgearbeitet. Es werden alle Register des Z80 Prozessors angezeigt, sowie der Maschinencode und die Mnemonic des abgearbeiteten Befehls.

Zur Untersuchung des Programmablaufs kann man natürlich auch die Break-Funktion verwenden. Es ist die Goto-Funktion zu benutzen, mit deren Hilfe wird die Startadresse des zu testenden Programms, sowie die Adresse bei welcher sich der Monitor wieder "melden" soll, eingegeben. Nach dem Start läuft das Programm in Echtzeit bis zu dem gesetzten Break (Stop-Punkt), und meldet sich dann wieder mit der gleichen Anzeige der Register wie bei der Trace-Funktion. Natürlich können mehrere Breakpoints gesetzt werden, auch wird die komplette Verwaltung der Breakpoints vom Monitor durchgeführt. Sehen Sie hierzu die Beschreibung des GOTO Befehls sich an.

Unser Monitor verfügt über eine Vielzahl von Möglichkeiten. Die anschließend in Kurzform beschreiben werden. Es empfiehlt sich jedoch, jeden Befehl einmal mit Hilfe der ausführlichen Befehls-Beschreibung, auszuprobieren.

Übersicht

- Assembler Speichereingaben in Assembler Schreibweise.
- Basic Rückkehr zu Schneider Basic. (Cass. Vers.)
- CPM Warmstart CP/M System (Disk Version)
- Disassembler Lesen von Speicherstellen in Assembler Schreibweise.
- Editor Speichereingaben in ASC II Code (Texte).
- Fill Memory Füllen des Speichers mit einem konstanten Wert.
- Goto Setzen der Breakpoints und Starten eines Programms zum Zwecke des Testens.
- Hexadezimal Rechnen mit Hexadezimalen Zahlen.
- IN/OUT Port Einlesen von einem Port.
- IN/OUT Port Schreiben zu einem Port.
- Speichertest Lokalisieren defekter Speicherstellen, bzw. System überprüfen.
- Laden Laden eines Programms von der Cassette oder Diskette in den Arbeitsspeicher.
- Inhaltsverz. Listen aller Programmnamen, welche sich auf Cassette befinden.
- Memory Listing des Speichers in Hex und ASC II.
- Memory Speichereingaben in Hex (Byteweise).
- Next Nächster Befehl, auch Trace, Singstep oder Debugging (Entwanzen) genannt.
- Offset Verrechnen einer Offset Adresse in Bezug auf alle Adreßeingaben.
- Save Abspeichern eines Programms auf Cassette/Disk.
- Trans Kopieren, Verschieben von Speicherbereichen.
- Umschaltung Umschalten der Speicherbank (ROM-RAM usw.).
- Vergleichen Vergleichen von Speicherbereichen.
- Suchen Suchen nach Zeichenfolgen in Hex.
- Suchen Suchen nach Zeichenfolge in ASC II.
- Register Register anzeigen.
- Register Register ändern.
- Parameter Schreibgeschwindigkeit der Cassette ändern. Disk/Kass. Umschaltung, Restart ändern.
- Drucker Zuschalten des Druckers zur Ausgabe.
- ? Help-Funktion, es werden alle Befehle gelistet.

Erläuterung für im Text verwendete Abkürzungen:

'ESC'	Escape-Taste.	Steuerzeichen = 1BH
'CR'	Carriage return - Taste.	Steuerzeichen = 0DH =ENTER beim CPC464
'SPACE'	Leerschritt-Taste.	Steuerzeichen = 20H
'BS'	Backspace-Taste.	Steuerzeichen = 08H
'DEL'	Delete-Taste	Steuerzeichen = 7FH
+ 'CR'	Es muß zusätzlich die Taste 'CR' betätigt werden. Nicht die Plus-Taste ist hier gemeint!.	
'CTRL'	Control-Taste	muß gleichzeitig mit einer weiteren Taste gedrückt werden.

FUNKTION : Assembler
 AUFRUF : a
 ZWECK : Speichereingaben in Assembler Schreibweise

```

-----
DIALOG DES MONITORS   EINGABEN   KOMMENTAR
>                   a           Aufruf des Assemblers

>Assembler
Anfangsadresse      : 100       Adresse ab welcher der
                           Code abgelegt wird.
Befehl eingeben     : ld a,OFF   Mnemonic eingeben

0100 3E FF LD A,FF     Maschinencode 3E FF
                           wird bei Adresse 100H
                           abgelegt und die akt.
                           Adresse um 2 Byte
                           erhöht.

Befehl eingeben     : 'ESC'     ESC Taste bewirkt
                           Abbruch der Funktion.

>
  
```

Beispiel für eine Eingabe mit Kommentar:

```

                                ld a,OFF ;Kommentar
0100 3E FF LD A,FF             ;Kommentar  Bildschirmanzeige.
  
```

Beschreibung:

Dieser Befehl dient der Eingabe von Programmen in den Speicher in Assemblerschreibweise. Die Assembler-Mnemonic wird in den Maschinencode übersetzt und in der aktuellen Adresse abgelegt. Anschließend wird der Adresszeiger entsprechend der Befehlslänge, in Byte, erhöht. Die Eingabe kann in Groß- oder Kleinschreibung erfolgen. Zu beachten ist, daß die Dezimalzahlen A-F nicht linksbündig geschrieben werden dürfen, es ist eine '0' voranzustellen. Beendet wird die Funktion mit der 'ESC' - Taste. Mit der 'BACKSPACE' - Taste dürfen Fehleingaben korrigiert werden, mit 'CR' wird die Eingabe abgeschlossen. Nach Abschluß der Eingabe überschreibt der Assembler die eingegebene Zeile und zeigt die Adresse, den Maschinencode, sowie den Befehl (in Großschreibung) an. Bei unbekanntem Befehlen oder falscher Schreibweise erscheint die Fehlermeldung 'SYNTAX ERROR'. Befehl erneut eingeben! Bei der Eingabe kann hinter den Assemblerbefehl ein Kommentar gesetzt werden, welcher mit einem ';' beginnen muß. Dieser Kommentar wird bei parallelgeschaltetem Drucker auf diesem ausgegeben. Als Besonderheit muß noch angegeben werden, daß 'JR' mit absoluten Adressen eingegeben werden. Der Assembler berechnet die relative Sprungweite selbst. Weiterhin werden in Bezug auf die IX und IY Register die Offsetadressierungen mit + oder - eingegeben. Maximal mit +7F oder -7F.

FUNKTION : Basic nur Cassette-Version
 CPM nur Disk-Version
 AUFRUF : b (Cass.-Vers.)
 c (Disk-Vers.)
 ZWECK : Rückkehr zu Schneider Basic
 Rückkehr zu CPM Sytem

```

-----
DIALOG DES MONITORS   EINGABEN   KOMMENTAR
>                   b           Aufruf Basic (Cass.)
                           c           Aufruf CPM (Disk)

>Basic               : 'CR'       mit 'CR' - Taste
                           bestätigt. (ENTER)
                           oder
                           : 'ESC'   zu Kommando Modus
                           des Monitors.
  
```

Rückkehr von Basic zu Monitor:

```

|GMON
  
```

Neustart MONII:

```

|GMON 'CR' (ENTER)
  
```

Beschreibung: (Cass.)

Rückkehr zu Basic. Es erfolgt ein Warmstart, wobei im Speicher befindliche Programme nicht zerstört werden. PC und SP werden so regeneriert, wie sie vor dem Monitoraufruf standen. Wurde die 'b' Taste versehentlich betätigt, kann mit 'ESC' zum Monitor Kommandomodus zurückgekehrt werden. Unter besonderen Bedingungen kann dieser Befehl zum Absturz des Systems führen, wenn Register oder der Bankstatus des Betriebssystems verändert wurden. Generell sollten daher die Register nicht verändert werden, da diese vom Schneider Betriebssystem für Interrupt's verwendet werden.

Soll von Basic aus der Monitor wieder aufgerufen werden (Warmstart), so ist mit dem Befehl |GMON eine Rückkehr möglich. D.H. es kann ein Programm in Basic geschrieben werden, und anschließend ist ein Untersuchen mit dem Monitor möglich.

Beschreibung: (Disk)

Warmstart des CP/M Systems. Programme welche sich zwischen der Adresse 100H und dem Beginn des Monitors befinden werden nicht zerstört. Vor dem eigentlichen Warmstart wird mit 'CPM' nochmals abgefragt ob der Monitor wirklich verlassen werden soll. Mit 'CR' (ENTER) wird dies quittiert, mit 'ESC' kann noch zum Monitor zurückgekehrt werden. Von CP/M aus wird der Monitor einfach mit 'GMON' aufgerufen.

FUNKTION : Disassembler
 AUFRUF : d
 ZWECK : Lesen von Speicherz. in Assembler Schreibweise

```

=====
DIALOG DES MONITORS   EINGABEN   KOMMENTAR
>                    d           Aufruf Disassembler

>Disassembler
Anfangsadresse      : 100         Adresse ab welcher
                               disas. werden soll.
Laenge              : 8           Langenangabe oder
                               optional
End Adresse         : (108)       die Endadresse.
  
```

Listing des Disassemblers:

```

100 31 00 30      LD SP,3000
103 3E FF        LD A,FF
105 CD 54 20     CALL 2054

>                : 'ESC'       ESC Taste bewirkt
                               Abbruch der Funktion.
  
```

Beschreibung:

Mit diesem Befehl besteht die Moglichkeit Programme zu analysieren, ohne sich um den Maschinencode kummern zu mussen. Der Disassembler ubersetzt ab der angegebenen Adresse den Maschinencode in die Assemblerschreibweise. Hierbei ist es moglich, die Lange des Programms oder wahlweise die Endadresse anzugeben. Ist das Listing langer als auf der Bildschirmseite dargestellt werden kann, so wird nach der ersten Seite eine Pause eingelegt und auf 'CR' gewartet. Abgebrochen wird durch Tastendruck auf 'ESC' oder wenn die Endadresse (Lange) erreicht wird. Mit einer beliebigen Taste, auer 'ESC', kann das laufende Listing angehalten, und wieder fortgesetzt werden. Auch kann eine Offset-Adresse mit berucksichtigt werden, siehe unter Offset-Funktion. In Zusammenhang mit dem U-Befehl (einschalten des ROM's) kann das gesamte Betriebssystem, sowie eventuell vorhandene Zusatzroms, des CPC 464 disassembliert werden.

FUNKTION : Editor
 AUFRUF : e
 ZWECK : Speichereingaben in ASC II (Texte)

```

=====
DIALOG DES MONITORS   EINGABEN   KOMMENTAR
>                    e           Aufruf Editor

>Editor (Stopp = 'ESC')
Anfangsadresse      : 100         Adresse ab welcher
                               der ASC II Code
                               abgelegt wird.

0100 Dieser Text wird eingegeben (CR)   Text eingeben, Zeile
                                         mit 'CR' abschlieen.

011D LF und CR werden mit abgelegt. (CR)
013D Mit Backspace kann korrigiert (CR)
015C werden.

                               'ESC'       'ESC' - Taste bewirkt
                                         Abbruch.

End Adresse : 0163                Anzeige der letzten
                                         eingegebenen Adresse.

>                                Kommandomodus
  
```

Beschreibung:

Hiermit konnen Texte im Speicher abgelegt werden, die z. B. fur eine Meldung Ihres Programmes notig sind. Ab der Anfangsadresse wird der Text abgelegt. Innerhalb einer Zeile kann mit der Taste Backspace eine Falscheingabe korrigiert werden. Mit der 'CR' Taste wird die Zeile abgeschlossen, wobei die Zeichen LF (0A Hex) und CR (0D Hex) mit im Speicher abgelegt werden. Mit jeder neuen Zeile wird die aktuelle Adresse angezeigt. Mit der Taste 'ESC' wird die Funktion abgebrochen, wobei die Adresse, bei welcher der letzte Buchstabe abgelegt wurde, angezeigt wird.

FUNKTION : Fill Memory

AUFRUF : f

ZWECK : Füllen des Speichers mit einer Konstanten

```

-----
DIALOG DES MONITORS   EINGABEN   KOMMENTAR
>                       f           Aufruf Funktion

>Fill Memory
Anfangsadresse       :   100

Laenge               :   60           Längenangabe oder
                                optional
End Adresse          :  (180)       Endadresse

Zeichen               :   ff         Zeichen mit welchem
                                der Speicher gefüllt
                                wird.

>                               Fertigmeldung des
                                Monitors.

```

Speicherlistings nach Ausführung:

```

0100 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF *.....
0110 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF *.....
0120 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF *.....

```

Beschreibung:

Dieser Befehl ermöglicht es, einen Speicherbereich mit einem konstanten Wert zu füllen. Als Zeichen kann ein Byte in Hexform angegeben werden. Für die Länge des Speicherbereiches kann die Endadresse oder die Länge angegeben werden.

FUNKTION : Goto

AUFRUF : g

ZWECK : Setzen Breakp. 1-3 und Programmstart

```

-----
DIALOG DES MONITORS   EINGABEN   KOMMENTAR
>                       g           Aufruf Goto Funktion

                                +-----+ alte Adresse
                                ! +-----+ eingegebene Adresse
                                ! !

>Go ab Adresse,L,N : 3000 100       L=listen Breakp.
                                N=löschen Breakp.
                                ( new )

Breakpoint 1 bei Adr: 3045 103      Breakpoint 1 setzen.

Breakpoint 2 bei Adr: 3078 1F7      Breakpoint 2 setzen
                                ( oder 'CR' ).

Breakpoint 3 bei Adr: 34F7 + 'CR'    Breakpoint 3 wird
                                geändert
                                Programm wird ge-
                                startet.
                                'CR'

```

Anzeigeformat bei einem Break:

```

BREAK bei 0103 DE FF 13 A,FF
A BC DE HL IX IY PC SP F-STATUS PC I AF' BC' DE' HL'
FF 00FF 09FF 1112 00FF 0000 0103 3000 02 1-----1- 0F 0000 FFFF 0F00 FFF7

```

Anmerkung: 1. Zeile Adresse und Befehl bei welchem der Break auftrat.
2. und 3. Zeile Auflistung der Register und Flags mit Inhalt.

Beschreibung:

Diese Funktion dient dazu, Programme zu starten und zu testen. Es können bis zu drei Breakpoint's gesetzt werden. Wird ein Breakpoint während der Programmausführung erreicht, meldet sich der Monitor mit der Anzeige sämtlicher Register. Wobei auch der Befehl, bei welchem dieser Break ausgelöst wurde, in Hex und Assemblerschreibweise ausgegeben wird. Mit Go und der Taste '(CR)' kann das Programm weiter fortgeführt werden, bis zum nächsten Auftreten eines Breakpoint's. Alle gesetzten Breakpoint's werden automatisch durch den Monitor wieder regeneriert. Hierzu ist es jedoch notwendig, daß mindestens zwei Breakp. gesetzt werden, da sonst vom Monitorprogramm kein Zugriff mehr zum Anwenderprogramm besteht. Von einem Anwenderprogramm aus muß mit einem Restart (RST 30) wieder zum Monitor zurückgekehrt werden. Auf keinen Fall darf über den Kaltstart-Einsprung zurückgekehrt werden. Bei Aufruf der Funktion werden die letzten verwendeten Adressen wieder aufgelistet. Sollen diese unverändert übernommen werden, so wird anstelle der neuen Adresseingabe nur mit 'CR' geantwortet.

FUNKTION : Hexadezimal rechnen.

AUFRUF : h

ZWECK : subtrahieren, addieren, multiplizieren,
dividieren von Hex Zahlen, sowie Umwandlung
von Hex zu Dez. und Dez. zu Hex.

DIALOG DES MON.	EINGABEN	AUSGABEN	KOMMENTAR
>	h		Aufruf Funktion
>Hex-Arithmetik	55+12 +'CR'	=0067	Hexadezimale Addition von zwei Byte.
2. Beispiel			
Hex-Arithmetik	FFFF-80 +'CR'	=FF7F	Subtraktion.
3. Beispiel			
Hex-Arithmetik	ad354*125 +'CR'	=F1DF24	Multipl.
4. Beispiel			
Hex-Arithmetik	fd9000/167 +'CR'	=B4D0	Division.
5. Beispiel			
Hex-Arithmetik	ffff=d +'CR'	=65335	Wandlung hex zu dez.
6. Beispiel			
Hex-Arithmetik	100=h +'CR'	=64	Wandlung dez. zu hex.

Beschreibung:

Mit dieser Funktion ist es möglich, Hexadezimal (Sedezimal), zu rechnen. Es ist das Addieren, Subtrahieren, Multiplizieren und Dividieren von bis zu 32 Bit Werten durchführbar. Es wird immer im hexadezimalen Zahlensystem gerechnet. Weiterhin können Hexadezimalzahlen in Dezimalzahlen und umgekehrt umgewandelt werden. Als Eingabe sind ein bis acht stellige Hex-Zahlen zulässig, wobei zu beachten ist, daß das Ergebnis nicht größer wie acht Hexadezimalzahlen sein kann.

FUNKTION : IN/OUT Port

AUFRUF : I

ZWECK : Einlesen oder Schreiben von/zu Port

DIALOG DES MONITORS	EINGABEN	KOMMENTAR
1. Abrufen:		
>	I	Aufruf der Funktion
>IN/OUT Port	0000	Port-Adresse in 16 Bit welche gelesen werden soll.
0000 90	'CR'	Mit 'CR' weiter-
0001 00	'CR'	schalten zum nächsten
0002 FF	'CR'	I/O Port.
	'ESC'+ 'CR'	Abbrechen.
Zu Port schreiben:		
>	I	Aufruf
>IN/OUT Port	0000	Port-Adresse.
0000 90	FF	Der Wert 90 wird vor- her gelesen, dann der Wert FF eingegeben.
Quittierung:	0000 FF	Nochmaliges autom. Lesen vom gleichen Port.

Beschreibung:

Diese Funktion dient zum Ausgeben eines Bytes an einen Ausgabeport, bzw. zum Einlesen eines Bytes von einem Eingabeport. Im Gegensatz zu vielen Z80 Systemen ist es beim Schneider PC notwendig, 16 Bit Adressen für Ein/Ausgabe - Befehle zu verwenden. Dies wird durch den Monitor voll unterstützt. Durch Betätigen der 'CR'-Taste wird die Portadresse automatisch hochgezählt und der jeweils gelesene Wert angezeigt. Hierdurch ist es möglich, mehrere Portadressen schnell abzufragen. Soll ein Byte zu der angesprochenen Output Adresse gesendet werden, muß dieses lediglich hinter den gelesenen Wert geschrieben werden. Anschließend wird automatisch nochmals von derselben Adresse gelesen, und dieser Wert gelistet.

FUNKTION : Speichertest

AUFRUF : J

ZWECK : Lokalisierung defekter Speicherstellen

```

-----
DIALOG DES MONITORS   EINGABEN   KOMMENTAR
>                               j       Aufruf der Funktion

>Ramtest
Anfangsadresse       : 100 + 'CR'   Adresse ab welcher
                                                getestet wird.
Laenge               : 80 + 'CR'   Längenangabe oder
                                                optional die
End Adresse          : (180)       Endadresse eingeben.

kein Ramfehler                               Der getestete Bereich
                                                ist fehlerfrei.

QREBI

Ramfehler bei       : 0100           Fehler bei den Adr.
Ramfehler bei       : 0101           100-104.
Ramfehler bei       : 0102
Ramfehler bei       : 0103
Ramfehler bei       : 0104

```

Beschreibung:

Diese Funktion dient zum routinemäßigen Überprüfen des Speicherbereiches (RAM), oder zum Lokalisieren defekter Speicherzellen. Der Test arbeitet zerstörungsfrei, d. h. es werden keine Programme, die in diesem Bereich liegen zerstört oder verändert. Es wird jeweils ein Testmuster in jede Speicherzelle geschrieben, geprüft und anschließend der ursprüngliche Inhalt der Zelle wieder zurückgeschrieben.

FUNKTION : Laden

AUFRUF : I

ZWECK : Laden von Cassette in den Speicher

```

-----
DIALOG DES MONITORS   EINGABEN   KOMMENTAR
>                               I       Aufruf der Funktion

>Laden von Cassette
Dateiname            : datei         Name des Files ist
                                                z. B. "DATEI"
Cass, Puffer 2k      : 6800 + 'CR'   Hinweis auf Lage des
                                                Cass.-Buffers. Mit
                                                'CR' quittieren.
                                                wie in "BASIC"

Press PLAY then any key: + 'CR'

Anfangsadresse       : 0170 + 'CR'   oder andere Adreß-
                                                eingabe möglich.
Endadresse           : 0480         Hinweis auf die
                                                Endadresse des
                                                Programms "DATEI".

>                               'ESC'   ESC Taste bewirkt
                                                Abbruch der Funktion.

```

Beschreibung:

Hiermit ist es möglich Programme, welche mit dem 'SAVE' Kommando auf Cassette abgespeichert wurden, wieder in den Speicher zu laden. Mit Eingabe des Zeichens " an Stelle des Dateinamens wird das erste auffindbare Programm von der Cassette geladen. Jede Dateiart ob Basic, Binär usw. ist ladbar. Es wird automatisch erkannt von welcher Adresse das Programm einmal auf Band geladen wurde. Diese Adresse kann akzeptiert oder wahlweise auch geändert werden. Hierdurch ist es möglich verschiedene Programme zusammenzufügen. Zum Laden des Programms wird ein 2k Byte großer Cassetten-Buffer benötigt welcher nicht innerhalb des Monitors liegt. Die Voreinstellung liegt auf Adresse 6800 und kann wahlweise geändert werden, wenn an dieser Stelle bereits ein Programm steht.

FUNKTION : Memory (Memory aendern)

AUFRUF : m + 'CR'

ZWECK : Speicherinhalt modifizieren und eingeben.

DIALOG DES MONITORS	EINGABEN	KOMMENTAR
>	m	Aufruf der Funktion
>Memory Anfangsadresse :	'CR'	keine Adresse sondern 'CR' eingeben.
Memory aendern :	100 + 'CR'	Adresse eingeben, ab welcher die Eingabe erfolgen soll.
0100 3E 55 0100 55	'CR'	Hier wird das Byte 3EH gegen 55H ausgetauscht, und anschließend quittiert.
0101 FF	'CR'	Diese Speicherzellen
0102 00	'CR'	werden nicht ver-
0103 3E	'CR'	ändert, sondern mit 'CR' nur gelistet.
>	'ESC'	ESC Taste bewirkt Abbruch der Funktion.

Beschreibung:

Dieses Kommando dient zum Listen und Modifizieren von Speicherstellen. Aufgerufen wird das Kommando ähnlich wie die Memory-Funktion (siehe letzte Seite). Jedoch wird die Abfrage: 'Anfangsadresse' mit 'CR' beantwortet. Dann erfolgt eine Meldung: 'Change Adresse', was bedeutet: wechsele Inhalt der Adresse. Jetzt ist die Adresse einzugeben, ab welcher der Speicherinhalt modifiziert werden soll. Es wird die Adresse und der Inhalt gelistet, und auf eine Eingabe gewartet. Wird mit 'CR' geantwortet, erfolgt eine Incrementierung der Adresse ohne den Inhalt zu verändern. Wird ein Byte eingegeben, wird dieses in die Speicherzelle geschrieben und sofort nochmals, einschließlich Adresse, angezeigt. Hierdurch ist gewährleistet, daß die Eingabe auch tatsächlich zustande kam. Anschließend wird die Adresse wieder incrementiert und der nächste Inhalt angezeigt.

FUNKTION : Nächster (Befehl), TRACE-Funktion

AUFRUF : n

ZWECK : Austesten von Programmen.

DIALOG DES MONITORS	EINGABEN	KOMMENTAR
>	n	Aufruf der Funktion
>Next ab Adresse : CBSF	100 + 'CR'	Es wird der voreinge- gestellte Programm- Counter angezeigt, (CBSF). Wird ein anderer Progr. Start gewünscht, diesen eingeben (100), sonst 'CR'.
Anzahl Schritte :	1	Einen Befehl abarbeiten.

Anzeige nach Abarbeitung des Befehls

```
BREAK BEI 0103 21 55 02 LD HL,0255
A BC DE HL IX JY PC SP F SZHWPNC I AF' DC' BE' HL'
C7 0000 0300 0255 A900 8236 0114 9FFF 0 1----- 2F 0000 0000 FFFF 0000
```

> Monitor wieder im Kommandomodus.

Beschreibung:

Durch diesen Befehl wird das Austesten von Programmen in Einzelschritten ermöglicht. Es werden nach Abarbeitung des Befehls alle Registerinhalte, sowie die letzte Adresse und der letzte Befehl, im Maschinencode und Assemblerschreibweise angezeigt. Es ist darauf zu achten, daß beim ersten Aufruf der Funktion, der PC-Counter geladen wird. Alle folgenden NEXT Aufrufe müssen nur mit 'CR' beantwortet werden, da der PC-Counter autom. richtig gesetzt wird. Diese Funktion funktioniert nur im RAM, da intern mit Breakpoints gearbeitet wird!. Aus diesem Grunde werden 'CALL' Aufrufe in dem zu testenden Programm auch übersprungen, wenn sie über die Firmware Sprungtabelle ins Betriebssystem ROM oder in den Monitor selbst führen. Ein Programmabsturz wäre unvermeidlich. Natürlich wird das Unterprogramm trotzdem durchlaufen, allerdings erfolgt eine Anzeige erst nach der kompletten Ausführung. Die Ausführung des Unterprogramms erfolgt hierbei in 'Echtzeit'. Call Aufrufe in Anwenderprogramme im RAM werden aber in Einzelschritten abgearbeitet und angezeigt. Restarts (RST nn) werden nicht ausgeführt, bzw. übergangen. Diese Testmöglichkeit von Programmen wird in 'Neudeutsch' auch als TRACE - Funktion bezeichnet.

FUNKTION : Offset
 AUFBRUF : o
 ZWECK : Verrechnen einer Offset-Adresse.

DIALOG DES MONITORS	EINGABEN	KOMMENTAR
>	o	Aufruf der Funktion
>Offset = 0000		Bisherige Offset-Adresse.
Neuer Offset =	100	100 als Offset Adr.
	Offset = 0100	Quittierung.

Beispiel für Memoryfunktion mit Offset:

>Memory		
Anfangsadresse	: o,100 + 'CR'	Soll die gesetzte Offset Adr. mit ausgewertet werden, ist ein o, vor die Adresse zu setzen.

Beschreibung:

Der eingegebene Offset wird bei Adreßeingaben mit der tatsächlichen Adresse verrechnet. Dies ist zum Beispiel sinnvoll, wenn ein relatives Assemblerlisting vorliegt. Hierdurch ist es möglich, das Programm in einem beliebigen Adressraum abzuliegen, es aber mit der gewünschten Adresse einzugeben. Soll die Offset-Adresse zum Beispiel bei der Memoryfunktion berücksichtigt werden, muß vor die Adresse ein 'o' gesetzt werden. Prinzipiell ist diese Funktion bei jeder Adreßeingabe möglich.

FUNKTION : Save
 AUFBRUF : s
 ZWECK : Abspeichern eines Programms auf Cassette.

DIALOG DES MONITORS	EINGABEN	KOMMENTAR
>	s	Aufruf der Funktion
>Save auf Cassette Filename	: date1	Name des Files ist z. B. "DATE1"
Cass, Puffer 2k	: 6800 + 'CR'	Hinweis auf Lage des Cass.-Buffers. Mit 'CR' quittieren.
Anfangsadresse	: 170 + 'CR'	Anfang des Programms "DATE1".
Laenge	: 200 + 'CR'	Länge des Programms
Endadresse	: (370 + 'CR')	oder optional die Endadresse des Programms "DATE1". (Länge mit 'CR' quittieren.
Press REC and PLAY then any key	: 'CR'	Wie in Basic mit 'CR' beantworten. Funktion ab hier wie in "BASIC".
>	'ESC'	ESC Taste bewirkt Abbruch der Funktion.

Beschreibung:

Hiermit können Programme von einer beliebigen Adresse aus, auf Cassette abgespeichert werden. Es muß der Dateinamen unter welchem das Programm abgespeichert werden soll, sowie die Anfangs-Adresse, Länge, oder als Option die End-Adresse, angegeben werden. Die Programme werden im Binär-Format abgespeichert (siehe Basic-Handbuch). Wie bei der LOAD-Funktion wird auch hier ein Cassettenbuffer benötigt. Dieser ist auf die Adresse 6800H voreingestellt, kann aber abgeändert werden.

FUNKTION : Trans
 AUFRUF : t
 ZWECK : Kopieren, verschieben von Speicherbereichen.

DIALOG DES MONITORS	EINGAREN	KOMMENTAR
>	t	Aufruf der Funktion
>Transfer Zieladresse	: 100	Adresse zu welcher der Transfer erfolgen soll eingeben.
Sendeadresse	: 4400	Hier beginnt der zu kopierende Bereich.
Laenge	: 400	Längenangabe oder optional die
Endadresse	: (4800)	Endadresse.
>		Fertigmeldung.

Beschreibung:

Mit diesem Kommando können Speicherinhalte beliebig kopiert, bzw. verschoben werden. Es muß die Ziel- und die Quelladresse angegeben werden. Wird die Längenangabe mit einem 'CR' quittiert, kann auch die Endadresse angegeben werden.

FUNKTION : Umschaltung
 AUFRUF : u
 ZWECK : Umschaltung der Speicherbank

DIALOG DES MONITORS	EINGAREN	KOMMENTAR
>	u	Aufruf der Funktion
>Bank = RAM Neue Bank, RAM=A, ROM=0, akt. Bank, od. Nr.		A = RAM einschalten B = aktuelle Bank 0 = ROM CPC einsch. 1 = Extens ROM 1 2 = Extens ROM 2 usw. 0 + 'CR' Eingabe.

Quittierung des Monitors:

ROM = 0 ROM 0 ist eingeschaltet

Beschreibung:

Mit dieser Funktion kann zwischen den Speicher - Bank's umgeschaltet werden. Alle Listingfunktionen des Monitors erfolgen von dieser aktiven 'Bank'. 'A' steht für den Ram Bereich des CPC 464 von Adr. 0 - FFFF (Hex). Das CPC Betriebssystem, Adr. 0 - 4000, und der Basic-Interpreter Adr. C000 - FFFF für '0'. Hier ist zu beachten, daß sich das Ram von Adr. 4000 - C000 mit überlagert. Extensions - ROM's werden ebenfalls unterstützt. Diese liegen immer im Adressbereich C000-FFFF (Hex), und werden immer vom RAM des CPC von Adr. 0 - C000 überlagert. Hierdurch ist es möglich auch das Betriebssystem ROM oder das Basic ROM zu listen und disassemblieren. Es ist nur ein Lesen des Bank's möglich.

Die aktuelle Bank ist die Bank die vom Betriebssystem des CPC 464 gerade aktiviert ist. Generell ist zu sagen, daß mit der U-Funktion keine Bankstatusänderung des Betriebssystems des CPC 464 erfolgt, sondern nur mit Listingfunktionen des Monitors auf diese Bank's zugegriffen wird.

Bei folgenden Befehlen ist die Bankumschaltung wirksam:

Memory, Suchen, Verivy, Disassembler, Transfer.

FUNKTION : Vergleich

AUFRUF : v

ZWECK : Vergleichen von Speicherbereichen

DIALOG DES MONITORS	EINGABEN	KOMMENTAR
>	v	Aufruf der Funktion
>Verify Memory		
Anfang Block 1	: 100 + 'CR'	Blöcke, die verglichen werden sollen.
Anfang Block 2	: 1100 + 'CR'	
Testlänge	: 400 + 'CR'	

Fehlermeldungen auf Differenz:

0100=C3	1100=00	Differenz zwischen den Blöcken.
0101=88	1101=00	
0102=56	1102=00	

> Fertigmeldung.

Beschreibung:

Es können Speicherbereiche verglichen werden. Erfolgt keine Fehlermeldung sind diese Speicherbereiche mit ihrem Inhalt identisch. Bei Differenzen werden die Adressen beider Blöcke, sowie die differierenden beiden Byte's angegeben.

FUNKTION : Suchen

AUFRUF : y

ZWECK : Suchen nach Zeichenfolge in Hex oder ASC II.

DIALOG DES MONITORS	EINGABEN	KOMMENTAR
>	y	Aufruf der Funktion
>Suchen		
Anfangsadresse	: 100 + 'CR'	Adresse, ab welcher gesucht werden soll.
Laenge	: 8000 + 'CR'	Längenangabe oder optional die
Endadresse	: (8100)	Endadresse, bis zu welcher gesucht werden soll.
ASC II oder Hex	: a	a für ASC II
		h für Hex
ASC II	: TEXT	Zeichenfolge, welche gesucht werden soll.
Zeichen gefunden bei Adresse 6788		Meldung, wo Zeichenfolge beginnt.
<u>Beispiel für Suchen in Hex:</u>		
Hex	: 3E	Byte, welches gesucht werden soll.
Zeichen gefunden bei Adresse 0305		Hier steht das Byte 3EH.

Beschreibung:

Diese Funktion dient zum Suchen von Zeichenfolgen im Speicher. Es kann nach ASC II Zeichenfolgen bis zu max. 8 Zeichen, und nach Byte folgen (Hex) von max. 2 Byte gesucht werden. Ausgegeben wird, an welcher Adresse die Zeichenfolge beginnt.

FUNKTION : Register
 AUFRUF : r
 ZWECK : Register anzeigen und ändern.

DIALOG DES MONITORS EINGABEN KOMMENTAR

Register listen

> r Aufruf der Funktion
 >Register : 'CR' Mit 'CR' beantworten

A BC DE HL IX IY PC SP F *SZHAPC I M' N' O' P' HL'
 C9 0000 BCFF 267F BFBF FF0F F320 50F4 60 -11----- BF FF7F FF0F FF0F FF0F

Register ändern

> r Aufruf der Funktion
 >Register : a + 'CR' ab A-Register ändern

A C9 00 A 00 'CR' Der Wert 00 wird anstelle des Wertes C9 eingeschrieben (Accu).
 B 00 FF B FF 'CR' B-Register FFH gegen 00 tauschen.
 C 00 ! 'CR' C-Register nur listen.
 D BC ! 'CR'
 ! !
 ! +----- Neuen Wert eingeben.
 !----- alter Wert wird gelistet.

Beschreibung:

Mit dieser Funktion können die Inhalte aller Register angezeigt werden. Natürlich ist es nicht möglich in einem laufenden Computersystem Register anzuzeigen, mit einem Trick, wie in unserem Monitor verwendet, ist dies jedoch möglich. Es werden nach einem "Break" alle Register gerettet (STACK). Diese Werte werden dann in Speicherzellen abgelegt, welche mit dieser Funktion dann gelesen werden können. Mit der Goto-Funktion werden diese Werte wieder über den Stack in die Prozessorregister gebracht, so daß Sie Ihre Programme mit definiert gesetzten Registern starten können. Es können nur Einzelregister gesetzt werden, also "H" und "L" einzeln, und nicht "HL". Der Zweitregistersatz des Z80 sollte beim CPC 464 nicht verändert werden, da er vom Betriebssystem für Interrupt's verwendet wird.

FUNKTION : Parameter
 AUFRUF : p
 ZWECK : Parameter ändern

DIALOG DES MONITORS EINGABEN KOMMENTAR

> p Aufruf der Funktion
 >Parameter Speedwrite : 00 für 1000 Baud
 01 für 2000 Baud
 Cassette=0, Disk =1 : 00 Cassette aktiv
 01 Diskette aktiv
 Dieses Kommando gilt nur für die CP/M Version.
 Restart bei Break : 30 Restartadresse welche für Breakpoints benutzt wird. Dieser kann geändert werden.

ANMERKUNG:

Die Restartadresse kann auf einen beliebigen RST-Punkt gelegt werden. Dies ist jedoch mit Vorsicht zu benutzen, da ein Systemabsturz die Folge sein kann. Die Möglichkeit einer Änderung wurde nur vorgesehen, falls zukünftig der RST 30 vom CPC Betriebssystem belegt werden sollte.

Das Hin und Herschalten zwischen Cassette und Disk ist nur bei der CP/M Version möglich.

FUNKTION : Drucker
 AUFRUF : CTRL + P
 ZWECK : Parallelschalten des Druckers zur Ausgabe.

DIALOG DES MONITORS	EINGABEN	KOMMENTAR
>	CTRL+P	Aufruf der Funktion

Beschreibung:

Alle Ausgaben auf dem Bildschirm können auch auf dem Drucker mitgeschrieben werden. Hierzu muß die CTRL Taste gleichzeitig mit der "P" Taste gedrückt werden. Das Abschalten wird in gleicher Weise durchgeführt.

FUNKTION : Help
 AUFRUF : ?
 ZWECK : Auflistung aller Befehle (Hilfsmenü)

DIALOG DES MONITORS	EINGABEN	KOMMENTAR
>	?	Aufruf der Funktion

>Help

Es werden alle Kommandos in Form eines Hilfsmenü gelistet.

KURZBESCHREIBUNG

Mit Hilfe des Disk-Monitors besteht ein direkter Zugriff zu den Sektoren einer Diskette. Hierdurch ist es zum Beispiel auch möglich, versehentlich gelöschte Programme wieder zu "entlöschten". Ausgelegt ist dieses Programm fuer den Betrieb unter CP/M 2.2, wobei sich das Programm jeweils dem verwendeten Disk-Format selbststaendig anpaßt. Es werden keine CP/M spezifischen Programmteile benutzt, sondern nur CP/M Software-Schnittstellen. Hierdurch ist eine Verwendung auf jedem CP/M 2.2. Rechner moeglich. Die tatsaechliche Sectorgroesse ist nicht relevant, da nur logische CP/M Sektoren, a. 128 Byte, angesprochen werden. CP/M selber kennt ja keine anderen Sectorgroessen.

Das Help Menu:

Das Hilfsmenue erscheint nach dem Programmstart, und kann jederzeit durch Eingabe eines "H" wieder aufgerufen werden.

DISK Editor Kommandos:

=====

W = WRITE-Sector R = READ-Sector I = INIT-Disk D = DUMP Sector
 S = set SECTOR T = set TRACK P = set PARAM. Q = QUIT (ENDE)
 + = naechst. SEC - = letzter SEC. H = HELP L = LAUFWERK
 M = MEMORY E = EDITOR F = FIND

LAUFW : A TR = 00 SEC = 01 PS = 01

<DE>

Unter dem Menue finden wir die Angabe, welches Laufwerk angewaehlt wurde, welche Spur eingestellt ist (TR=Track) und welcher logische Sector (SEC) voreingestellt ist. PS ist nur ein Hinweis auf den entsprechenden "physikalischen" Sector.

Grundsaeztlich wird ein Sector in einen Puffer im RAM eingelesen. Wenn wir mit der Dump Funktion einen Sector ansehen, sehen wir nur das Abbild des Sectors. Alle eventuellen Aenderungen mit der Memory oder Edit-Funktion veraendern nur dieses "Abbild". Erst mit der Write-Funktion wird der Sector tatsaechlich zurueckgeschrieben. Mit der Read-Funktion kann nach einer Aenderung der urspruengliche Stand wieder hergestellt werden, solange die Write-unktion nicht benutzt wurde. Daher Vorsicht mit WRITE !.

Bei den Funktionen +, -, und FIND (Suchen) wird der Sectorpuffer automatisch nachgeladen, ein READ Aufruf ist also nicht erforderlich.

DM - Disk Monitor

Die DUMP Funktion

Dient zum Listen eines Sectors in Hex und ASCII.

<DE> DUMP

Inhalt des Sector Puffers :

Adr.	0	4	8	C	
0000	80C300DA	18424F4F	544C4144	45205345	*.....BOOTLADE SE*
0010	43544F52	2E204552	53544520	41445220	*CTOR. ERSTE ADR *
0020	49535420	5A49454C	4144522E	00000000	*IST ZIELADR.....*
0030	00000000	00000000	00000000	00000000	*.....*
0040	00000000	00000000	00000000	00000000	*.....*
0050	00000000	00000000	00000000	00000000	*.....*
0060	00000000	00000000	00000000	00000000	*.....*
0070	00000000	00000000	00000000	00000000	*.....*

LAUFW : A TR = 00 SEC = 01 FS = 01

<DE>

Die Memory Funktion.

wird zum Aendern oder Eingeben in Hex Schreibweise verwendet. Soll eine Adresse nicht geseendert, also uebersprungen werden, so wird nur "ENTER" gedrueckt.

<D> M MEMORY

<DE>

Beginn Adr. : 00

0000	80			
0001	C3			
0002	00	ff	0002	FF
0003	DA	ff	0003	FF
0004	18	ff	0004	FF
0005	42			

<DE>

Die Edit Funktion.

wird zum Eingeben von Texten verwendet, es wird jeweils am Zeilenanfang angezeigt, bei welcher Adresse man sich befindet.

<DE> EDITOR (Texteingabe)

Beginn Adr. : 0

0000 abcdefgh ABCDEFGH

0011 <CR>

0011

letzte Adr. : 0011

DM - Disk Monitor

Die LAUFWERK

Funktion, wird verwendet um das angewaehlte Laufwerk zuwech-seln, sofern vorhanden.

Die INIT

Funktion muss immer nach dem Wechseln einer Disk verwendet werden. Es wird dann ein sogenannter Login durchgefuehrt, und auf ein eventuell anderes Disk Format umgeschaltet.

Mit S und T

werden die Sektoren und Spuren (Track) angewaehlt, mit - und + wird jeweils die aktuelle Sektornummer erhoeht oder erniedrigt. Anstelle von - und + koennen auch die Pfeiltasten benutzt werden.

Mit Q wird das Programm verlassen.

Die FIND.

Funktion durchsucht die ganze Diskette nach einem ASCII Text (max. 8 Zeichen), oder nach einem Hex Wert (2 Zeichen).

Mit PARAMETER

koennen, wie der Name schon sagt, Parameter geseendert werden, zum Beispiel die Anzahl der Spuren einer Diskette. Dieser Wert sollte beim CPC 464 = 40 Track nicht ueberschreiten.

DM - Disk Monitor

Hinweise zum "Entloeschen"

von DATEIEN. Wurde eine Datei versehentlich geloescht, so besteht die Moeglichkeit, diese Datei wieder zum Leben zu erwecken, wann kein weiterer Schreibzugriff auf diese Diskette erfolgt ist. Dies ist moeglich, da das CP/M Betriebssystem die Datei nicht tatsaechlich loescht, sondern nur im Inhaltsverzeichnis eine Marke setzt, welche diese Datei als geloescht kennzeichnet.

Beispieli:

Inhalt des Sector Puffers :

```
Adr.      0          4          8          C
0000  1.-> 00444D20 20202020 20434F4D 00000016 *.DM      COM....*
0010      03000400 00000000 00000000 00000000 *.....*
0020  2.-> E5474D4F 4E202020 20434F4D 0000000C *.GMON    COM....*
0030      05000000 00000000 00000000 00000000 *.....*
0040  3.-> E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 *.....*
0050      E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 *.....*
0060      E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 *.....*
0070      E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 *.....*
```

LAUFW : A TR = 03 SEC = 01 PS = 01

Im Inhaltsverzeichnis von CP/M werden jeweils 32 Bit fuer einen Eintrag verwendet. Das erste Byte bezeichnet die User Nr., in unserem Falle 00 (Markierung 1.). Steht an dieser Stelle E5 so gilt dieser Inhaltsverzeichniseintrag als nicht belegt, bzw. geloescht. An der Markierung 2. sehen wir einen geloeschten Eintrag. Wird an dieser Stelle 00 eingetragen, so ist diese Datei (im Beispiel GMON.COM) wieder im Inhaltsverzeichnis der Disk enthalten. Voraussetzung ist natuerlich, dass nach dem versehentlichen Loeschen kein Schreiben zur Disk erfolgt ist. Es koennte sonst sein, dass die als frei markierten Bloecke, welche z.B. beim Eintrag GMON.COM auf Adresse 30-3F stehen, bereits anderweitig belegt wurden.

Ein 32 Byte Verzeichniseintrag gliedert sich folgendermassen:

```
Byte: 0          Benutzernummer
      1 - 8       Dateiname z. B. GMON
      9 - 11      Datei Art z. B. .COM
     12          Eintragsnummer fuer diesen Dateinamen
     13 + 14     reserviert fuer Interne Verwendung
     15          Anzahl der Aufzeichnungen in diesem Eintrag
     16 - 31     Nummern der fuer diese Datei belegten
                Bloecke.
```

Benoetigt eine Datei mehr Bloecke als in einen Dateieintrag passen, so legt CP/M unter dem gleichen Namen weitere Eintraege ab, jedoch mit jeweils steigenden Eintragsnummern. Der erste Eintrag hat immer die Nr. 0. Dies ist wichtig beim 'Entloeschten' einer Datei.