

## BETA - DISK - KEIN GEHEIMNIS

Ein Diskettenlaufwerk ist schon eine feine Sache. Nun entfaellt eine bis zu acht Minuten lange Wartezeit, die unter Umstaenden mit der Meldung "Tape loading error" ein nochmaliges laden erforderlich macht. Es gibt keine amoklaufenden Microdrives mehr, deren Datensicherheit nicht gerade begeistert und die darueber hinaus - bei mageren 90 KByte Speicherkapazitaet - auch noch relativ teuer sind.

Aus der Palette der verschiedene Systeme, die fuer den Spectrum mittlerweile angeboten werden, weist das Beta-Disk von Technology-Research einige Vorteile auf.

- hohe Speicherkapazitaet von knapp 640 KByte pro Diskette (entspr. etwa dem Inhalt einer C-60-Cassette)
- Industriestandard, da Shugart-kompatibel. Das bedeutet, dasz bei Systemwechsel die Diskettenlaufwerke weiterverwendet werden koennen (etwa fuer Atari ST, IBM PC und Kompatible)
- Bis zu vier Laufwerke lassen sich gleichzeitig anschliessen, nach 40 oder 80 Tracks oder 3.5" und 5.25" voellig gleichgueltig
- Die Beta-Disk funktioniert auch mit angeschlossenen Interface I
- Es werden nur etwa 112 Byte RAM fuer zusaetzliche Systemvariablen bzw. Puffer benoetigt

Fuer die Beta Disk gibt es inzwischen eine ganze Reihe von Betriebssystemen (DOS= Disc Operating System). Alleine von Technology-Research sind mir die Versionen 3.0, 4.0, 4.05, 4.07, 4.11, und 5.0 bekannt, wovon letztere auch zum Spectrum 128 kompatibel ist. Daneben existieren eine grosze Anzahl von modifizierten Ausfuehrungen, die von findigen Tueftlern fuer bestimmte Zwecke geschrieben wurden, sowie ein DOS, das unter der Bezeichnung VISION seit kurzem zu kaufen ist und eine Art Benutzeroberflaeche a la Atari ST bietet. Die DOS Versionen sind bis auf wenige Ausnahmen absolut kompatibel. Die Ausnahmen beschraenken sich im wesentlichen auf die unterschiedlich FORMAT-Prozedur, die Behandlung sequentieller und Random-Access-Files, die ab DOS 4.0 als Option vorhanden sind, sowie den RAM-Image-SAVER (auch bekannt als Magic Button), der ebenfalls ab Fassung 4.0 existiert.

Wenn Sie beim laden von Fremddisketten unerklaerliche Verzoegerungen bei Code-Blocken bemerken, so liegt das an der ab Version 4.0 geaenderten Formatierungsroutine. Diese Formatierungsroutine ist nicht ganz kompatibel zur 3.xx-Ausfuehrung. Mit 3.0-DOS formatierte Disketten laufen auf allen Versionen, waehrend die mit den neueren Fassungen formatierten Scheiben Schwierigkeiten machen, wenn Sie noch ueber eines der ersten Beta-Disk-Systeme verfuegen.

Die einfachste Abhilfemoeglichkeit besteht darin, alle Disketten mit dem alten DOS zu formatieren. Die etwas aufwendigere ist, eine der neuen Versionen mit ins EPROM des Interface einzubrennen und per Schalter umzuschalten. Da das Original-EPROM nur vier der vorhandenen acht KByte belegt, muss man sich nicht einmal ein neues EPROM anschaffen. Noch komfortabler wird die Geschichte mit einem 16 KByte EPROM oder gar etwa einem 32 KByte EPROM. Schwierig ist dann, noch einen Platz fuer den Umschalter in dem mit zwei Platinen vollgetopften Gehaeuse zu finden. Aber immerhin kann man dann ueber vier verschiedene DOS-Versionen per Umschalter verfuegen, wenn man beruecksichtigt, dasz ab 4.xx das DOS etwa 8 KByte verschlingt, da verschiedene Utilities mit aufgenommen wurden, die sonst von Disketten nachzuladen waren. Der Umschalter wird so installiert, dasz er die verschiedenen Baenke des EPROMS entsprechend umschaltet.

Die ab DOS 4.0 vorhandene Moeglichkeit, mit sequentiellen oder Random-Access-Files zu arbeiten ist eine wertvolle Hilfe fuer diejenigen, die grosse Datenmengen verwalten wollen. Bis zu 600 KByte koennen solche Dateien umfassen. Den letzten Eintrag einer solch grossen Datei findet der Spectrum in Windeseile. Er gibt ihn innerhalb einer Zehntelsekunde aus. Sie muessen jedoch darauf achten, dass diese Dateien nur mit dem DOS wieder eingelesen werden, mit dem die Abspeicherung stattfand, denn leider sind alle Ausfuehrungen in diesem Punkt inkompatibel. Unter Umstaenden tritt in solchen Faellen ein irreversibler Datenverlust auf. Gleiches gilt fuer BACKUPS, die mit dem Magic Button erstellt wurden. Auch hier sind die verschiedenen Versionen nicht miteinander vertraeglich.

#### SCHEIBE VON FORMAT

Das Beta-Disk-System teilt die Diskette, je nach Art des Laufwerks in 40 oder 80 Spuren zu je 16 Sektoren auf. Ein Sektor enthaelt 256 Byte. Das ergibt pro Spur oder Track eine Kapazitaet von  $16 \cdot 256$  Byte, also 4 KByte. Bei einem doppelseitigen Laufwerk mit 80 Tracks pro Seite bedeutet dies eine Gesamtkapazitaet von 640 KByte fuer eine Diskette. In Wirklichkeit werden fuer jede Spur jedoch knapp 6 KByte verbraucht, da das System schon beim Formatieren eine Menge von Daten auf der Scheibe ablegt, wie Prufsummen, Sektorkennziffern usw. Dem Benutzer stehen auch nicht volle 640 KByte zur Verfuegung, sondern nur 636 KByte davon. Die fehlenden vier KByte werden fuer die Buchfuehrung des Systems benoetigt. Dafuer ist Track 0 reseviert.

#### DEM TRACK NULL AUF DER SPUR

Immer, wenn Sie sich mit CAT das Inhaltsverzeichnis einer Diskette ansehen, faehrt der Lesekopf auf die Spur NULL und gibt die dort in den Sektoren 0 bis 7 stehenden Dateien an den Rechner aus. Mit andern Worten, diese Sektoren enthalten den Katalog, in dem jedes File mit seinen Kenndaten eingetragen ist. Ein Eintrag besteht aus 16 Byte und sieht folgendermassen aus.

BYTE	INHALT
0-7	Name des Files
8	File Typ (C=Code, B=BASIC, D=DATA, #=seq./rand. acc. Files ab Vers. 4.0)
9-10	enthaelt in Low-/High-Byte-Form entweder Startadresse bei Code oder effektive Laenge bei BASIC oder die Laenge des Feldes bei DATA
11-12	in Low-/High-Byte Form die RANDOMIZE USR Adresse eines Code-Files oder BASIC-Laenge ohne Variablen oder ab Version 4.0 die effektive Laenge des Code-Files
13	Anzahl der Sektoren, die das File belegt
14	Sektornummer, ab der das File beginnt
15	Nummer des Tracks in dem das File beginnt

Die restlichen Sektoren 10 bis 15 werden vom System nicht benutzt, erfreuen sich allerdings bei einigen Programmierern besonderer Beliebtheit, da sie sich fuer allerlei Gemeinheiten (Kopierschutz, Unkennlichtmachen des Katalogs usw.) einsetzen lassen.

#### DOS PRAXIS

Bis auf 256 Byte, die in den Adressen 15360 bis 15615 untergebracht sind, sieht man vom DOS normalerweise nichts. Der Rest der vier bzw. acht KByte ist unter dem Sinclair-ROM versteckt und wird bei Bedarf ein- oder ausgeblendet. Die Funktionen des DOS bestehen aus einer Reihe von Routinen. Ab Ausfuehrung 4.00 haben diese ihre eigenen Kennnummern, mit denen sie, aehnlich wie bei CP/M, aufgerufen werden koennen. Betrachten wir zunaechst das alte DOS mit der Versionsnummer 3.xx. Um Sektoren lesen oder schreiben zu koennen, muessen erst einmal einige CPU-Register mit den noetigen Parametern geladen werden.

#### Register     Parameter

Register	Parameter
HL	Zieladresse eines zu ladenden Sektors oder Quelladresse abzulegender Bytes
B	Anzahl der Sektoren, die gelesen oder geschrieben werden sollen
D	Nummer des Tracks, in dem das File steht (lesen) oder stehen soll (schreiben) (0-159)
E	Nummer des ersten Sektors, ab dem das File beginnt (lesen) oder beginnen soll (schreiben) (0-15)

Nachdem die Parameter gesetzt sind, koennen wir nun daran gehen, die entsprechenden DOS-Funktionen aufzurufen.

```
CALL 15530      DOS einschalten
CALL 15369      Sektoren lesen bzw.
CALL 15572      Sektoren schreiben
CALL 15550      DOS ausschalten
RET            zurueck zur Ebene des Hauptprogrammes
```

Um etwa ein Titelbild zu laden gehen Sie wie folgt vor:

```
LD HL,16384     Zieladresse (hier Bildspeicher)
LD B,27        Anzahl der zu lesenden Sektoren (27*256=
               6912)
LD C,x         x = Nummer des Tracks, in dem das Bild
               steht
LD E,y         y = Nummer des Sektors, in dem das Bild
               beginnt
CALL 15530     DOS einschalten
CALL 15369     Routine "Sektoren lesen" aufrufen
CALL 15550     DOS wieder abschalten
RET           zurueck zum Basic
```

Mit diesen Angaben ist es moeglich, jedes File auf der Diskette unzweideutig zu finden. Geben Sie folgendes kleine Basic-Program ein, um dies selbst einmal zu sehen:

```
10 CLEAR 32767: RANDOMIZE USR 15363: REM LOAD "BILD" CODE 32768,
  6912
20 FOR I=23773 TO 23786
30 IF I<23782 THEN PRINT CHR$ PEEK I
40 IF I<23781 THEN PRINT PEEK I
50 NEXT I
```

Statt "BILD" in Zeile 10 benutzen Sie den Namen eines Titelbildes auf einer Ihrer Disketten und laden dies gemaesz den Vorgaben in Zeile 10. In den Adressen 23773 bis 23786 werden Sie nun folgende Daten finden: BILDNAME/C/O 128/027/x/y (x und y sind Sektor-/ Spurnummer des Bildes auf Ihrer Diskette). Dieses Program funktioniert, weil bei jedem Ladevorgang das DOS eine Kopie des Katalogeintrages in die RAM-Adressen 23773 bis 23786 hineinschreibt.

Vielleicht haben Sie sich einmal gewundert, warum Sie kein File mehr abspeichern konnten, obwohl die Diskette noch gar nicht voll war. Die maximal moeglich Anzahl der Files, die sich auf einer Diskette speichern laeszt, betraegt 128. Ein Rechenexempel erklart, warum das so ist. Wir sagten, der Katalog steht in den Sektoren 0 bis 7. Jeder Eintrag hat die Laenge von 16 Byte. Da ein Sektor 256 Byte umfasst, passen in einen 16, in acht Sektoren folglich 128 hinein. Abhilfe schaffen hier doppelseitige Disketten, die man einseitig formatiert und dann eben von Hand umdreht. So hat jede Seite ihren Katalog mit je 128 Eintragungsmoeglichkeiten.

Sektor 8 der Spur 0 ist der sogenannte Organisationssektor. Er wird nach jedem Schreib- oder Loeschvorgang aktualisiert. Jede Veraenderung auf der Diskette musz hier festgehalten werden, damit das System hinterher noch etwas wiederfindet. Die Eintragungen in diesem Sektor beginnen ab Byte 225:

BYTE	INHALT
E1	225 Nummer des naechsten freien Sektors
E2	226 Nummer des naechsten freien Tracks
E3	227 Diskformat: Bit 0->0 40 Tracks Bit 0->1 80 Tracks Bit 1->0 einseitig Bit 1->1 zweiseitig
F4	228 Anzahl aller Files einschliesslich der geloeschten
E5-E6	229-230 Anzahl der freien Sektoren
F7	231 Anzahl der Sektoren pro Track
E8-E9	232-233 enthalten immer 0
F1-F2	234-242 acht Zeichen fuer Paswort (Version 3.xx)
F3	243 enthlaelt immer 0
F4	244 Anzahl der geloeschten Files
F5-F6	245-252 acht Zeichen Diskname

Sektor 9 der Spur 0 dient dem System als Puffer. Wenn das DOS im Katalog herumsucht, wird der Inhalt des Printerpuffers zwischengespeichert, denn dieser ist dabei als Kurzzeitspeicher erforderlich. Sein Inhalt wird nach der Operation wieder zurueckgeschrieben. Vorsicht ist also bei schreibgeschuetzten Disketten geboten. Der Printerpuffer kann bei diesen Disketten nicht gerettet werden und verschwindet unter Umstaenden auf Nimmerwiedersehen.

Die restlichen Sektoren 10 bis 15 werden vom System nicht benutzt, erfreuen sich allerdings bei einigen Programmierern besonderer Beliebtheit, da sie sich fuer allerlei Gemeinheiten

Kommen wir noch zu einem etwas praktischeren Beispiel, das den Katalog der Diskette in den Speicher einliest, so dass Sie nach Belieben darin herumsuchen koennen.

```
LD HL,40000      Zieladresse im Speicher
LD B,9          die ersten neun Sektoren der Diskette Das
LD D,0          ergibt sich aus dieser Track-Nummer
LD E,0          zusammen mit dieser Sektornummer
CALL 15530      DOS einschalten
CALL 15369      Sektoren lesen
CALL 15550      DOS abschalten
RET            zurueck zum Basic
```

Es kann sein, dass diese Verfahren bei Vorhandensein des Interface I nicht funktioniert.

Ab Version 4.00 sieht die ganze Sache etwas anders aus. Hier gibt es 20 durchnummerierte Funktionen, deren Kennung Sie nur ins C-Register laden muessen. Die noch erforderlichen Aenderungen fuer das vorletzte Beispiel (Laden eines Titelbildes) werden spaeter erklart.

Einige der 20 Funktionen sind offenbar Erweiterungen vorbehalten, denn sie tun nichts weiter, als nur mit RET zum Hauptprogramm zurueckzuspringen. Hier die wichtigsten:

NR.	FUNKTION	PARAMETER
0	Systemeinbindung (Parameter setzen 1 auf werkbestimmung usw.)	keine
5	Sektoren schreiben	keine
6	Sektoren lesen	keine
7	Katalog ausgeben	A-Register mit Ausgabekanal
11	Bytes saveen	HL -Startadresse DE -Laenge
14	Bytes laden	A -Reg.=0 -> Zieladresse unveraendert A -Reg.=1 -> HL muss neue Zieladresse enthalten
18	File loeschen	Filename/ Typ

Das DOS wird mit CALL 15464 eingeschaltet, CALL 15385 bewirkt das Suchen der Funktion, deren Ausfuehrung und das Wegschalten des DOS. Vor Aufruf der Funktionen 11 14 18 muss in den Speicherstellen 23773 bis 23780 der Name des Files und in 23781 sein Typ abgelegt werden.

Man koennte nun auf die Idee kommen, es genuege zum lesen eines Sektors einfach die entsprechende Funktion aufzurufen, nachdem man die notwendigen Parameter uebergeben het. Das ist zwar im Prinzip richtig, funktioniert aber in der Praxis nur mit quaelender Langsamkeit. Daher ist es wichtig, vor dem Aufruf lesen oder Schreiben die Funktion 0 anzuwaehlen. Vermutlich liegt hier auch die Ursache dafuer, dasz bei den mit DOS 4.xx formatierten Disketten in der alten Version die Geschwindigkeit sehr zu wuenschen uebrig laeszt.

An einem praktischen Beispiel soll das Wirken des neuen DOS nun noch einmal gezeigt werden. wir wollen einen Screen in den Bildspeicher laden. Dazu poken wir zunaechst einmal den File-Namen und -Typ in die genannten Speicherzellen und starten dann folgendes Maschinenprogramm:

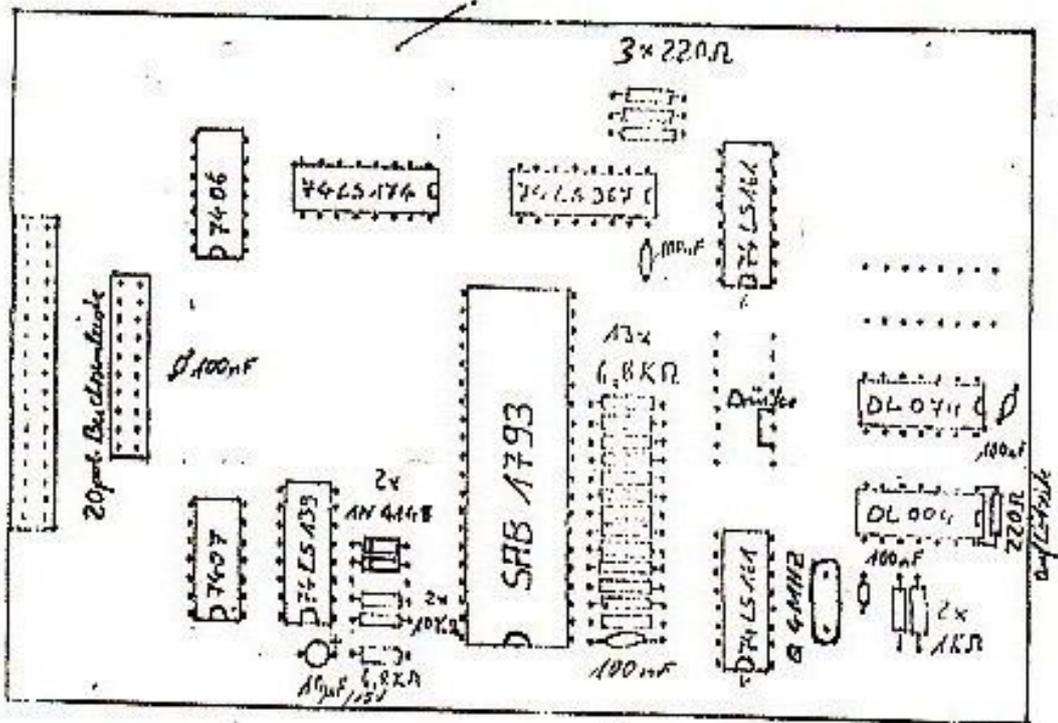
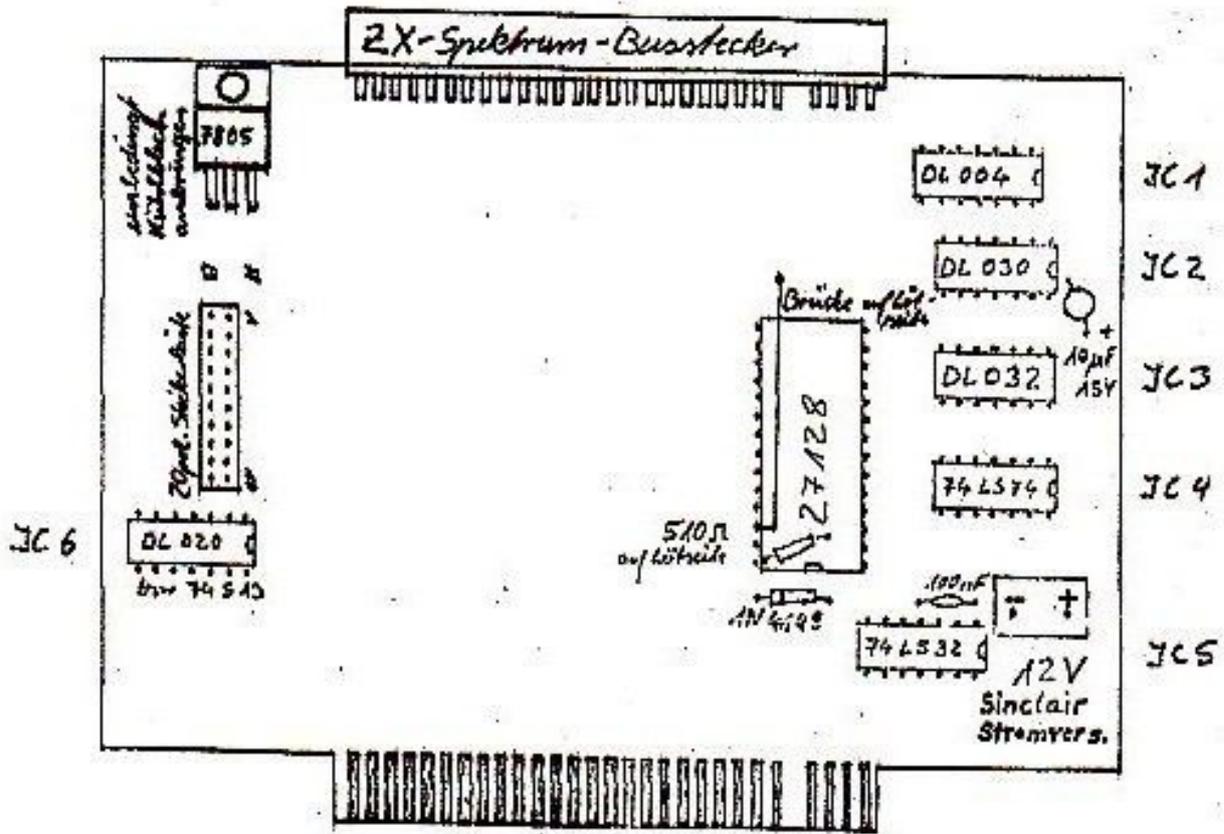
CALL 15464	DOS einschalten
PUSH HL	DOS-Auswahladresse auf den Stapel
LD C,0	Funktion 0 rufen
CALL 14385	Funktion 0 suchen und ausfuehren
LD HL,16384	Zieladresse in HL laden
LD B,27	27 Sektoren lesen (6912 Bytes)
LD D,X	X =Tracknummer in D laden
LD E,Y	Y =Sektornummer in E laden
LD C,5	Funktion 5 waehlen
CALL 14385	Funktion 5 suchen und ausfuehren
RET	zurueck zum Basic bzw. Hauptprogramm

Falls ein File mit einer anderen Zieladresse als auf der Disk abgelegt, geladen werden soll, ist hinter dem PUSH HL noch einzufuegen : LD A,1 (Merker fuer: Achtung neues Ziel);  
LD HL,neue Zieladresse

Laut Bedienungsanleitung ist es recht einfach auftretende Fehler zu erkennen. Die Fehlernummer steht immer in Adresse 23823. Sie muessen nur noch wissen, dasz diese Adresse lediglich im Fall eines Fehlers veraendert wird und sie sinnvollerweise vor irgendwelchen Aktionen auf Null setzen, falls Sie Wert auf eine exakte Fehlermeldung legen. Ihre Experimente, die Sie nun auf Grund all dieser Informationen durchfuehren koennen, sollten Sie nur mit solchen Disketten wagen, auf denen keine wichtigen Daten stehen. Das waere zumindest am Anfang zweckmaeszig.

# Floppy - Controller

## Bestückungsplan



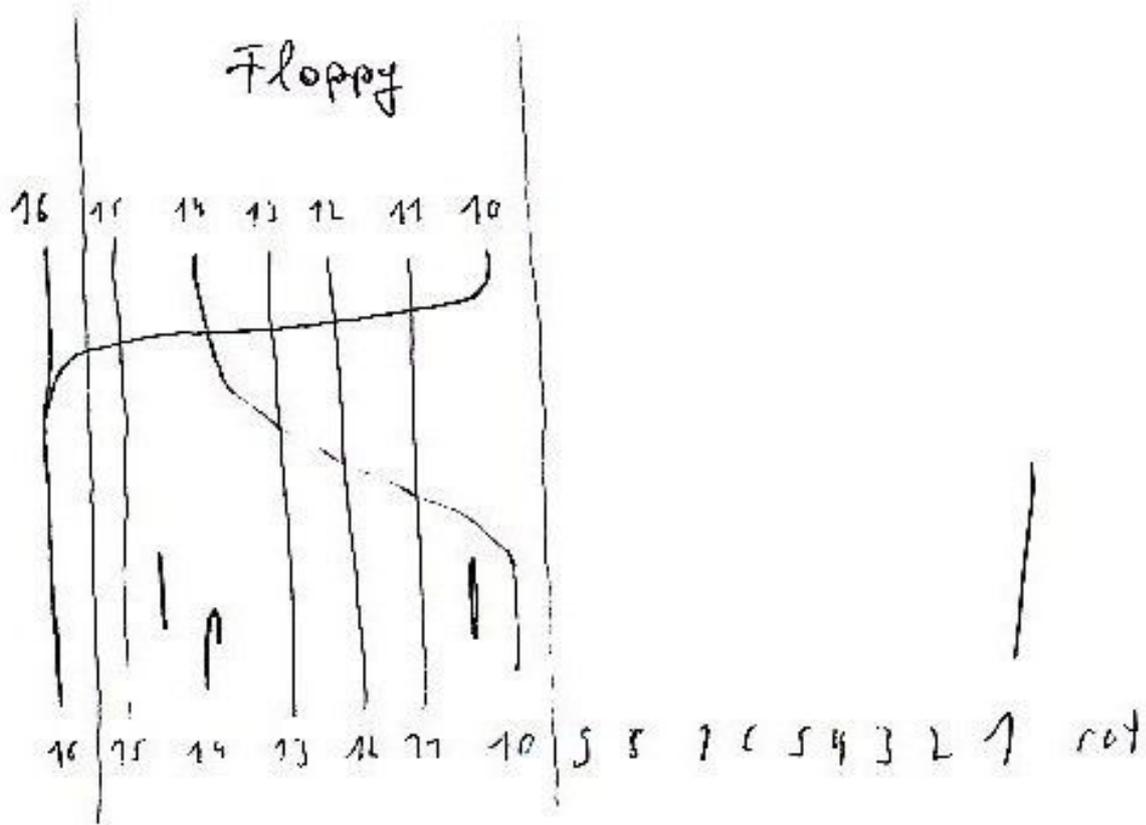




## Shugart-Anschluß

0V - 1	2 -	
0V - 3	4 -	
0V - 5	6 - Select drive D	
0V - 7	8 - Index	9A rt
0V - 9	10 - Select drive A	5B or
0V - 11	12 - Select drive B	
0V - 13	14 - Select drive C	:3A
0V - 15	16 - Load head, Motor on	4B bl
0V - 17	18 - Direction	10B gn
0V - 19	20 - Step	6B br
0V - 21	22 - Write data	8B rs
0V - 23	24 - Write enable	9B gr
0V - 25	26 - Track zero	5A ws
0V - 27	28 - Write protected	6A ge
0V - 29	30 - Read data	8A vt
0V - 31	32 - Side 1	
0V - 33	34 -	
	1A, 11A, 12A, 13A, 1B	Masse blau
	2B, 3B	+ 5V rot
	11B, 12B, 13B	+12V gelb

Sicherung primär ca. 200 mA



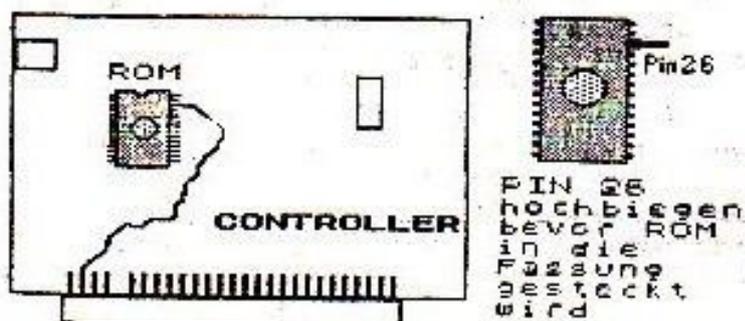
↑  
Beta Interface

Andrew Kabel  
for 3.5" HD Laufwerke

Mike  
1988

### Einbau des ROM in den B-Disk Controller:

Das alte B-Disk Betriebssystem wird herausgezogen, und gegen das beiliegende ROM ausgetauscht. Zu beachten ist, daß PIN 26 vorher hochgebogen wird, da dieser noch mit einer Busleitung verbunden werden muß. Achten Sie darauf, daß die Kerbe des Bausteins, von oben gesehen, nach hinten zeigen muß. Haben Sie den Baustein so eingesetzt, dann löten Sie einen Draht an den hochgebogenen Pin. Das andere Ende des Drahtes wird mit dem oberen Busleistenkontakt 2 verbunden. Nun können Sie den Controller wieder zuschrauben. Hier noch die entsprechenden Anschlußbilder, um die Arbeiten korrekt durchzuführen:



So wird die Steckerleiste  
mit dem ROM verbunden:  
A19 (zweiter PIN von links)