

### Willy's Monitor

Die Software ist in ein EPROM eingebrannt, sodaß sie unmittelbar nach Einschalten des Rechners verfügbar ist. Der Monitor ist, bis auf die fehlenden QD-Routinen, kompatibel zu dem normal eingebauten Monitor 9Z-504M.

#### Eigenschaften

Die QD-Routinen wurden aus dem EPROM entfernt, um Platz für Floppy-Routinen zu schaffen. Mit Willy's ist es möglich, alle wichtigen Funktionen unmittelbar, ohne Zuladen einer Utility, auszuführen:

- \* Laden, Starten, Kopieren mit ein oder bis zu vier Floppies von einzelnen Dateien, auch des Masters, oder ganzen Disketten (nur SHARP/BASIC-Format, nicht CP/M)
- \* Formatieren
- \* Mastern = Festlegen eines OBJ-Programms auf der Diskette, das beim Einschalten des Rechners gestartet werden soll.
- \* CMT-Routinen wahlweise mit halber Lade- und Speicherzeit und halbem Bandbedarf (normal und Rapid)
- \* Alle Speicherbereiche sind durch einen Programmiertrick vom Monitor aus zugänglich (CG-ROM, Video-RAM, und Programm-RAM, auch "hinter" den Monitor-Bereichen 0000.. 1000h und E000.. FFFFh)
- \* Eine Reihe weiterer Funktionen, wie Vergleichen, Suchen, Text eingeben, RAM-Card schreiben und lesen, Drucker zuschalten, ASCII- oder SHARP-Zeichencode...
- \* Supermoni (Software) erhältlich, mit der ein Daten-Transfer zwischen CP/M- und SHARP-Disketten möglich ist

#### Installation

Vor Beginn Netzkabel und alle anderen Verbindungen entfernen! Allgemeine Hinweise sind im Handbuch Seite 7-18 bis 20 zu finden. Das EPROM steckt in einem 28-poligem Sockel vor dem 44-poligen Anschluß für den Steckkartenträger. Es ist durch vorsichtiges Hebeln mit einem Schraubenzieher leicht zu entfernen. Ein Verkanten sollte möglichst vermieden werden, damit sich die Beinchen nicht verbiegen. Beim Einsetzen des neuen EPROMS biegen Sie, falls erforderlich, die Beinchen etwas zurecht, so daß diese gut in die Federkelche des Sockels hineinpassen und beim Hineindrücken nicht umknicken. Die Gehäusekerbe des EPROMS muß über der Kerbe des Sockels liegen, also in normaler Lage des Rechners nach links zeigen.

#### Programm-Aufbau

Willy's Monitor wurde so in den bekannten Monitor eingefügt, daß eine möglichst gute Kompatibilität erhalten blieb. Die QD-Befehle wurden zugunsten der Floppy-Kommandos entfernt. Die Bootroutinen wurden an zwei Stellen geändert, wobei die Aufruf-Adressen blieben: Es werden jetzt auch Programme richtig gestartet, die zwischen 0000 und 1200H beginnen, und dem BASIC wird über die Register B und C mitgeteilt, von welcher Disk es geladen wurde, so daß das "AUTO RUN" auf dem richtigen Laufwerk gesucht wird.

Der neue Monitor interpretiert bereits die erste Eingabe als einen Befehl. Tippt man "L", so wird nachgefragt, ob "CMT starten, kopieren, laden" gemeint ist. Mit "<CR>, K, L" kann man sich für Laden und Starten, Kopieren oder nur Laden entscheiden. Bei letzterem folgt "ab" mit blinkendem Cursor. Sobald die gewünschte Ladeadresse eingegeben ist, kann das Laden beginnen.

Für die Kassetten-Kommandos ist noch ein besonderer Leckerbissen eingebaut: Nutzt man bei "L" und "S" den Prefix "R", so wird RAPID 3geladen, nämlich mit doppelter Geschwindigkeit. Beim Booten erscheint außer den bekannten Texten jetzt auch "R:Rapid CMT-load". Für die RAPID-Aufzeichnung ist Voraussetzung, daß geeignetes Band-

Material (Daten-Cassetten) verwendet wird, und nicht etwa alte ausrangierte Phonocassetten. Weiter muß darauf geachtet werden, daß mit der Aufzeichnung nicht bereits auf dem Vorlaufband begonnen wird. Alle Kommandos wurden mit so ausführlichen Kommentaren versehen, daß man sich nach einer kurzen Eingewöhnung ohne Gebrauchsanleitung zurechtfinden kann. Bei Befehlen, die eine Zerstörung der Datei auf der Disk nach sich zieht, wird mit einem "OK? J/N" nachgefragt, ob die Eingaben wirklich stimmen. Umfangreichere Ausgaben am Bildschirm lassen sich mit der Leertaste stoppen und wieder starten. Wenn ein Fehler auftritt, wird dieser gemeldet und der laufende Vorgang abgebrochen.

Alle Befehle sind als Unterprogramme aufgebaut, so daß sie sich auch in anderen Programmen wiederverwenden lassen. Die Floppy-Routine zum Schreiben und Lesen ist im alten 700-er Teil des ROMs (Monitor-Hauptprogramm) ab Adresse 0070H untergebracht. Dadurch ist es möglich, die Bootspur und die Directory in das RAM ab D000 zu schreiben. Um das RAM über den gesamten Adreß-Bereich erreichbar zu machen, werden vom Monitor Umlade-Routinen nach 1110H kopiert, die die Daten manipulieren und an den Monitor zurückgeben können. Durch Niederhalten der M-Taste beim Einschalten des MZ-800 erscheint unmittelbar der Monitor.

#### Speicher-Bedarf

1080 bis 11FF (für Stack u.a.Daten), CEE7...E7FF für Floppy-Routinen. Bei FORMAT 1200 bis 2FFF für Formatierdaten.

#### Normal-Modus

MZ-700, ROM 0...0FFF und E010...FFFF, Display D000...DFFF, SHARP-Code, Drucker aus

#### Prefix A

ASCII-Code: Alle Text-Daten werden als ASCII-Daten interpretiert. Beispiel: Mit A C können ASCII-Texte aus dem Speicher gelesen werden.

#### Prefix P

Printer-Protokoll (Centronics-Drucker) parallel zur Schirm-Anzeige. Störende Codes werden ausgeblendet.

#### Prefix R

RAM von 0000...FFFF. Mit diesem Prefix ist das gesamte RAM erreichbar, auch das " hinter " dem Monitor von E000 bis FFFF. Bei den CMT-Befehlen L und S :RAPID = doppelte Geschwindigkeit. Die Cassetten-Rekorder-Befehle überdecken nur den Bereich 1200...CFFF, die Floppy-befehle immer den Bereich 1200...FFFF.

#### A+R+P

sind untereinander und mit C...Z kombinierbar. Nach Ende des Kommandos wird der Normal-Modus wieder hergestellt.

#### 1,2,3 oder 4

Einleitung der Floppy-Disk-Befehle für die gewählte Disk ab RAM 1200 bis CEE5. In den Bereich CEE6 bis E7FF werden die Daten der zu bearbeitenden Datei, die Allocation-Tabelle, die Boot-Spur und die Directory untergebracht. Achtung: Bei der Bearbeitung von BSD/BRD-Dateien können Fehler entstehen, weil diese nicht wie OBJ- oder BTX-Dateien (= Typ 1 oder 2) in einem geschlossenen Block geschrieben sein müssen.

#### 1...4,C

COPIEREN 320 kB/ 80 Spuren in Blöcken zu 40 kB, auch mit einem Laufwerk möglich, BREAK möglich. Es wird angegeben, ab welche Spur gelesen bzw. geschrieben wird, und ggfs. zum Wechseln der Disk aufgefordert. Da die Original-SHARP-Disketten nur 70 Spuren nutzen, tritt beim Lesen des 8. Blockes ein Fehler auf, der vernachlässigt werden kann.

1...4,D

DIRECTORY mit Angabe des Masters, Dateityp, Start-Adresse, Länge, Ladeadresse, Startspur und -Sektor. Stop der Ausgabe mit der Leertaste.

1...4,E : ERASE=löschen einer oder mehrerer Dateien

1...4,F

FORMATieren einer Disk, 80 Spuren, BREAK möglich. 314 KB frei (ab Spur 1, Sektor 8). Bei Lesefehlern neu starten. Bei mehrfachem Fehler in einer Spur Disk verwerfen.

1...4,I: INITIALISIEREN einer Disk = Directory löschen

1...4,K: KOPIEREN einer Datei, auch Master, auf beliebige Disk

1...4,L: LADEN einer Datei ab anzugebende Adresse

1...4,M

MASTERN einer OBJ-Datei ( =Typ 1).Die deklarierte Datei wird in den Bootsektor übertragen und der Name auf 12 Zeichen gekürzt. Der Datei-Eintrag in der Directory wird durch die ungültige Typ-Nr. 80H unlesbar und unlösbar gemacht.

1...4,R

RUN = Starten einer OBJ-Datei oder des Masters. Nacheinander werden alle OBJ-Dateien zum Start angeboten.

1...4,S

SAVE = Abspeichern einer OBJ-Datei aus dem RAM ab 1200H oder höher. Es ist z.B. möglich, BASIC ab 1200H zu laden, zu verändern und wieder mit den Original-Adressen abzusaven.

1...4,T: TRACK = ab Spur/Sektor an anzugebende Adresse lesen

1...4,W: WRITE = ab Spur/Sektor beschreiben mit Daten ab Adresse

B Bell: Piepton bei jedem Tastendruck ein- und ausschalten

C Character-Ausgabe des gewählten Speicherbereiches

D Dumpen eines Speicherbereiches; Stop mit Leertaste

E Exit ins BASIC

F Booten, wie nach Reset; aber erst nach Ihrem OK!

G GOSUB/GOTO, wie der Befehl J

H Hören der CMT-Cassette

I 'IN',einlesen ins RAM aus der RAMCARD (ohne Directory)

J Sprung = Jump/Call,z.B. BASIC-Start => R J nach 0000

L LOAD von CMT mit Anzeige des Headers, mit Prefix R RAPID-Laden. Es kann geladen und gestartet werden oder kopiert oder nur geladen an wahlfreie Adresse. Nach dem Kopieren wird abgefragt, ob ein Verifizieren (Testlesen) gewünscht ist. Mehrfach-Kopien der eingelesenen Datei sind möglich, Ende mit SHIFT+BREAK.

M Memory-Schreiben im HEX-Code. Weiter oder zurück mit Pfeil-Tasten rechts und links, Ende mit SHIFT+BREAK.

O 'OUT',aus dem RAM in die RAMCARD (ohne Directory)

Q Randfarbe ändern; Inkrement mit jedem Tastendruck

S Speichern auf Cassette mit anschließendem Testlesen, wenn gewünscht. Siehe auch " R " und " L ".

T Transfer von Daten im Speicher, auch überlappender Bereiche

U Unterschied zweier Speicher-Bereiche. Abweichende Bytes werden ausgegeben.

V Verifizieren von Byte-Folgen (4 Bytes) mit Jokern = < CR >

W Text in den Speicher schreiben. Mit einem < CR > wird das Schreiben abgeschlossen.

X Kopieren eines Speicher-Bereiches (von..bis nach). Bei diesem Befehl ist das Zeichen-ROM von 1000...1FFF und Character-RAM von C000...CFFF eingeschaltet. Quell- und Ziel-Bereich dürfen sich nicht überlappen. "X von 1800 bis 18FF nach C000" schaltet den zweiten Zeichensatz mit doppelten Konturen für die Großbuchstaben ein.

Y Summe und Differenz von zwei 4-stelligen Hex-Zahlen

Z Füllen des Speicher-Bereiches mit einem Byte nach Wahl

## Programm-Beschreibung

Die einzelnen Befehle sind so weit wie möglich als Unterprogramme aufgebaut. Bei auftretenden Fehlern ist es jedoch erforderlich, aus beliebig geschachtelten Unterprogrammen herauszuspringen. In diesen Fällen und bei einigen Befehlen, wo das Ende einer Routine in einem Unterprogramm erkannt wird, wird nach der Adresse zurückgekehrt, die auf dem Stack als oberste Adresse steht, also bei 10EEh. Der Stackpointer wird auf 10F0h gesetzt.

## Voreinstellungen bei Einsprung nach EA5Eh:

Stackpointer: 10F0h, EA5Eh auf Stack, (FLAGS)=00h, (HEADNR)=00h, (DLYCMT)=19h = Standard-Aufzeichnung für CMT; UMLAD-Programme nach 1110h; CMT-Stop; Floppy-Stop; Neue Zeile, Ausgabe des Promts "#"; Mit Cursor Eingabe abwarten

## Symbole

(FLAGS): Der Name oder Wert in Klammern gibt die Adresse der Speicherstelle an, auf die zugegriffen wird.

U = als Unterprogramm aufgebaut

F = Das Programm enthält Fehler-Abfragen, die über die Fehlermeldung nach (10EEh) verzweigt.

<SHIFT+BREAK> wird auch als Fehler behandelt.

(10EEh) = Das Programm kehrt auf die in 10EEh stehende Adresse zurück.

M = Rückkehr in die Monitor-Hauptschleife (nach den Voreinstellungen).

Die mit Stern \* gekennzeichneten Register werden vom Unterprogramm geändert.

# = Alle Register sind geschützt.

Befehl	Startadr.	Typ	Bemerkungen
1..4	E010h	M	Floppy-Befehle: IX=CEE9h; (IX)=00h..03h
A	EBEAh	M	ASCII:Bit 1+7 von (FLAGS) = H
B	EBF7h	U	Soundwork: Invertiert Bit 0 von (SWRK)
C	EC20h	(10EEh),F	Character-Ausgabe
D	EC2Eh	(10EEh),F	Dump (Hex.+ Char.)
E	EC01h	Jump	BASIC- Restart bei 0000h
F	EC53h	Jump	Floppy-Boot
G	EC19h	U,F	Call/Jump an eingegeb. Adresse
H	EFF6h	(10EEh)	Hören der Kassette
I	EC6Ah	U,F	Einlesen aus der RAM-Card
J	EC19h	U,F	wie "G"
L	F022h	U,F	CMT-Laden
M	ECA6h	(10EEh)	Memory-Change, mit Pfeiltasten links/rechts auf/ab
O	EC6Eh	U,F	Schreiben in die RAM-Card
P	EBEDh	M	Drucken +Schirm: Bit 1 von (FLAGS)=H
Q	EF60h	U	Randfarbe:(FRAME) wird inkrementiert
R	EBE2h	M	RAM ein:Bit 4 von (FLAGS) = H, CMT auf Rapid: (DLYCMT) = 09h
S	F12Ch	U,F	CMT-Schreiben, auch Verifizieren
T	EF6Fh	U,F	Transfer eines Datenblocks
U	EE65h	U,F	Unterschied zweier Datenblöcke
V	EEA7h	U,F	Vergleichen = suchen eines Wortes (1..4 Bytes) in einem Datenblock
W	EF18h	U	Write:Text (Sharp oder ASCII) ins RAM schreiben, Ende mit <CR>
X	EF87h	U,F	Transfer wie T, CG-ROM (1000..1FFFh) u.CG-RAM (C000..CFFFh) eingeschaltet.
Y	EFAFh	U	Addieren u.Subtr. von Hex.-Zahlen



(IX+11)	CEF4h	letztes Motor-Controller-Kommando
(IX+12)	CEF5h	Motor-Code, 0= aus
(IX+13)	CEF6h	Laufwerk-Flag für Lw. 0
	CEF7h	1
	CEF8h	2
(IX+16)	CEF9h	3
DSKFLG	CEFCh	(IX+13h)= Diskflags: 80h =Schreiben, 1= Lesen, 0= Testlesen
	CEFEh	Fehler-RETURN-Speicher für MZ-Monitor
HEADER	CF00h	aktueller Header einer Datei
HBUF	CF20h	Header-Buffer
ALOC	CF40h	Allocation-Table, aus (DF00h) zur Bearb. kop.

Zwischen-Speicher für die Bearbeitung der Directory einer Floppy ist D000h bis E7FFh. In diesen Bereich wird die Spur 0, ab Sektor 0 bis Spur 1, Sektor 7 kopiert, je nach Kommando verändert und zurückgeschrieben.

Folgende Zuordnungen sind wichtig:

Im folgenden wird nur noch von absoluten Sektoren gesprochen. Da jede Spur 16 oder 0Fh Sektoren hat, sind bei 40 Spuren pro Seite insgesamt 500h Sektoren möglich. (00h..4FFh).

D000h: Header der Bootdatei: Ein gültiger Eintrag beginnt mit 03h und "IPLPRO",vgl. im ROM E4CAh. Sonst Aufbau wie normaler Header.

D100h.. DEFFh: Freier Platz, oft genutzt für ein Bootprogramm, wie der Booter des CP/M .

DF00h: Allocation-Tabelle: Ab DF06h wird für jeden belegten Sektor ein Bit gesetzt und beim Löschen einer Datei wieder zurückgesetzt.

DF00h: Kenn-Byte ("volume"), muß mit (E001h) übereinstimmen

DF01h: Der erste freie Sektor (30h bei SHARP, 18h bei Willy's).

DF02h/03h: Die belegten Sektoren, einschl. der 18h/30h reservierten.

DF04h/05h: Der maximale Sektor (45Fh bei Sharp, 500h bei Willy's entsprechend 70 bzw. 80 Spuren).

E000h Wert 80h (Kennung)

E001h wie (DF01h)

E020h... Datei-Einträge mit 20h Länge und folgendem Aufbau:

E7E0h: 1.Byte: Kennung: 00h = ungültig, gelöscht; 01h = OBJ-Datei; 02h = BASIC-Datei; 03h = BASIC-serielle Datei (BSD); 04h = BASIC-Random-Datei (BRD); 05h = RB-Datei (?), BASIC-Datei von Cassette überspielt; > 80h = ungültig, aber nicht löscherbar.

2. bis maximal 18. Byte: Name (max. 16 Zeichen)+ <CR>. Die Bootdatei in D000h beginnt mit 03h, "IPLPRO" und Namen (max. 12 Zeichen) + <CR>

19. Byte =01h: LOCKED (entfällt bei Bootdatei)

21/22. Byte = Länge

23/24. Byte = Lade-Top-Adresse

25/26. Byte = Start-Adresse

27/30. ? nicht genutzt ; LOCKED-Markierung in BASIC

31/32. Byte = absoluter Startsektor

Dateien des Typs 3 und 4 belegen zunächst nur einen Sektor. Der folgende, absolute Sektor ist in der Datei im 255/256. Byte eingetragen! Das Ende ist mit 00h, 00h markiert. - Diese Routine ist in Willy's nicht enthalten.

Die wichtigsten Unterprogramme

Alle Unterprogramme des 700er Monitors sind aus Kompatibilität übernommen worden, nicht jedoch der 700er Monitor selbst. Dadurch wurde Platz für neue Programme geschaffen. Die meisten Programme enthalten eine Fehler-Routine.

## 1. Floppy-Routinen

TRACK 0070h Floppy lesen, schreiben und testlesen  
Eingangsdaten:  
IX  
(IX)\* = anzusprechende Floppy  
(DE)\* = absoluter Startsektor  
(HL)\* = Topadresse im RAM  
(BC)\* = Länge  
(DSKFLG)\* = 80h Schreiben, 01h Lesen, 00h Testlesen.  
Nach Schreiben erfolgt ein Testlesen. Das RAM ab  
D000h wird erst ein-, dann wieder ausgeschaltet !

FDIRWR FB75h Schreiben des Directory-Bereichs D000h .. E7FFh  
FDIRLD FB7Fh dto., laden  
ANZDIR1 FBF0h Ausgabe der Header ab (HEADNR), wenn OBJ  
ANZDIR7 FBF4h Ausgabe aller gültigen Header ab (HEADNR)  
ANZNA3 FD6Fh Ausgabe von Typ in (HL) und folgendem Dateinamen  
FREIPL FAB7h Die Dir. wird geladen, und untersucht, ob der  
Eintrag in (HBUF) schon vorkommt. Wenn nicht,  
wird in (HEADNR) der freie Dir.-Platz zugewiesen.

SEKTOR FD79h Die Spur in A und Sektor in C (00h..0Fh) wird in  
den absoluten Sektor in DE umgerechnet. A\*,C\*,DE\*

## 2. Ausgabe-Programme

ANZDEL 01BAh # Der Cursor rückt nach links, das Zeichen wird  
gelöscht.

PRINT F430h \* Ausgabe von (A) an den Drucker  
ANZPRT 0F5Eh # Ausgabe von (A) an den Bildschirm:  
Wenn Bit 0, (FLAGS) = H als ASCII-Zeichen  
Wenn Bit 0, (FLAGS) = L als Sharp-Zeichen  
Wenn Bit 1, (FLAGS) = H auch an den Drucker  
als ASCII-Zeichen. Als Steuercode wird nur  
<CR> = 0Dh verarbeitet. Wenn ein <CR> ausgegeben  
werden soll und die <SPACE>-Taste gedrückt ist,  
wird bis zum nächsten Tastendruck gestoppt.  
Alle folgenden Ausgabe-Prg. benutzen ANZPRT !

KEINCR 0164h A\* <CR> wird als Leerzeichen ausgegeben  
CR 017Bh A\* Ausgabe von <CR>  
SPACE 0177h A\* Ein Leerzeichen  
TWOSP 0174h A\* Zwei Leerzeichen  
ASCDUM EC48h \* Gibt aus (HL) B Zeichen im Hex.-Code aus und  
anschließend ein <CR>.

NIBHEX 0192h A\* Gibt das untere Nibble von A als Hex. aus  
PRTA 0189h A\* Gibt A als Hex. aus  
PRTASP 017Fh dto., zusätzlich SPACE = Leerzeichen  
PRTHL 0184h A\*,HL\* Gibt HL als Hex. aus  
PRTHLSP 016Fh dto., zusätzlich SPACE  
PRTHLCR 016Ah dto., + <CR>  
PRTHLTW 015Fh dto., + 2 SPACE  
WADR 0155h A\* wie PRTHLTW, springt aber nach (10EEh), wenn  
HL < DE ist.

TEXT 019Dh # Nach CALL TEXT werden die unmittelbar folgenden  
Zeichen als Text in Sharp-Code interpretiert und  
ausgegeben. Bit 0, (FLAGS) wird zurückgesetzt und  
Ende wieder gesetzt, wenn Bit 7, (FLAGS) gesetzt  
ist. 00h = NOP wird als Textende interpretiert  
und das aufrufende Programm fortgesetzt.

### 3. Eingabe-Routinen

??GET	06E7h		Wartet mit blinkendem Cursor auf die Eingabe und liefert in A ein Zeichen im Sharp-Code. Bei <SHIFT+BREAK> wird "Break !" ausgegeben und nach (10EEh) gesprungen. Bei Eingabe <CR> ist das Zero-Flag gesetzt.
?BREAK	0748h	A*	Bei <SHIFT+BREAK> wie ??GET, sonst RETURN
?ADR	E222h	*	Fragt nach Topadresse, Startadr. und der Lage im RAM und legt die Daten im HBUF und CMT-Header zum Abspeichern richtig ab.
?NAME	E9B7h	*	Erlaubt die Eingabe eines Programm-Namens, welcher in HBUF und ab 10F1h eingetragen wird. Es wird OBJ = Typ 1 angenommen.
INP2	EB3Fh	*	Eingabe einer zweistelligen Hex.-Zahl nach A.
INP4	EB27h	*	Vier Stellen nach HL
AB	EB0Ch	*	Text-Ausgabe " ab" und INP4
NACH	EB15h	*	"nach" +INP4
VON	EB1Fh	*	" von" +INP4
VONBIS	EB84h	*	" von" +INP4, " bis" +INP4: DE = (ADRVON), HL = (ADRBIS), BC = (LAENGE) = HL-DE-1
DATA	EB36h		"Data "+INP2

### 4. Kassetten-Routinen

Die Routinen entsprechen weitgehend den CMT-Programmen, wie sie bereits im 700er Monitor enthalten sind. Sie wurden soweit geändert, daß die Aufzeichnungs- und Lesegeschwindigkeit durch (DLYCMT) bestimmt werden kann.

WRDAT	F1D0h	Schreibe Datenblock
WRINF	F1DFh	Schreibe Informationsblock
VFINF	F287h	Verifiziere Informationsblock
RDINF	F28Ah	
RDDAT	F276h	
VFDAT	F273h	

### 5. Fehler-Routinen

Es werden weitgehend die Fehlertexte für das BASIC 800 verwendet.

--	0D4Dh	F	Es wird der im Akku A angegebene Fehlertext am Bildschirm ausgegeben. (A =1..46h); vgl. S. A-28 im Handbuch.
FESUM	0D2Dh	F = 70	
FEHARD	0D30h	41	
FESPAC	0D33h	53	
FEXIST	0D36h	42	
FEREAD	0D39h	54	
FEDAT	0D3Ch	3	
FEPRO	0D3Fh	46	
FEFIL	0D42h	51	
FEMISM	0D45h	52	
FELPT	0D48h	65	
FEMEM	0D4Bh	6	
FEBREA	0D8Eh	Fehler "Break !"	
FENOT	0D93h	" Make ready FD "	
ROMFEH	0D96h	Es wird der Text ausgegeben, auf den (DE) zeigt und nach (10EEh) gesprungen. Textende = <CR>	

## 6. Sonstige Routinen

Die Routinen für Speicher-Zugriffe stehen ab EBABh und werden vom Monitor nach 1110h kopiert. Wenn Bit 4, (FLAGS) gesetzt ist, wird vor einem Speicher-Zugriff das RAM ab 0000h und ab D000h ein- und danach wieder abgeschaltet.

WRITE	111Dh	Schreibt A nach (HL), HL = HL + 1
READ	1124h	Liest A aus (HL), HL = HL + 1
LDIRU	112Bh	Kopiert ab (HL) nach (DE) mit der Länge BC
LDDRU	1132h	wie LDIRU, aber HL und DE werden dekrementiert.
CPIRU	1139h	Siehe Befehl CPIR, Z 80
COMP	1140h	Vergleicht A mit (HL)
SHAASC	07A8h	Setzt Sharp-Code in ASCII um. Unbekannte Codes und Steuerzeichen außer <CR> werden SPACE.
ASCSHA	07C9h	ASCII in Sharp-Code
MONVGL	EA5Dh	(HL) = Zeichentabelle, A = Zeichen, (DE) = Sprungadressen (je 2 Bytes), B = Anzahl. Springt nach der Adresse, die dem Zeichen (HL) = A entspricht. Wird das Zeichen nicht gefunden, erfolgt Fehlermeldung und Sprung nach (10EEh).
PUSPOP	F45Ah	Der Aufruf von PUSH-POP in einem Unterprogramm bewirkt, daß die Register IX, HL, BC, DE und AF auf den Stack geschoben werden und beim RETURN dieses Unterpr. die alten Werte zurückerhalten.

## Probleme mit Programmen und Willy's Monitor

DP-8001.0

Dieses Boot- und Ladeprogramm fragt ab, ob im im ROM auf Adresse F002 der Wert "CD" (hex.) steht. Der Sinn ist unklar. Ohne die beschriebene Korrektur ist das Programm nicht betriebsfähig.

Die einfachste Abhilfe besteht darin, den Z 80-Befehl

```
3A 02 F0      LD A,(F002h) abzuändern in
3E CD 00      LD A,CDh      NOP.
```

Das geschieht mit dem Monitor folgendermaßen:

1. Floppy in Laufwerk 1 einlegen (Sicherungskopie vorhanden ?)
2. Directory aufrufen (Eingabe: <1> <D> )
3. Im RAM ab Adresse D127 korrigieren (Eingabe <R> <M> )

```
D127 3A 3E
D128 02 CD
D129 F0 00
D12A
```

<SHIFT+ BREAK>

Durch den Prefix <R> wird im Daten-RAM korrigiert (wichtig, weil sonst die Änderung im Bildschirmspeicher erfolgen würde!). Vor dem Start des Bootprogramms stehen die Daten ab 1227.

4. Auf die Diskette zurückschreiben:

```
Eingabe <1> <W>
Schreiben ab Spur 00..4F: <00>
Sektor 00..0F: <00>
von <D000> bis <D1FF>
```

## Disk-File-Utility Version 2.6

Die Utility unterstellt, anders als die BASIC-Disketten, daß die Direktory-Einträge über den Sektor 18h hinausgehen. Will man Abstürze dieses Programms vermeiden, muß auf die Diskette von Sektor 18h bis 2Fh leer sein. ( = 00h), was man mit einem Dummy-Programm oder durch Formatieren mit FDCOPY erreichen kann.

## Supermonitor (Erweiterungs-Software)

Nur mit Willy's Monitor-ROM lauffähig!

Lage: C200 bis CDFD

Zweck: Daten-Transfers

Befehle: Alle Befehle von Willy's Monitor und zusätzlich folgende:

a Wandelt ASCII- in SHARP-Text  
d Wandelt Display- in SHARP-Text  
s Wandelt SHARP- in ASCII-Text  
i Invertiert die Daten (CPL)  
l Lesen der Daten, mit <CR>, bei ASCII wird BIT 7 ignoriert  
r Setzt Bit 7 auf Null. Bei WORDSTAR-Dateien ist Bit 7 von Wortendungen nach Zeilen-Umbruch-Befehlen gesetzt.  
t Tauschen eines Suchstrings mit 1..20 Bytes in einen Ersatz mit 0..20 Bytes.-Eingabe in Hex-Code, ASCII- oder SHARP-Code. Beispiele: Zeilen-Vorschub 0Ah nach 0Dh einfügen oder löschen; Ae durch Ä ersetzen. Wenn sich die Text-Länge ändert, dauert die Korrektur!

### Floppy-Befehle

d,l,s,t,w wie D,L,S,T,W aber für CP/M !

Größere CP/M-Dateien werden in Teilen zu max. 32 KB geladen. Das Saven ist bis zu einer Länge von 9F00 (fast 32 KB) möglich. Bei t und w werden Blöcke der CP/M-Diskette, entsprechend 2 KB, bearbeitet. Die Bootspur und die reservierte CP/M-Spur sind nicht erreichbar. Die Befehle t und w sind geeignet, CP/M-Disketten auf Cassetten (in Teilen von 40 KB) umzukopieren.

Beim Saven im BASIC/SHARP-Modus ist die Angabe des Dateityps vorgesehen. Typ 03 = BSD wird richtig gespeichert und geladen. Dadurch ist der Transfer von BTX-Programmen aus anderen Systemen als Textdatei leicht möglich. (Im BASIC als "Name", A laden bzw. speichern).