

22.04.1987 (c) Conitec JH

## CP/M plus Software fuer DISI

Dieser Text beschreibt die auf der Diskette gelieferten CP/M plus Programme fuer die EPROM- / RAM-Karte DISI.

--- DISI.ASM ---

Diese Datei enthaelt den Source-Code des DISI-Treiber-Programms fuer CP/M plus. Bevor die DISI-Karte in Ihrem System verwendet werden kann muessen Sie DISI.ASM fuer Ihre Anwendung konfigurieren und in Ihr BIOS einbinden.

DISI.ASM unterstuetzt bis zu vier DISI-Karten, maximal zwei logische Laufwerke und folgende Bausteine:

RAMs	:	8K*8 (z.B. 6264), 32K*8 (z.B. 63256)
EPROMs	:	2716, 2732, 2764, 27128, 27256, 27513, 27011

Der Treiber ist so ausgelegt, dass er ohne Aenderung auf jedem CP/M plus System verwendet werden kann, sofern der BIOS-Kern den Empfehlungen von Digital Research entspricht.

### Konfiguration

Am Anfang von DISI.ASM befinden sich 14 EQU-Anweisungen. mit denen der Treiber konfiguriert wird. Die Anweisungen haben folgende Bedeutung:

disia, disib, disic, disid

Geben Sie hinter diesen Labels die Basisadressen der verwendeten DISI-Karten an. Werden weniger als vier Karten verwendet, dann sind die verbleibenden Labels auf den Wert 0FFh zu setzen.

nr0, nr1

Diese beiden Labels definieren die Anzahl der verwendeten Speicherbausteine (max. 64 zusammen) fuer das erste und zweite Laufwerk. Ist nr1 auf null gesetzt, so wird nur ein Laufwerk definiert; alle sonstigen Angaben fuer das zweite Laufwerk sind dann ohne Bedeutung. Bei RAM-Bausteinen sollte immer die Anzahl der Bausteine angegeben werden, die auch physikalisch vorhanden sind. Bei EPROMs ist es hingegen sinnvoll, einige Bausteine mehr zu definieren als tatsaechlich benoetigt werden. Neue Bausteine koennen dann einfach hinzugesteckt werden, ohne dass das BIOS geaendert werden muss. Die Speicherbausteine muessen fortlaufend bestueckt sein, es gilt folgende Reihenfolge: Kartel/Z1, Kartel/Z2, ... , Kartel/Z16, Karte2/Z1 ... usw.

offset1

Die Zahl hinter diesem Label gibt die Nummer des ersten Speicherbausteins vom zweiten Laufwerk an. Beachten Sie bitte, dass die Zaehlweise bei Null beginnt. Falls die Bausteine fuer das zweite Laufwerk anders als beim Ersten gejumpt werden, muss offset1 durch vier teilbar sein. Das erste Laufwerk beginnt immer bei Baustein Nummer Null, deshalb ist hier kein Offset notwendig.

bkap0, bkap1

Diese beiden Labels definieren die Bausteinkapazitaet fuer jedes DISI-Laufwerk in K-Byte. Die Kapazitaet fuer das erste Laufwerk ergibt sich aus nr0 \* bkap0, die Kapazitaet fuer das zweite Laufwerk aus nr1 \* bkap1. Ist die Bausteinkapazitaet groesser als 32 K-Byte, wird automatisch angenommen, dass es sich um ein gebanktes EPROM handelt.

rw0, rw1

Diese beiden Labels geben fuer jedes Laufwerk an, ob es sich um eine RAM-Floppy oder um eine EPROM-Floppy handelt. "ja" definiert RAM-Floppy, "nein" definiert EPROM-Floppy.

amsk0, amsk1

Je nach Bausteintyp muss noch eine Odermaske definiert werden, die vor dem Zugriff auf die DISI-Karte mit dem Trackregister verknuepft wird. In der Jumpertabelle fuer die Speicherbausteine ist die zugehoerige Odermaske angegeben.

prof180

Dieses Label wird auf "ja" gesetzt, falls der Treiber in einem PROF-180X System eingesetzt wird. Bei allen anderen Computern ist dieses Label auf "nein" zu setzen.

## Beispiel

Sie wollen zwei DISI-Karten mit RAM- und EPROM-Laufwerk auf einem beliebigen CP/M plus Rechner einsetzen. Die RAM-Floppy umfasst 20 Bausteine vom Typ 63256, die EPROM-Floppy maximal 12 Bausteine vom Typ 27513. Als Basisadresse waehlen Sie B8 (Hex) und BC (Hex) aus. Das erste Laufwerk soll die EPROM-Floppy enthalten, das zweite Laufwerk die RAM-Floppy. Der Treiber muss demnach folgendermassen konfiguriert sein.

```
prof180    equ    nein

disia      equ    0b8h
disib      equ    0bch
disic      equ    0ffh
disid      equ    0ffh

bkap0      equ    64
amsk0      equ    00h
nr0        equ    12
rw0        equ    nein

offs1      equ    12
bkap1      equ    32
amsk1      equ    00h
nr1        equ    20
rw1        equ    ja
```

Auf der ersten Karte Z1 bis Z12 werden die EPROMs bestueckt, auf der zweiten Karte und den restlichen Sockeln der ersten Karte die RAMs.

## Jumper

DISI.ASM arbeitet grundsaeztlich mit einer Sektorlaenge von 128 Byte, J1 bis J5 muessen demnach in Stellung 2-3 stehen.

Die Bausteinjumperung ergibt sich aus folgender Tabelle:

```
xxx        : Jumperstellung unwichtig
---        : kein Jumper
1-2 / 2-3  : entsprechende Pins bruecken
```

Type	Jx1	Jx2	Jx3	Jx4	Odermaske (amsk0/amsk1)
EPROM 2716	xxx	xxx	2-3	xxx	10h
EPROM 2732	xxx	xxx	2-3	xxx	00h
EPROM 2764	---	---	xxx	2-3	00h
EPROM 27128	---	---	1-2	2-3	00h
EPROM 27256	---	1-2	1-2	2-3	00h
EPROM 27513	---	2-3	1-2	2-3	00h

EPROM 27011	---	2-3	1-2	2-3	00h	
RAM 6264	xxx	2-3	1-2	1-2	40h	
RAM 43256		2-3	2-3	1-2	1-2	00h

Wenn RAMs oder gebankte EPROMs bestueckt werden muss J7 unbedingt gebrueckt sein.

### Einbinden ins BIOS

Nachdem die Konfiguration, wie oben angegeben, durchgefuehrt wurde (Texteditor), kann DISI.ASM mit dem Assembler RMAC assembliert werden.

#### RMAC DISI

Auf der Diskette muessen sich allerdings noch die Macro-Libraries CPM3.LIB und Z80.LIB (beim PROF-180X statt Z80.LIB HD64180.LIB) befinden. Der Assembler liefert als Ergebnis die Datei DISI.REL, die ueber die global (public) definierten Labels fdisi0 und fdisi1 (fdisi1 nur wenn zwei Laufwerke konfiguriert wurden) mit dem restlichen BIOS verbunden werden kann. Diese Verbindung geschieht ueber das Drive Table Modul DRVTBL.ASM (beim PROF-180X heisst diese Modul P6.ASM). Tragen Sie also fdisi0 (und fdisi1, wenn zwei Lauferke konfiguriert wurden) in DRVTBL.ASM ein. Durch die Stellung des Eintrags in DRVTBL.ASM ergibt sich die Laufwerksbezeichnung. Conitec schlaegt als nichtfluechtige RAM-Floppy den Laufwerskbuchstaben F und als EPROM-Floppy den Buchstaben G vor. Nachdem DRVTBL.ASM neu assembliert wurde, k|nnen die BIOS-Module nach der Angabe Ihres Computerherstellers gelinkt werden (mit zusaetzlichem Modul DISI). F}r den PROF-80 von Conitec sieht der Linkvorgang z.B. so aus:

#### LINK

```
BNKBIOS3[B]=BIOSKRNL,SCB,BOOT,CHARIO,MOVE,DRVTBL,FDPROFBK,DISI
```

Der Linker erzeugt ein neues BIOS, dass wie gewohnt mit GENCPM zu einem neuen CPM3.SYS zusammengebunden wird.

#### --- ERASER.COM ---

Bevor zum ersten Mal auf eine nichtfluechtige statische RAM-Floppy zugegriffen wird, muss das Directory so mit Daten gefuellert werden, dass das CP/M die RAM-Floppy als geloescht erkennt.

Das Programm ERASER.COM loescht auf diese Weise die RAM-Floppy. Nach dem Start fragt ERASER nach dem Laufwerksbuchstaben der RAM-Floppy.

\*\*\* Achtung \*\*\*

ERASER loescht unter Umst{nden auch normale Disketten, wenn deren Laufwerksbuchstabe angegeben wird. Als kleinen Schutz haben wir ERASER so konzipiert, dass bei Angabe der Laufwerksbuchstaben A-D mit einem Fehler abgebrochen wird. ERASER liegt im Turbo-Pascal Source-Code vor, wir empfehlen dringend, ERASER so abzuaendern, dass nur die RAM-Floppy auf der DISI-Karte geloescht werden kann (die Aenderung ist im Source-Code beschrieben).

\*\*\* Wichtig \*\*\*

Da die DISI-Laufwerke im Treiber mit festem Medium definiert wurden, bekommt CP/M plus das Loeschen der RAM-Floppy erst nach einem erneuten Start mit. Nach dem Aufruf von ERASER muss CP/M deshalb neu gebootet werden.

--- GPF.COM ---

Wie jedes andere Betriebssystem ben|tigt CP/M die Daten auf den Massenspeichern in einer bestimmten Struktur. Da diese Struktur beim Schreiben auf die Floppy automatisch angelegt wird, braucht man sich im allgemeinen nicht darum zu kuumern. Anders ist dies bei einer EPROM-Floppy, da das Betriebssystem von einer EPROM-Floppy nur lesen kann.

Die Dateien, die in einer EPROM-Floppy untergebracht werden, muessen also die gleiche Struktur haben, die entstehen wuerde, wenn das CP/M direkt in die EPROMs schreiben koennte. Das Programm GPF.COM erzeugt genau diese Struktur.

GPF.COM legt ein virtuelles Laufwerk an, das durch eine Anzahl von Ziel-Dateien representiert wird. Diese Ziel-Dateien werden dann in EPROMs gebrannt und mit Hilfe der DISI-Karte als EPROM-Laufwerk benutzt.

GPF arbeitet mit zwei Disketten-Laufwerken, dem Quell-Laufwerk und dem Ziel-Laufwerk. Nach dem Start liest GPF vom Quell-Laufwerk die CP/M-Dateien, die in die EPROM-Floppy kopiert werden sollen und legt dabei auf dem Ziel-Laufwerk das virtuelle Laufwerk an, representiert durch die Ziel-Dateien.

Die Ziel-Dateien sind durchgehend nummeriert und heissen DISI000.HXC ..... DISI063.HXC. Die Groesse der Ziel-Dateien entspricht der Groesse der verwendeten EPROMs und koennen deshalb direkt in EPROMs gebrannt werden.

Stellt GPF fest, dass auf der Ziel-Diskette nicht mehr genuegend Platz fuer eine neue Ziel-Datei ist, fordert es zum Diskettenwechsel im Ziel-Laufwerk auf. Zum Schluss wird dann

allerdings noch einmal die erste Ziel-Diskette angefordert, da in die Ziel-Datei DISI000.HXC noch das Directory geschrieben werden muss. Soll eine Datei kopiert werden, die sich nicht auf dem Quell-Laufwerk befindet, fordert DISI ebenfalls einen Diskettenwechsel im Quell-Laufwerk an. Durch diesen Mechanismus ist GPF in der Lage, ein virtuelles Laufwerk zu verwalten, dass in seiner Kapazitaet die der Disketten weit uebersteigt.

GPF.COM erhaelt seine Anweisungen aus einer Datei namens GPF.CTR. Es handelt sich dabei um eine gewoehnliche Textdatei, die mit Wordstar, ED oder aehnlichen Editoren erstellt werden kann. GPF.CTR muss sich beim Aufruf von GPF.COM auf der Default-Disk befinden. GPF.CTR hat folgenden Aufbau:

Beispiel fuer GPF.CTR

```
-----  
16           ; Groesse der EPROMS in K-Byte  
20           ; Maximale Anzahl der EPROMS  
A           ; Quell-Laufwerk  
B           ; Ziel-Laufwerk  
2           ; logische CP/M Blockgroesse in K-Byte  
256         ; Anzahl der Directory-Eintraege  
TURBO.COM   ; Quell-Datei Nr.1  
TURBO.OVR   ; Quell-Datei Nr.2  
TURBOMSG.MSG S ; Quell-Datei Nr.3, setze SYS-Atribut  
PIP.COM SAR ; Quell-Datei Nr.4, setze alle Attribute  
DATE.COM   ;  
DIR.COM    ;  
SID.COM    ;  
usw.
```

Die beiden Parameter <log.Blockgroesse> und <Anzahl der Directory-Eintraege> muessen mit den Werten im CP/M Treiber uebereinstimmen. Am einfachsten ist es, wenn beide Parameter weggelassen werden (geht nur gemeinsam). GPF.COM setzt diese Werte dann automatisch so, wie sie auch vom Treiber DISI.ASM gesetzt werden. Beachten Sie bitte, da~ GPF.COM nur maximal 512 DIR-Eintr{ge anlegen kann. Falls Sie eine EPROM-Floppy mit mehr Eintr{gen definiert haben, bricht GPF.COM mit einer Fehlermeldung ab.

Es besteht die Option, die Attribute Read-Only, System und Archiviert, beim Kopieren ins vituelle Laufwerk zu setzen, alle eventuell vorhandenen Attribute der Quell-Datei werden automatisch geloescht. Die Angabe der Attribute erfolgt durch die Buchstaben 'R' fuer Read-Only, 'S' fuer System und 'A' fuer Archiviert. Die Atributzeichen werden durch ein Blank getrennt hinter den Dateinamen geschrieben.

Erscheint in einer GPF.CTR Zeile ein Semikolon, so wird der Rest der Zeile einschliesslich des Semikolons als Kommentar betrachtet.

Die Anzahl der von GPF.COM erzeugten Ziel-Dateien richtet sich nach der Anzahl und Grösse der Quell-Dateien. In der Steuer-Datei GPF.CTR sollte aber immer die maximal mögliche EPROM Anzahl angegeben werden. Die von GPF.COM erzeugten Zielformate sind nämlich so aufgebaut, dass jeweils das erste und letzte EPROM ueberprogrammiert werden koennen, falls neue Dateien in die EPROM-Floppy gebracht werden sollen.