

Februar 1990

Ich habe mich bemüht, diese Übersetzung so exakt wie möglich vorzunehmen! Für eventuell vorhandene Fehler kann ich keine Garantie übernehmen!

Die Gliederung des Original-Handbuchs wurde weitgehend übernommen. Teilweise habe ich kleine Ergänzungen und Anmerkungen vorgenommen, wo es mir sinnvoll erschien.

Auch einige kleinere Unstimmigkeiten im Original-Text wurden so weit sie mir auffielen, korrigiert!

Ich hoffe, mit dieser Übersetzung einigen BETA-Disk-Usern den Umgang mit diesem vielseitigen und meiner Meinung nach sehr leistungsfähigem Disk-System zu erleichtern.

Vielleicht noch kurz eine Beschreibung meines 'Geräteparks'!

SPECTRUM 48 mit dk'tronics-Tastatur
ISO-Face mit Monitor-ROM
BETA-Interface Version 5.03
2 Laufwerke NEC 1036 A, 3.5", 80 Tracks DS
KEMPSTON E-Centronics-Interface
Drucker SHINWA CP-80
Schwarz/Weiß-Monitor
Großes Eigenbau-Netzteil für alle Verbraucher

Daneben gibt's natürlich noch den unverzichtbaren Kassetten- Rekorder, einen EPROMMER und einen Ersatz-Spectrum 48 kB!

Ich wünsche allen SPECTRUM- und BETA-Disk-Usern weiterhin viel Freude und Erfolg!

Mit freundlichen Grüßen

Wilhelm

BETA

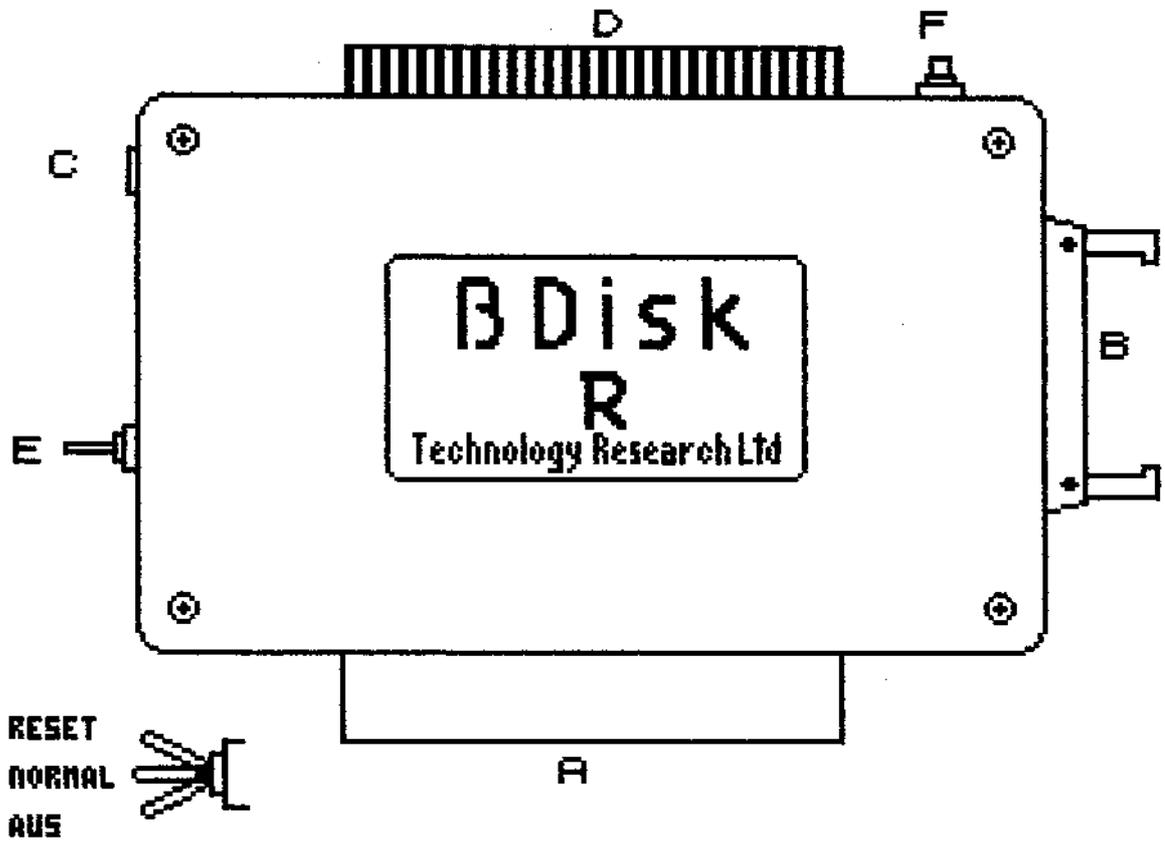
DISK

INTERFACE

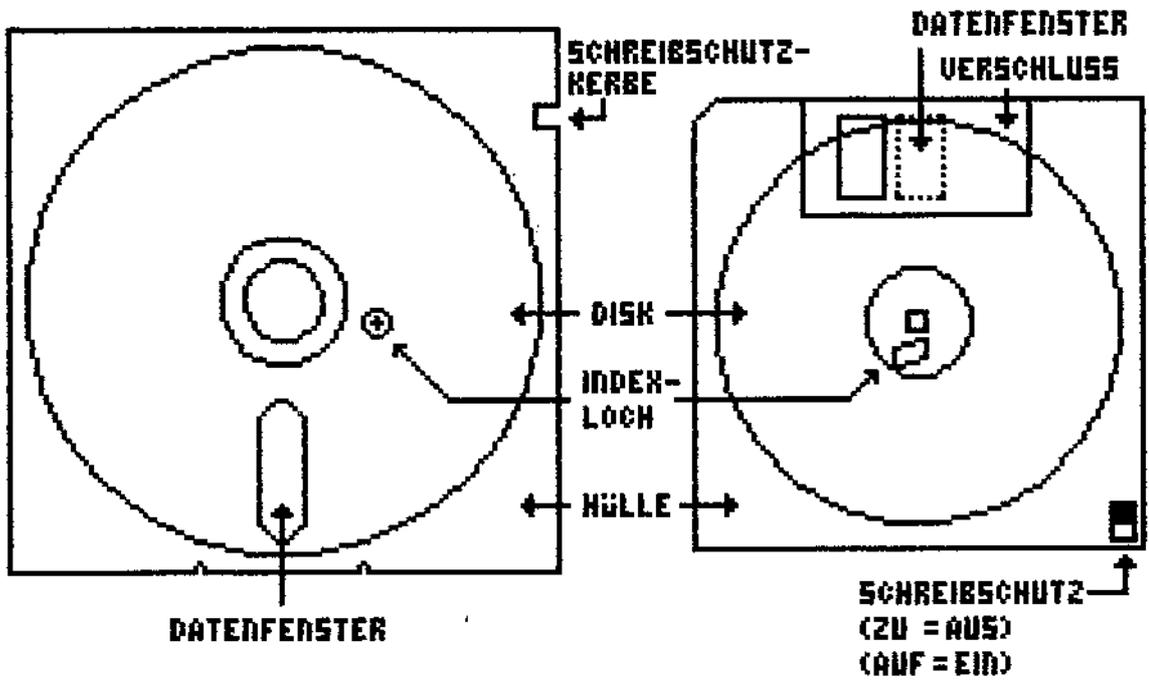
Handbuch für
Version 5.03



© 1986 Technology Research Ltd.



- A = SPECTRUM-Stecker
- B = Laufwerk-Anschluss
- C = Netzteil-Anschluss
- D = Stecker fuer Interfaces
- E = System-Schalter
- F = Magic-Button



Disk 5.25"

Disk 3.5"

BETA 128 DISK-INTERFACE

Sehr geehrter Kunde!

Gratulation zu Ihrer guten Wahl! TECHNOLOGY RESEARCH LTD. ist überzeugt, daß Sie lange Jahre Freude an dem Interface haben werden. Sie werden Ihre Entscheidung nicht bereuen, denn kein anderes Interface für den SPECTRUM bietet sonst die im Folgenden aufgeführten Möglichkeiten:

- * Einfacher und schneller Zugriff: Wesentlich schnellerer Zugriff auf Daten und Programme als bei Kassette gewohnt!
- * Flexibilität bei der Laufwerkswahl: 5.25", 3.5" oder 3", 40 oder 80 Spuren, ein- oder doppelseitige Laufwerke, auch gemischt, sind zulässig!
- * Kapazität: Bis zu 4 Laufwerke können angeschlossen werden. Speicherung in doppelter Dichte. Max. Kapazität bei 4 Laufwerken: 2.5 Millionen Zeichen!
- * Flexibilität: Die Laufwerke sind unabhängig vom Interface an anderen Computern verwendbar!
- * Kompatibilität: Das Interface kann sowohl am SPECTRUM, SPECTRUM PLUS und SPECTRUM 128 arbeiten!
- * Auto-boot: Automatischer Prog.-Start nach Reset oder Einschalten möglich (nicht beim 128 er!).
- * £ Auto-Check: Prüft und stellt selbsttätig die richtige Laufwerksversion ein!
- * Magic-Button: Hiermit lassen sich die meisten Programme per Knopfdruck auf Disk speichern!
- * System-Schalter: 3 Schalterstellungen ermöglichen Ein- und Ausschalten des Interfaces und Reset!
- * Betriebssystem im EPROM: Das 'Disk operating System' (TRDOS) ist in einem EPROM enthalten und es benutzt nur 112 Bytes RAM für Systemvariablen!
- * Syntax: Einfache Befehlseingabe per SPECTRUM-Keywords erlaubt Zugriff auf Disk aus dem BASIC und MC!
- * File-Behandlung: Das TRDOS erlaubt Behandlung numerischer- und String-Arrays, serieller- und random access- (wahlfreier Zugriff) Files!

Diese Beschreibung enthält genaue Informationen darüber, was Sie wissen müssen, um das TECHNOLOGY RESEARCH BETA 128 DISK INTERFACE (im Folgenden kurz Disk-Interface genannt) richtig nutzen zu können!

Alles in diesem Handbuch bezieht sich, bis auf extra genannte Ausnahmen, auf den normalen SPECTRUM 48, SPECTRUM PLUS UND SPECTRUM 128. Beim SPECTRUM 48 und SPECTRUM PLUS (nachfolgend nur noch SPECTRUM & PLUS genannt) gibt es nämlich keine Unterschiede in der Handhabung, wogegen für den SPECTRUM 128 einige Sonderregeln gelten, auf die gegebenenfalls extra hingewiesen wird!

Das Disk-Interface wird 'mit einer 'Utility-Disk' ausgeliefert, die einige Hilfsprogramme enthält (siehe Anhang). Sie ist wahlweise in 5.25" oder 3.5" lieferbar, entsprechend Ihrer Ausrüstung!

Inhaltsverzeichnis :

- 1) **ALLGEMEINES ZUM DISK-SYSTEM**
 - 1.1) Disketten und Laubwerke
 - 1.2) Tracks, Sektoren und Schreibdichte (Density)
- 2) **STARTVORBEREITUNGEN**
 - 2.1) Der richtige Anschluß
 - 2.2) Auto-boot
 - 2.3) Auto-check des Laufwerks
 - 2.4) Die '40'- und '80'-Befehle
 - 2.5) Verwendung weiterer Interfaces
- 3) **DER SYSTEMSCHALTER**
 3. 1) Reset
 - 3.2) Abschalten des Interface
- 4) **BEFEHLS-LISTE**
- 5) **BEFEHLS-SYNTAX**
 - 5.1) Wechsel vom TRDOS zum SOS
 - 5.2) Wechsel vom SOS zum TRDOS
 - 5.3) TRDOS-Aufruf aus dem SOS und BASIC-Programmen
 - 5.4) Wahl eines Hauptlaufwerks
 - 5.5) Vorübergehende Wahl eines Laufwerks
- 6) **FORMATIEREN EINER DISKETTE**
 - 6.1) Einseitiges Formatieren
- 7) **DISKETTEN-KATALOG**
 - 7.1) Katalogausgabe auf dem Bildschirm
 - 7.2) Katalogausgabe auf dem Drucker
- 8) **FILES KOPIEREN**
 - 8.1) Files kopieren zumgleichen Laufwerk
 - 8.2) Einzel-Files und ganze Disks kopieren mit einem Laufwerk
 - 8.3) Kopieren mit mehreren Laufwerken und 'Backups'
- 9) **NAMEN ÄNDERN, LÖSCHEN UND AUFRÄUMEN EINER DISK**
 - 9.1) File-Namen ändern - der NEW-Befehl
 - 9.2) Files löschen - der ERASE-Befehl
 - 9.3) Disk aufräumen - der MOVE-Befehl
- 10) **DER SAVE-, VERIFY-, LOAD-, MERGE- UND RUN-BEFEHL**
 - 10.1) Der SAVE- und VERIFY-Befehl
 - 10.2) Der LOAD- und RUN-Befehl
 - 10.3) Der MERGE-Befehl
- 11) **PROGRAMME VON CASSETTE AUF DISK ÜBERTRAGEN**
 - 11.1) Programm-Anpassung
 - 11.2) Der 'Magic-Button'
- 12) **FILES MIT DATA-FELDERN**
- 13) **SEQUENTIELLE UND RANDOM ACCESS FILES**
 - 13.1) Allgemeine Beschreibung
 - 13.2) Sequentielle Files
 - 13.3) Random access Files
- 14) **DIREKTES SCHREIBEN UND LESEN IN SEKTOREN**
 - 14.1) Der PEEK-Befehl
 - 14.2) Der POKE-Befehl
- 15) **PROGRAMMIERUNG DER TRDOS-BEFEHLE IN MASCHINENSPRACHE**
- 16) **FEHLER-MELDUNGEN**
 - 16.1) Direkt-Meldungen
 - 16.2) Fehlermeldung-Codes
- 17) **WISSENSWERTES ÜBER DAS TRDOS**
- 18) **ANSCHLUSSBELEGUNG DES LAUFWERK-STECKVERBINDERS**
- 19) **ANHANG: HILFSPROGRAMME**
 - 19.1) Programm "tapecopy"
 - 19.2) " "magic"
 - 19.3) " "doctor"

1) ALLGEMEINES ZUM DISK-SYSTEM

Das Arbeiten mit Disketten ist eine professionelle Methode um Daten und Programme von Home- und Personal-Computern zu speichern. Es gibt eine Vielzahl von Vorteilen gegenüber den Verfahren, die Kassetten oder Microdrives benutzen. Es arbeitet sicherer, ist einfacher zu Handhaben und wesentlich schneller!

Im Folgenden werden nun einige Begriffe, die im Zusammenhang mit Disk-Laufwerken und Disketten immer wieder auftauchen, erläutert. Damit soll das Verständnis für die verwendeten Verfahren und eine optimale Nutzung Ihres BETA-Interface erleichtert werden!

1.1) Disketten und Laufwerke

Das TECHNOLOGY RESEARCH BETA-DISK-INTERFACE kann 5.25"-, 3.5"- und auch 3"-Laufwerke ansprechen. Die Disketten werden auch häufig als 'Floppy-Disks', 'Mini'- oder 'Micro-Disk' bezeichnet, wir nennen sie einfach 'Disk'. Gegenwärtig ist die 5.25"-Disk noch sehr populär, der Trend geht aber eindeutig zur handlicheren 3.5"-Disk.

Bild 2 und 3 zeigen beide Disk-Typen. Die 5.25"-Disk besteht aus einer flexiblen Kunststoffscheibe in einer Umhüllung aus Karton oder Kunststoff. Trotz ihrer Flexibilität sollte man sie niemals knicken! Die 3.5"-Disk ist ähnlich aufgebaut, steckt aber wie auch die 3"-Disk in einem stabilen Kunststoffgehäuse und ist dadurch besser geschützt. Sie ist dadurch zur Zeit noch etwas teurer.

Nach dem Einlegen ins Laufwerk rotiert die Disk in ihrer Umhüllung mit ca. 300 Umdrehungen pro Minute. Zum Schutz haben die Disks gewöhnlich einen Verstärkungsring im der Mitte, in den der Transportmechanismus des Laufwerks einrastet.

Beim Kassettenrecorder wird das Band an einem stationären Tonkopf vorbeigezogen. Im Disk-Laufwerk bewegt sich nun nicht nur die Disk am Kopf vorbei, sondern auch der Kopf wird vom Rand zur Mitte oder zurück geführt. Ein Loch (Index-Loch) in der Hülle und ein entsprechendes in der Disk erlaubt dabei eine exakte, schrittweise Steuerung des Kopfes auf die nächste Spur nach Erkennen einer vollen Umdrehung der Disk. Die aktuellen Daten werden vom Kopf im Bereich des langen Schlitzes in der Hülle geschrieben oder gelesen!

Eine Sicherung gegen ungewolltes Löschen wertvoller Daten bietet die 'Schreibschutz-Kerbe', die bei Bedarf überklebt werden kann. Ähnlich den kleinen Öffnungen auf der Rückseite der Kassetten bietet diese Kerbe Schutz vor versehentlichem Überspielen oder Löschen. Bei den 3.5"-Disks besteht der Schreibschutz aus einem kleinen Schieber, der zum Beschreiben mit Daten geschlossen werden muß!

Es gibt 40- und 80-Spur-Laufwerke am Markt, manche davon umschaltbar, damit auch Disketten in beiden Formaten verwendet werden können. Einige besitzen nur einen Schreib/Lesekopf (single sided), diese können dann natürlich auch nur eine Diskseite bearbeiten. Andere mit zwei Köpfen erlauben dagegen die Benutzung beider Seiten der Disk.

Eine goldene Regel besagt, daß immer eine Kopie angelegt werden sollte, bevor Originaldisks auf einem Laufwerk mit abweichender Spezifikation benutzt werden. Ideal ist, wenn eine mit 40 oder 80 Tracks bespielte Disk jeweils auch wieder auf einem entsprechenden Laufwerk verwendet wird.

Trotzdem können Disketten, die mit dem einen Laufwerkstyp formatiert oder beschrieben wurden, eventuell mit einem anderen gelesen werden. Selbstverständlich funktioniert das nur bei Disks gleicher Abmessung! Die folgende Tabelle zeigt, welche Typen miteinander verträglich sind.

Disk-Format:		40Tr SS	40Tr DS	80Tr SS	80Tr DS
Laufwerk:	40Tr SS	C	?	X	X
	40Tr DS	C	C	X	X
	SOTr SS	R	?	C	?
	SOTr DS	R	R	C	C

Erklärung: Tr = Spur Track)
SS = Einseitige Disk (single sided)
DS = Doppelseitige Disk (double sided)
C = Verträglich
R = Verträglich nur beim Lesen
X = Unverträglich
? = Unverträglich, wird von manchen Laufwerken jedoch nicht korrekt erkannt

Ein Beispiel: ein 40 Track DS-Laufwerk kann also eine 40 Track SS-Disk lesen und beschreiben, ein 80 Track DS-Laufwerk kann dagegen nur Lesen von Disks, die mit 40 Tracks formatiert sind!

1.2) Tracks, Sektoren und Schreibdichte (Density)

Im Formatierungs-Prozess (Abschnitt 6) wird durch das TRDOS mit Hilfe des Laufwerks die Diskette auf elektronischem Wege in 40 oder 80 Spuren (entsprechend den Möglichkeiten des Laufwerks) und die Tracks wiederum in Sektoren eingeteilt. Die Anzahl der Sektoren pro Track und die Anzahl der Bytes pro Sektor ist dabei vom jeweiligen Disk-Betriebssystem (DOS) abhängig.

Das TRDOS erlaubt 16 Sektoren pro Track und 256 Bytes pro Sektor. Eine große Anzahl kleiner Sektoren hat mehrere Vorteile:

Erstens, wenn nur eine kleine Datenmenge gespeichert werden soll, wird dadurch nicht unnötig viel Platz auf der Disk belegt. Dies führt dazu, daß eine größere Fileanzahl Platz finden kann!

Zweitens ergibt sich dadurch bei Verwendung von Random Access Files (Files mit wahlfreiem Zugriff) eine größere Programm-Flexibilität und höhere Bearbeitungsgeschwindigkeit.

Um sich auf der Disk zurechtzufinden, benutzt das TRDOS die Spur 0 (die äußerste der Disk) als Inhaltsverzeichnis (Directory)!

Mit Hilfe obiger Angaben läßt sich nun leicht die Sektorenanzahl und Speicherkapazität einer formatierten Disk errechnen. Letztere ist eine wichtigste Angabe bei jedem Disk-System, jedoch sollte sie nur in Zusammenhang mit der Sektorenanzahl und der Sektorgröße gesehen werden. Die folgende Tabelle zeigt die TRDOS-Kapazität in Kilobytes bei unterschiedlichen Laufwerken!

40 Tracks	einseitig	=	39 * 16	=	624	Sektoren * 256	=	156	Kb
40 Tracks	doppelseitig	=	79 * 16	=	1264	Sektoren * 256	=	316	Kb
80 Tracks	einseitig	=	79 * 16	=	1264	Sektoren * 256	=	316	Kb
80 Tracks	doppelseitig	=	159 * 16	=	2544	Sektoren * 256	=	636	Kb

Dies entspricht 4 Kb pro Spur oder 4 Sektoren pro 1 Kb. Die letzte Angabe ist wohl die hilfreichste, wenn der Platzbedarf eines Programms auf Disk berechnet werden soll! Wenn die Programme per CAT- oder LIST-Befehl (Abschnitt 7) angezeigt werden, wird auch immer gleichzeitig die Zahl der noch freien Sektoren angegeben. Diese Zahl durch 4 geteilt, ergibt wieviel Kb auf der Disk noch frei sind.

Bisher wurde die Schreibdichte noch unerwähnt gelassen. Einfache und doppelte Dichte beziehen sich auf das Aufzeichnungsverfahren nach dem die Daten auf der Disk abgelegt werden. TRDOS verwendet 'doppelte Dichte', deshalb können 16 Sektoren mit je 256 Bytes pro Spur untergebracht werden!

2) STARTVORBEREITUNGEN

2.1 Der richtige Anschluß

Nun, wo Sie das Disk-Interface und eins oder mehrere Laufwerke besitzen, lesen Sie bitte zuerst die folgende Anleitung! Beginnen Sie mit dem Zusammenbau erst, wenn Sie alles verstanden haben. Falls Sie weitere Interfaces verwenden, lesen Sie bitte auch Abschnitt 2.5! Vor allen Arbeiten am SPECTRUM-BUS unbedingt die **Spannungsversorgung abschalten!!!**

- 2.1.1 Sehen Sie sich zuerst Bild 1 an und vergleichen Sie die 4 Steckverbinder. Der Stecker (A) sollte direkt auf den SPECTRUM-Bus aufgesetzt werden, sofern kein INTERFACE 1 angeschlossen ist. Ein eventuell vorhandenes IF 1 muß zwischen den SPECTRUM und dem Disk-Interface eingebaut werden. Auf die Steckerleiste (D) werden sonstige Interfaces (für Drucker, Joystick usw.) aufgesteckt!
- 2.1.2 Vorm Anschluß des Disk-Interfaces an den SPECTRUM sollten Sie die Steckerleiste am Rechner durch Abreiben mit einem Alkoholgetränkten Lappen reinigen!
- 2.1.3 Ihr Laufwerk wird gewöhnlich schon mit einem passenden Stecker an seinem Anschlußkabel ausgerüstet sein. Diesen sollten Sie in den Anschluß (B) stecken und verriegeln. Achten Sie darauf, daß die farbig markierte Seite des Flachbandkabels dabei zum SPECTRUM zeigen muß!
- 2.1.4 Stellen Sie die Verbindung zum Netzteil der Laufwerke her und schalten Sie dies ein!
- 2.1.5 Schalten Sie den Fernseher oder Monitor ein!
- 2.1.6 SPECTRUM & PLUS: Systemschalter (E) in Mittelstellung!
SPECTRUM 128 : Systemschalter (E) in 'Aus'-Stellung!
- 2.1.7 Stellen Sie nun die Verbindung zwischen Buchse (C) am Disk-Interface und dem SPECTRUM-Netzteil her!
- 2.1.8 SPECTRUM L PLUS: Auch bei mehr als einem Laufwerk wird nun erstmal nur Laufwerk 'A' aktiviert und es erscheint nach wenigen Sekunden etwa folgendes Bild:

* TR-DOS Ver 5.xx *
(C) 1986 Technology Research Ltd.
(U.K.)
BETA 128

A>K

Dabei enthält 5.xx die Nummer der vorliegenden BETA-Version und A> zeigt, daß das TRDOS eingeschaltet ist?

Manchmal kann es auch vorkommen, das der SPECTRUM beim Einschalten das Disk-Interface erst mal ignoriert, erkennbar daran, daß das obige Einschaltbild nicht erscheint. Dann sollte ein Reset per System-Schalter vorgenommen werden (siehe Abschnitt 3.1)!

SPECTRUM 128 : Der SPECTRUM 128 startet ganz normal. Um ins TRDOS zu gelangen, muß hier

RANDOMIZE USR 15616

eingegeben werden, gefolgt von <ENTER>. Dann erscheint obige Einschaltmeldung!

Nun befindet man sich im TRDOS. Erst nachdem Sie den Rest des Handbuchs gelesen haben, werden Sie alle Befehle und Möglichkeiten voll nutzen können. Deshalb zuerst zum Ausprobieren ein Beispiel, bei der Sie die Utility-Disk verwenden können. Diese ist schreibgeschützt (siehe Abschnitt 1.1) und sollte es auch bleiben!

Legen Sie die Disk ins Laufwerk 'A' ein (und schließen Sie den Einsteckschacht falls erforderlich). Mit dem CAT-Befehl (siehe näheres in Abschnitt 7.1) kann nun der Inhalt der Utility-Disk aufgelistet werden. Geben Sie hinter dem A> dazu folgendes ein und beobachten Sie, was geschieht:

CAT <ENTER>

Beachten Sie, daß es sich bei CAT um das Befehlswort auf Taste 9 (Extended Modus + Shift) handelt, gefolgt von ENTER! Danach sieht der Bildschirm dann etwa wie folgt aus:

```
Title: $Utili
6 File(s)
0 Del. File
A:secread <C>      1:tape5.0 <C>      4
A:doctor <B>      22:tapecopy<B>    1
A:boot <B>        3:magic <B>     12

1221 free
```

WARNUNG!!!

**Niemals ein Interface am SPECTRUM anstecken oder abziehen solange die Spannungsversorgung noch eingeschaltet ist!
Niemals eine Disk im Laufwerk lassen, wenn der SPECTRUM neu eingeschaltet wird. Das kann gelegentlich schon mal zu Datenverlusten oder Fehlern führen!**

2.2 Auto-boot

SPECTRUM 128 : Der SPECTRUM 128 bietet diese Möglichkeit nicht!

Nach dem Einschalten oder einem Reset befindet man sich automatisch im TRDOS (vorausgesetzt, der System-Schalter stand in Mittelstellung!) und Laufwerk 'A' ist aktiviert. Das TRDOS versucht dabei, ein BASIC-Programm mit Namen "boot" zu laden. Falls ein Programm dieses Namens nicht existiert oder gar keine Disk eingelegt war, erscheint anschließend die normale Einschaltmeldung (Siehe Abschnitt 2.1.8).

Wenn während des Einschaltens eine Disk eingelegt war, kann es wie schon erwähnt, eventuell zu Fehlern auf der Disk kommen. Dies tritt am Häufigsten auf bei Laufwerken mit ständig 'geladenem Kopf' (d.h. der Kopf ist permanent auf Schreib/Lese-Position abgesenkt)! Folgendes Vorgehen ist deshalb aus Sicherheitsgründen empfehlenswert:

- 1) SPECTRUM einschalten
- 2) Disk mit BASIC-Programm "boot" in Laufwerk 'A' stecken
- 3) Reset betätigen (siehe Abschnitt 3.15)

2.3 Auto-check des Laufwerks

Beim Einschalten des TRDOS wird bekanntlich automatisch geprüft, was für ein Laufwerk 'A' angeschlossen ist, allerdings nur unter der Voraussetzung, daß Sie die Spannungsversorgung schon vorher oder zugleich mit dem SPECTRUM eingeschaltet haben!

Das TRDOS prüft selbstständig die 'Step rate' (Schritt-Geschwindigkeit) des jeweiligen Laufwerks, wodurch sicher gestellt ist, daß auch die höheren 'Step rates' der modernen Laufwerke voll ausgeschöpft werden können. Ebenso wird natürlich getestet, ob es sich um eine 40- oder 80-Trackversion, einseitig oder doppelseitig handelt.

Diese Tests werden auch vorgenommen, wenn keine Disk im Laufwerk steckt. Ist mehr als ein Laufwerk angeschlossen, werden die zusätzlichen jeweils beim ersten Aufruf geprüft!

2.4 Die *40*- und *80*-Befehle

Einige ältere 40-Spur-Laufwerke besitzen noch keine Endabschaltung und deshalb versagt der beschriebene Selbsttest dabei. Dann glaubt das TRDOS, es liege ein 80-Spur-Laufwerk vor. Deshalb gibt es den '40'-Befehl, der dem System mitteilt, daß das angesprochene Laufwerk 40 Tracks hat. Der Befehl lautet:

40 <ENTER>

Für zwischen 40 und 80 Tracks umschaltbare Laufwerke bietet sich noch die Möglichkeit die jeweils andere Versionen zu wählen, nach der Prüfung durchs TRDOS. Dazu gibt man dann z.B. zum Wechsel von 40 auf 80 Tracks ein:

80 <ENTER>

2.5 Verwendung weiterer Interfaces

2.6

Das Disk-Interface arbeitet grundsätzlich mit der von SINCLAIR vertriebenen Hardware, darunter auch dem INTERFACE 1 und Microdrives zusammen. Es verträgt sich ebenfalls mit der meisten sonst am Markt erhältlichen Hardware, die einfach auf den Busstecker (D) aufgesteckt werden kann.

Es existieren aber auch Interfaces, die ein ROM enthalten, daß den gleichen Speicherplatz beansprucht wie das TRDOS. Falls Sie so etwas zusammen mit dem BETA-Interface betreiben wollen und das Zusatz-Interface nicht abschaltbar ist, dann ist es erforderlich, zuerst das Disk-Interface abzuschalten (System-Schalter dazu in 'Aus'-Stellung)!

3) DER SYSTEMSCHALTER

Dieser befindet sich auf der linken Seite des Disk-Interface (siehe Bild 1) und es handelt sich um einen Schalter mit 3 Stellungen. Normalerweise sollte er sich in Position 2 befinden, Position 1 ist die Reset- und 3 die 'Aus'-Stellung.

SPECTRUM 128 : Systemschalter in Position 3 bringen (= Aus)

Gewöhnlich, wenn man im TRDOS arbeitet, steht der Schalter in Mittelstellung. Wie Sie sich sicher erinnern (siehe Abschnitt 2.1.6) ist diese Position bei SPECTRUM & PLUS notwendig, damit das TRDOS sich einblenden kann.

Um nun vom TRDOS ins SOS (= normales Betriebssystem) zu gelangen, ist kein Betätigen des Schalters notwendig (siehe dazu Abschnitt 5)!

3.1 Reset

SPECTRUM 128 : Die Reset-Stellung hier nicht zum Zurücksetzen des Rechners verwenden, sondern dafür den Reset-Knopf des SPECTRUM 128 drücken!

Die Reset-Möglichkeit wurde ins Disk-Interface integriert, damit sich der Rechner leicht neu starten läßt. Ein Reset hat die gleiche Wirkung, wie ein Aus- und wieder Einschalten Ihres SPECTRUM! Sie brauchen dazu nur den Schalter in Position 1 und anschließend zurück in Stellung 2 bringen.

Der Bildschirm wird danach kurz weiß, im gleichen Augenblick schaltet sich das TRDOS ein und nach wenigen Sekunden sehen Sie die Einschaltmeldung (Abschnitt 2.1.8)!

Nach dem Reset ist RAMTOP (Speicherobergrenze) auf den normalen Wert gesetzt und der Speicher mit allen Variablen gelöscht. So läßt sich also sehr schnell der ganze Speicher von allen Daten aus vorhergehenden Operationen befreien. Er bietet somit eine schnelle Methode, von einem zum anderen Programm zu wechseln, falls die vorhandenen Daten nicht mehr gebraucht werden.

3.2) Abschalten des Interface

SPECTRUM 128 : Hier immer nur die 'Aus'-Position wählen
Wie schon in Abschnitt 2.5 erwähnt, gibt es einige Interfaces, deren ROM den gleichen Adressbereich beansprucht, in dem sich auch das TRDOS einschaltet. Falls es sich nicht Abschalten läßt, soll der System-Schalter am Disk-Interface in Aus-Stellung gebracht werden, **bevor** die Spannung eingeschaltet wird!

4) BEFEHLS-LISTE

Befehl	Beschreibung	Abschnitt
*"a:"	= Laufwerk 'A' zum Hauptlaufwerk machen	5.4
*"b:"	= Laufwerk 'B' zum Hauptlaufwerk machen	5.4
*"c:"	= Laufwerk 'C' zum Hauptlaufwerk machen	5.4
*"d:"	= Laufwerk 'D' zum Hauptlaufwerk machen	5.4
40	= Laufwerk auf 40 Tracks umstellen	2.4
80	= Laufwerk auf 80 Tracks umstellen	2.4
CAT	= Disk-Katalog auf Bildschirm ausgeben	7.1
CAT #x	= Katalog auf Drucker ausgeben (x=Kanalnr)	7.3
LIST	= Detailierten Katalog auf Screen ausgeben	7.2
LIST #x	= Detailierten Katalog auf Drucker (x=Stream)	7.2
OPEN #x	= Seriellen/Random-File öffnen (x=Streamnr)	13
CLOSE #x	= Seriellen/Random-File schließen (x=Streamnr)	13
INPUT #x	= Seriellen oder Random-Access-File einlesen	13
PRINT #x	= Seriellen oder Random-Access-File speichern	13
COPY	= Von einem zum anderen Laufwerk kopieren	8.1
COPY s	= Einzelne Files kopieren bei einem Laufwerk	8.2
COPY b	= Gesamte Disk kopieren bei einem Laufwerk	8.2
ERASE	= File-Namen im Katalog der Disk löschen	9.2
LOAD	= Programm, Code oder Datas von Disk laden	10.2
RUN	= BASIC-Programm von Disk laden und starten	10.2
MERGE	= BASIC-Prog. mit einem von Disk kombinieren	10.3
SAVE	= Prog., Code oder Datas auf Disk abspeichern	10.1
VERIFY	= Gesavetes Programm auf Richtigkeit prüfen	10.1
NOVE	= Entfernen gelöschter Files (= Verdichten)	9.3
NEW	= Vorhandenen File-Namen auf Disk ändern	9.1
PEEK	= Einen bestimmten Sektor ins RAN laden	14.1
POKE	= Daten aus RAM in bestimmten Sektor schreiben	14.2
GOTO	= Per Magie Button gesaveten Code laden/starten	11.2
FORMAT	= Disk im eingestellten Format formatieren	6
RETURN	= Rückkehr aus dem TRDOS ins normale SOS	5.1

Die Tabelle bietet eine Übersicht der möglichen Befehle im TRDOS. Vor Anwendung bitte die betreffenden Abschnitte genau durchlesen.
Die meisten Befehle verwenden normale SPECTRUM-Keywords (SPECTRUM & PLUS). Einige benötigen Ergänzungen, die im jeweiligen Kapitel genauer beschrieben werden!

5) BEFEHLS-SYNTAX

Das TRDOS ist sehr flexibel und Sie können es folgendermaßen ansprechen:

- (a) direkt aus dem TRDOS
- (b) direkt aus dem normalen SPECTRUM-Betriebssystem (SOS)
- (c) aus einem BASIC-Programm
- (d) aus Maschinensprache-Programmen

Wenn immer das TRDOS eingeblendet ist, sehen Sie das Symbol für das jeweils aktivierte Laufwerk und dahinter einen Pfeil. Dies dient als Einschaltzeichen (Prompt) des TRDOS, dazu 2 Beispiele:

A> oder B>

Abhängig von der gerade vorgenommenen Eingabe können natürlich hinter dem Einschaltzeichen TRDOS-Befehle folgen, z.B.:

A> RUN "boot"

TRDOS-Befehle werden immer unmittelbar hinter dem Prompt eingegeben. Falls einmal versehentlich ein falscher Befehl benutzt wurde, kann er mit Hilfe der DELETE-Taste entfernt werden!

5.1 Wechsel vom TRDOS zum SOS

Zurück vom TRDOS ins SOS (SPECTRUM Operating System = normales Betriebssystem) geht's mit der RETURN-Taste. Im folgenden Beispiel dazu sind die in () gesetzten Worte als Erläuterung aufzufassen. Sie gehören also nicht zur Befehlsyntax;

A> (TRDOS-Einschaltmeldung im Bild)
A> RETURN (Bild nach Drücken der Y-Taste)
<C> Sinclair Copyright... (Bild nach Drücken von <ENTER>)

Die ENTER-Taste muß also, wie auch sonst beim SPECTRUM üblich, als Abschluß zur Befehlsausführung jedesmal gedrückt werden. Das gilt natürlich für alle Befehle, die im Handbuch noch erläutert werden, auch wenn nicht mehr extra drauf hingewiesen wird!

5.2 Wechsel vom SOS zum TRDOS

Das TRDOS läßt sich vom normalen Betriebssystem (SOS) aus (im K-Modus!) folgendermaßen einschalten:

RANDOMIZE USR 15616

Dazu wird das Befehlswort <RANDOMIZE> auf der T-, anschließend im E-Modus <USR> auf der L-Taste und zum Schluß die Adresse 15616 eingetippt.

Bei nur einem Laufwerk aktiviert dieser Aufruf immer das A-Laufwerk, andernfalls wird das zuletzt als Hauptlaufwerk festgelegte angesprochen (siehe dazu Abschnitt 5.4).

Auch wenn es nicht direkt zu sehen ist, bleibt ein im Speicher befindliches Programm natürlich erhalten. Davon können Sie sich leicht nach Beendigung der eingegebenen TRDOS-Befehle und Rückkehr ins SOS überzeugen, indem Sie das Programm Listen.

5.3 TRDOS-Aufruf aus dem SOS und BASIC-Programmen

5.4

Die Befehlsstruktur ist eigentlich immer die Gleiche, aber wenn das TRDOS vom SOS oder aus BASIC-Programmen aufgerufen werden soll, ist die Zwischenschaltung eines zusätzlichen REM-Befehls notwendig. Bei Direkteingabe aus dem SOS wird der Befehl auch augenblicklich ausgeführt, die Rückkehr erfolgt dann jedoch im Gegensatz zum Beispiel 5.2 auch ins SOS! Folgendermaßen muß der Befehl z.B. aussehen, wenn der Katalog einer Disk in Laufwerk "B"

gezeigt werden soll:

RANDOMIZE USR 15619: REM: CAT "b:"

Der Gebrauch von Adresse 15619 anstelle von 15616 bewirkt die erwähnte Rückkehr ins normale Betriebssystem!

Außer wenn ein übergeordneter Befehl wie z.B. RUN ausgeführt wurde, meldet sich der SPECTRUM nach Befehlsende also mit dem Einschaltzeichen (bei Befehlen direkt aus dem TRDOS) oder mit dem normalen SINCLAIR-'OK' (bei Befehlen aus dem SOS) zurück. Dazu ein Beispiel:

```
A> CAT <ENTER>          ergibt nach Beendigung Meldung:      A>
RANDOMIZE USR 15619: REM: CAT <ENTER> ergibt Meldung:        OK.
```

Auch wenn eine Befehlseingabe wie beschrieben aus dem SOS möglich ist, so ist diese doch nur bedingt empfehlenswert. Erstens ist die Eingabe aus dem TRDOS sicherer, da z.B. Fehler sofort im Klartext gemeldet werden, zweitens gibt's dann weniger Tipperei und drittens wird dabei auch angezeigt, welches Laufwerk gerade aktiviert ist. Das alles kann manchmal ärgerliche Datenverluste vermeiden helfen.

Falls TRDOS-Befehle in BASIC-Programmen verwendet werden sollen ist auch wieder die folgende Grundstruktur zu verwenden:

9990 RANDOMIZE USR 15619: REM:

Die Zahl 9990 steht hier für eine beliebige Zeilennummer. Zusätzlich ist zu beachten, daß dem Befehl innerhalb der Zeile kein weiterer mehr folgen darf! Das muß in jedem Fall beim Schreiben oder Umarbeiten eines vorhandenen Programms für BETA-Disk beachtet werden. Also immer die Regel beachten:

Der TRDOS-Befehl beendet immer die aktuelle Zeile!!!

Dazu wieder ein Beispiel:
Kassetten-Programmzeile:

```
10 INK 7: PAPER 1: LOAD "" CODE: GOSUB 500: CLS: PRINT "Text"
```

Geändert für BETA-Disk:

```
10 INK 7: PAPER 1: RANDOMIZE USR 15619: REM: LOAD "Title" CODE
20 GOSUB 500: CLS: PRINT "Text"
```

Beachten Sie bitte, das der Programmname bei Diskbetrieb nicht durch "" ersetzt werden darf! Zusätzlich zum TRDOS-Befehl, der Zeile 10 beendet, muß also der exakte Programmname eingefügt werden, unter dem das gewünschte Programm abgespeichert wurde.

5.4 Wahl eines Haupt laufwerks

Mit Hauptlaufwerk ist hier jenes gemeint, auf das alle TRDOS-Befehle, die nicht ausdrücklich eine neue Laufwerksangabe enthalten, automatisch zugreifen! Nach dem Einschalten oder einem Reset ist dies immer Laufwerk 'A'.

Das TRDOS kann mit max. 4 Laufwerken arbeiten. Zum dauerhaften Wechsel auf ein anderes Laufwerk lautet der Befehl:

***"Laufwerk:"**

Dabei wird statt "Laufwerk" der entsprechende Kennbuchstabe A,B,C oder D eingesetzt. Dazu wieder Beispiele:

```
*"B:*          Laufwerk B zum Hauptlaufwerk machen
*"a:*          Laufwerk A wieder zum Hauptlaufwerk machen
```

Es dürfen sowohl Klein- als auch Groß-Buchstaben zur Laufwerkskennzeichnung benutzt werden, obwohl auf dem Bildschirm immer nur Großbuchstaben vor dem Pfeilsymbol erscheinen!

Nach dem Wechsel von z.B. 'A' nach 'B' werden also alle Befehle das Laufwerk 'B' ansprechen, bis zu einer neuerlichen Änderung.

Ist allerdings nur ein Laufwerk angeschlossen, bleibt es bei dem Einschaltzeichen A> und Sie können dann natürlich die Vorteile, die mehrere Laufwerke bieten, nicht nutzen.

5.5 Vorübergehende Wahl eines Laufwerks

Oft ist es wünschenswert, beim Hauptlaufwerk zu bleiben, aber trotzdem vorübergehend mal auf ein anderes Laufwerk zuzugreifen. Dafür wird der jeweilige Befehl einfach mit einem Zusatz versehen, der das gewünschte Laufwerk angibt. Der Zusatz ähnelt dem Befehl zum Hauptlaufwerkswechsel, allerdings wird der "*" weggelassen! Beispiel:

"a:" oder "B:" oder "c:" oder "D:"

Der komplette Befehl dafür aus dem SOS sieht dann z.B. so aus:

RANDOMIZE USR 15619: REM: LOAD "b:Progname"

Damit ist sichergestellt, daß ein Programm "Progname" vom Laufwerk B geladen wird, unabhängig davon welches Laufwerk augenblicklich als Hauptlaufwerk festgelegt ist.

Direkt aus dem TRDOS heraus sieht obiges Beispiel dann so aus:

LOAD "b:Progname"

Auch hier wird vorübergehend Laufwerk B angesprochen, ohne die Hauptlaufwerksdefinition zu verändern.

6) FORMATIEREN EINER DISKETTE

Eine Diskette muß formatiert werden, bevor der Computer damit etwas anfangen kann. Das heißt, die Sektoren jeder Spur müssen geprüft, identifiziert und elektronisch markiert werden mit Hilfe des TRDOS. Erst danach ist das TRDOS in der Lage, die Diskette zu verwalten.

Im TRDOS ist die Formatier-Routine schon enthalten, deshalb braucht dazu keine zusätzliche Software geladen werden. Daher kann auch jederzeit eine Disk formatiert werden, auch wenn sich ein Programm im Speicher befindet!

Die zu formatierende Disk wird ins gewünschte Laufwerk gesteckt und dann das Befehlswort FORMAT (Extended Modus + Symbol Shift + Taste 0) eingegeben, gefolgt von einem bis zu 8 Zeichen langen Disk-Namen in Anführungszeichen. Der Name darf aus Groß- und/oder Kleinbuchstaben sowie Leerzeichen usw. bestehen:

FORMAT "Disk-1"

Drücken Sie <ENTER> und warten Sie einen Moment. Die Zeit, die das TRDOS jetzt zum Formatieren benötigt, ist abhängig vom Laufwerkstyp. Bei doppelseitigen Laufwerken werden automatisch gleich beide Seiten formatiert. Nach Beendigung erscheint folgende Meldung auf dem Bildschirm:

Disk-1
624/624 oder **1264/1264** oder **2544/2544**
A>

Sie zeigt den gewählten Disknamen, gefolgt von der Anzahl der formatierten Sektoren und dahinter die Zahl der maximal möglichen Sektoren beim jeweiligen Diskformat! Ist nun die erste Zahl kleiner als die zweite, so haben Sie eine fehlerhafte Disk erwischt!

Die maximale Sektorenzahl variiert je nach Laufwerksversion. Track 0 wird bekanntlich vom System selbst beansprucht, es bleiben demnach bei 40 Track SS noch 39, bei 40 Track DS bzw. 80 Track SS noch 79 und bei 80 Track DS noch 159 Sektoren verfügbar. Bei 16 Sektoren pro Spur ergibt das demnach als max. Sektoranzahl 624, 1264 oder 2544 (vergleiche Abschnitt 1.2)!

6.1 Einseitiges Formatieren

Möglicherweise wollen Sie eine Disk auch mal nur einseitig formatieren (z.B. für Tauschzwecke). Falls Ihr Laufwerk nur einseitig ausgelegt ist, geht das natürlich ohne Probleme mit dem normalen FORMAT-Befehl!

Verwenden Sie jedoch ein doppelseitiges Laufwerk, muß in diesem Fall das erste Zeichen des Disknamens ein "\$" sein. Beispiel:

FORMAT "\$Disk 2"

Nach Beendigung des Formatierens erscheint dann die Meldung:

```
$Disk 2
624/624 oder 1264/1264 abhängig vom Laufwerk 40/80 Tr. SS
A>
```

7) DISKETTEN-KATALOG

7.1 Katalogausgabe auf dem Bildschirm

Es gibt 2 Befehle, um den Inhalt einer Disk zu zeigen. Der Erste und gebräuchlichste ist der CAT-Befehl. Daneben gibt es noch den LIST-Befehl!

Auf beide Befehle kann man zugreifen, auch wenn sich ein Programm im Speicher befindet. Der CAT-Befehl zeigt den Titel, Typ und die Sektoranzahl der Files und reicht für die meisten Fälle. Der LIST-Befehl zeigt dagegen noch weitere Informationen und ist deshalb für Programmanalysen interessant. Die Befehle dafür sehen so aus:

CAT oder LIST

Anmerkung:

In Einzelfällen, wenn nur noch sehr wenig Platz im Speicher ist, kann es beim LIST-Befehl schon mal zur Fehlermeldung 'Out of memory' kommen. Das hängt in erster Linie mit der Suche nach der Autostartadresse zusammen, die mit angezeigt wird bei BASIC-Programmen. Dafür werden kurzzeitig mind. 512 Bytes zusätzlicher Platz im RAM benötigt! Für einen CAT-Befehl reicht der Platz dann immer noch.

Selbstverständlich kann auch der Katalog eines anderen als dem Hauptlaufwerk abgerufen werden:

CAT "b:" oder LIST "b:"

Auch aus dem SOS oder BASIC-Prog. (mit Zeilennr.) heraus geht's:

```
RANDOMIZE USR 15619: REM: CAT "a:" oder
RANDOMIZE USR 15619: REM: LIST "a:"
```

Beim CAT-Befehl wird dann folgendes gezeigt:

```
Disk-Name
File-Anzahl
Anzahl gelöschter Files
Laufwerk, File-Name, -Typ, -Länge (Anzeige zweispaltig)
Anzahl freier Sektoren
Einschaltzeichen
```

Ein Beispiel hierzu:

```
Title: Disk 1                    Disk-Name
4 File(s)                    4 Programme (Files)
1 Del. File                    1 File gelöscht

A:  Name            <B>        12    B = BASIC-Prog.                    12 Sekt. (3.0    Kb)
A:  Name1           <C>        6     C = MC-Prog.                        6    "        (1.5    Kb)
A:  Name2           <#>       13    # = Seriell/Random                13    "        (3.25   Kb)
A:  Name3           <D>        7     D = Data-Feld                       7    "        (1.75   Kb)
2503 Free                    2544-38=2506, gelöschter File soll hier
                                 3 Sekt. belegen, daher 2503 Files frei!
A>
```

Wenn mehr als 30 Files gezeigt werden sollen, kommt die Frage "scroll?". Ein Tastendruck bringt dann die übrigen Files ins Bild. BREAK oder Taste N unterbrechen die Ausgabe!

Nun ein Beispiel dafür, wie das Bild nach LIST aussieht:

```
Title: Disk 2 Disk Drive: B
4 File(s)      80 Track D. Side
1 Del. File(s) Free Sector 2480

File Name      Start  Lenght Line
Name1 <B>      5 00298 01200 25
Name2 <C>     32 32768 08000
Name3 <#>      7 01780 01780
Name4 <D>      8 30000 32000
```

A>

Hier werden nun alle wichtigen Details für die einzelnen Files auf einer Disk gezeigt, z.B. Diskname, daß die Disk in einem doppelseitigen Laufwerk B mit 80 Tracks steckte. Weiter ist zu sehen, daß 4 Files auf der Disk sind, außerdem ein gelöschtter und es ist noch Platz für 2480 Sektoren (ca. 620 Kb)!

Zusätzlich zu den vom CAT-Befehl gebotenen Angaben zum File-Namen, -Typ und -Anzahl werden bei LIST unter der Rubrik 'Start' bei BASIC-Prog. die Länge inklusive Variablen, bei CODE-Prog. die Startadresse angeführt. Unter 'Lenght' findet man dagegen bei Programmen des Typs (BASIC) die Länge ohne Variablen und bei Typ <C> die Länge des CODE-Teils. Handelt es sich um ein BASIC-Programm mit Autostart, wird auch noch deren Zeilennummer angezeigt!

7.2 Katalogausgabe auf dem Drucker

Manchmal hätte man gern das Inhaltsverzeichnis einer Disk in schriftlicher Form, um sich unabhängig vom SPECTRUM schon mal einen Überblick zu verschaffen. Abschreiben vom Bildschirm ist wohl etwas umständlich und fehlerträchtig! Deshalb ermöglicht das TRDOS auch den Katalogausdruck, falls Sie einen Drucker angeschlossen haben.

Das folgende Beispiel zeigt, wie bei einem Ausdruck über das INTERFACE 1 (Serielle Schnittstelle) zu verfahren ist. Das normale Öffnen des Streams muß natürlich aus dem SOS heraus vorher erledigt werden. Dazu muß, falls das TRDOS gerade eingeblendet ist, zuerst mit RETURN ins normale Betriebssystem (SOS) umgeschaltet werden. Danach kann dann, der normalen IF 1-Syntax folgend, eingegeben werden:

```
FORMAT "T"; 9600: OPEN $4; "T"
```

Fall Sie das KEMPSTON E-Interface (mit Centronics-Schnittstelle) verwenden, geben Sie aus dem SOS statt obiger Befehle folgendes ein (bei anderen Interfaces bitte im Handbuch nachsehen):

```
COPY: REM /1 (=Line feed on / carriage return on)
```

Anschließend geht man, um sich unnötig Tipperei zu ersparen, am Besten wieder ins TRDOS (mit 'RANDOMIZE USR 15616'). Nun kann man den CAT- oder LIST-Befehl wie gewohnt benutzen, eine entsprechende Stream-Angabe bringt den Katalog jedoch nun auf's Papier. Zuerst ein Beispiel für's INTERFACE 1:

```
CAT #4          Katalogausgabe über Stream 4 zum Drucker
List #4, "b:"  Katalogausgabe von Laufwerk B zum Drucker
```

Die entsprechenden Befehle beim KEMPSTON E-Interface lauten:

```
CAT #3 bzw LIST #3, "b:"
```

Beim SPECTRUM können Streams von 0 bis 15 benutzt werden, wobei allerdings Stream 0 bis 3 vom System selbst schon benutzt werden, über welchen Stream zwischen 4-15 die Ausgabe bei Ihrem Interface auch geschieht, ist in der Regel gleich. Nur es sollte der geöffnete (Stream dann auch bei den CAT #- oder LIST #-Befehlen verwendet werden.

Beim KEMPSTON E-Interface wird jedoch der vom System für den Printer festgelegte Stream 3 angesprochen. Der Stream 2 ist übrigens für den Bildschirm zuständig und wird automatisch im Normalfal 1 benutzt!

8.2 Einzel-Files und ganze Disks kopieren mit einem Laufwerk

Bei nur einem Laufwerk ist der normale COPY-Befehl nicht zu gebrauchen. Deshalb wurden die Befehle 'COPY s' und 'COPY b' bereitgestellt.

Mit 'COPY s' kann ein File von einer Disk auf eine zweite mit dem gleichen Laufwerk überspielt werden. Dagegen ist 'COPY b' zum Kopieren einer kompletten Disk mit nur einem Laufwerk gedacht (Backup). Zuerst Beispiele für 'COPY s'

**COPY s "Tapecopy" oder
COPY s "Doctor" CODE**

Beides sollen auf der Originaldisk existierende Files sein. Beachten Sie den Unterschied zu vorhin, wo der neue Name zuerst eingegeben werden mußte. Nun teilen wir dem TRDOS nur den existierenden Namen des zu kopierenden Files in Anführungsstrichen mit unmittelbar hinter dem COPY 's'.

Nach dem abschließenden <ENTER> erscheint zur Sicherheit die Meldung: 'Insert Source disk, then press Y'. Also Originaldisk einlegen, falls noch nicht geschehen und Y-Taste gedrückt! Der File wird nun eingelesen, danach kommt die Aufforderung: 'Insert Destination disk, then press Y'. Neue Disk einlegen, Y-Taste gedrückt und schon ist die Kopie beendet!

Kopien sollte man grundsätzlich von allen wichtigen Programmen anlegen, was allerdings bei Kassettenbetrieb immer eine langwierige Prozedur ist, mit Kopieren, Zurückspulen und Verifizieren.

Mit dem TRDOS und Disks dauert das alles nur einen kurzen Augenblick. Auch ein Überprüfen der Kopien ist blitzschnell erledigt. Darum wird man auch nicht mehr vorm Herstellen von Sicherheitskopien wichtiger Disketten zurückschrecken, denn empfehlenswert ist das Anlegen solcher Backup's allemal!

Der 'COPY b'-Befehl bietet nun eine Erweiterung des 'COPY s' und ist dafür bestens geeignet. Dazu wird nichts weiter als

COPY b

einggegeben und mit <ENTER> gestartet. Danach erscheinen wieder die schon beschriebenen Meldungen. Nur wird diesmal nicht nur ein File geladen, sondern der Reihe nach alle, bis die ganze Disk kopiert ist. Dazu ist zwangsläufig in den meisten Fällen ein mehrmaliges Wechseln von Original- und Ziel-Disk erforderlich, wenn die Aufforderung dazu erscheint.

8.3 Kopieren mit mehreren Laufwerken und 'Backups'

Kopieren auf ein zweites Laufwerk bedeutet zwangsläufig auch kopieren auf eine andere Diskette. Obwohl der Datenaustausch natürlich zwischen allen der 4 möglichen Laufwerke funktioniert, werden wir in den Beispielen nur A und B benutzen! Die Befehlsstruktur ist schließlich unabhängig von anderen Kombinationen.

Geben Sie also den Befehl COPY, gefolgt von den beiden Namen in Anführungsstrichen ein, diesmal allerdings angeführt vom Zusatzbefehl für vorübergehenden Laufwerkswechsel! Da auf zwei verschiedene Disks kopiert wird, dürfen eventuell auch beide Namen gleich sein. Beispiel:

COPY "A:Tasword", "b:Tasword"

Obwohl A das Hauptlaufwerk ist, kopieren wir hier von Laufwerk B nach A bei Verwendung des alten Namens, der allerdings auf der Disk in A noch nicht vertreten sein darf. Falls die Disks in umgekehrter Reihenfolge eingelegt wurden, lautet der Befehl:

COPY "B:Tasword", -A:"Tasword"

Damit wird das Programm dann von A nach B kopiert!

Das Anlegen von Sicherheitskopien (Backup) wird mit zwei Laufwerken stark vereinfacht. Dafür die zu kopierende Disk einfach ins eine, die empfangende (leer und formatiert oder auch teilweise bespielt) in jedes beliebige andere Laufwerk stecken!

Hier arbeiten wir z.B. mit Laufwerk A als Quelle und B als Empfänger. Die Befehlsstruktur entspricht weitgehend derjenigen beim Einzel-File kopieren. Allerdings ersetzt ein "*" die Namen (im Computer-Jargon 'Wildcard' genannt).

COPY "B:*","A:*"

Nach Eingabe erledigt das TRDOS den Rest! Vorausgesetzt, daß ein Name nicht schon auf beiden Disks gleichzeitig existiert und genügend Platz auf der empfangenden Disk ist, endet die Operation mit dem bekannten 'A>'-Prompt.

Ist jedoch ein Name auf beiden Disks schon vertreten, erscheint die Meldung

"Overwrite existing file? Y/N".

Bei Druck auf die Y-Taste wird folglich der alte Name gelöscht und das neue Programm überspielt, mit Taste N dagegen das alte erhalten und das Kopieren beim nächsten File fortgesetzt! Somit hat man auch die Möglichkeit, versehentlich doppelt vorkommende Namen verschiedener Files später zu ändern und erst dann zu kopieren.

Der größte Vorteil des 'Wildcard-Kopierens' liegt in der im Vergleich zum Einzel-File überspielen, wesentlich höheren Geschwindigkeit der gesamten Operation.

9)

NAMEN ÄNDERN, LÖSCHEN UND AUFRÄUMEN EINER DISK

9.1 File-Namen ändern - der NEW-Befehl

Die Möglichkeit, einen Namen auf der Disk bei jedem beliebigen File zu ändern, ist eine der flexibelsten der zur Verfügung stehenden Befehle. Namensänderungen sind nahezu unverzichtbar im Laufe von Programmentwicklungen, außerdem lassen sich damit Tippfehler (auch die soll's geben!) korrigieren.

Im Gegensatz zur Angabe im englischen Originaltext muß sich die zu behandelnde Disk nicht unbedingt in Laufwerk A befinden und der NEW-Befehl kann auch ohne weiteres aus einem BASIC-Programm heraus verwendet werden. Beim Zugriff auf ein anderes als das A-Laufwerk muß jedoch wie üblich vorm neuen und alten Filenamen der jeweilige Laufwerkscode angegeben werden! Beispiel:

10 RANDOMIZE USR 15619: REM: NEW "b:Text-neu","b:Text-alt" CODE

Beim Aufruf aus dem TRDOS sonst Disk ins eingestellte Hauptlaufwerk schieben und das SPECTRUM-Befehlswort NEW (die A-Taste) eingeben. Vorher kann man sich natürlich per CAT-Aufruf den zu ändernden Namen, ansehen. Dem NEW muß nun in Anführungszeichen wieder der neue Name und durch Komma getrennt davon ebenso der alte eingegeben werden. Auch dazu ein Beispiel:

NEW "boot","prog"

Nach Abschluß mit <ENTER> erscheint nach kurzer Zeit wieder die Einschaltmeldung als wäre nichts geschehen. Erst ein erneuter CAT-Befehl bringt die Änderung des Namens von "prog" nach "boot" ans Tageslicht!

9.2 Files löschen - der ERASE-Befehl

Wird ein File nicht mehr benötigt, kann er gelöscht werden. Das entsprechende Befehlswort ERASE ist im E-Modus, Symbol Shift, Taste 7 erreichbar. Beispiel:

**ERASE "Oldprog"
ERASE "Liste-1" DATA**

Wie auch beim eben beschriebenen Befehl ist erst nach einem CAT- (oder natürlich auch LIST-) Aufruf zu bemerken, daß der Name gelöscht wurde. Gleichzeitig wird im Kopf des Katalogs die Zahl der gelöschten Files um 1 erhöht, wenn es nicht der letzte auf der Disk gespeicherte File war.

9.3 Disk aufräumen - der MOVE-Befehl

Wenn ein File gelöscht würde, ist das gelöschte Programm noch nicht wirklich von der Disk verschwunden. *Es wurde nur das erste Zeichen des Namens im Katalog mit dem Code 1 (oder 0 falls es sich um das jeweils letzte File auf der Disk handelte) überschrieben.* Danach läßt sich ein so markiertes File zwar nicht mehr listen und aufrufen, aber es belegt noch den ursprünglichen Platz!

Deshalb müssen die davon belegten Sektoren erst wieder für neue Files nutzbar gemacht werden. Um diese durchs Löschen 'verlorenen' Sektoren wieder zurückzugewinnen, verwendet man den MOVE-Befehl. Dies SPECTRUM-Befehlswort ist im E-Modus durch gleichzeitiges Drücken der Symbol-Shift-Taste und Taste 6 zugänglich! Beispiel:

MOVE oder MOVE "b:"

Obwohl das zweite Beispiel natürlich möglich ist, sollte man beim Arbeiten mit MOVE besser vorher das verwendete Laufwerk als Hauptlaufwerk einstellen. Dieser Befehl wird vorzugsweise aus dem TRDOS heraus aufgerufen (*eine Verwendung aus einem Programm heraus ist aber auch möglich*), nachdem das TRDOS eingeschaltet und vielleicht noch eine Überprüfung der Disk per CAT stattfand und möglicherweise noch Files mit ERASE entfernt wurden, MOVE eingegeben und damit die belegten Sektoren wieder verfügbar gemacht.

Der MOVE-Befehl erledigt alles: das Reorganisieren der Diskette und ihres Katalogs. Nach Beendigung erscheint dann wieder die bekannte Einschaltmeldung des TRDOS. Ein anschließender CAT-Befehl zeigt den Erfolg, denn die Zahl der gelöschten Files wird nun mit '0' angegeben und die Anzahl der freien Sektoren ist gestiegen.

Wie alle BETA-Disk-Befehle ist auch der MOVE-Befehl sehr schnell, trotzdem dauert die Ausführung, je nach Anzahl und Länge der als gelöscht markierten Files und auch abhängig von ihrer Lage auf der Disk, unterschiedlich lange.

Um die Wartezeiten zu verkürzen, ist es ratsam, schon zwischendurch nach einigen Löschungen mal per MOVE die Disk aufzuräumen! Das hat auch den Vorteil, daß die noch zur Verfügung stehenden Sektoren der Disk immer exakt angezeigt werden und man, falls der Platz beim Speichern mal knapp wird, nicht erst unterbrechen muß zum Aufräumen!

Wichtige Anmerkung:

Der MOVE-Befehl läßt sich nur verwenden, wenn noch mindestens 4096 Bytes freier Speicher im SPECTRUM zur Verfügung steht, sonst erscheint Fehlermeldung 'Out of memory', denn diesen Platz braucht der SPECTRUM als Zwischenspeicher beim Aufräumen. Dann hilft nur ein Löschen des Speichers und erneuter MOVE-Aufruf!

10) DER SAVE-, VERIFY-, LOAD-, MERGE- UND RUN-BEFEHL

Die Syntax all dieser Befehle ist sich ähnlich. Wie Sie sehen werden, haben sie auch eine enge Verwandtschaft mit den entsprechenden Kassetten-Befehlen!

Alle diese Befehle behandeln Files. Ein File auf der Disk kann sowohl ein BASIC- oder Maschinen-Code-Programm, als auch ein DATA-Feld oder ein Serieller-/Random access-File sein. Das Laden und Speichern von DATA-Feldern wird in Abschnitt 12 behandelt und die Seriellen-/Random access-Files in Abschnitt 13. Hier wird nun zuerst die Vorgehensweise bei BASIC- und CODE-Programmen beschrieben!

Während der Abarbeitung aller 5 Befehle (SAVE, VERIFY, LOAD, RUN, MERGE) kann jederzeit mit mit <BREAK> abgebrochen werden. Falls bei Befehlsaufrufen keine Disk im Laufwerk steckt, wird die Fehlermeldung "No disk" ausgegeben (nur bei Aufruf aus dem TRDOS) und abgebrochen.

Ähnlich läuft es ab, wenn der gewünschte File auf der Disk nicht existiert. Dann wird bei Aufruf aus dem TRDOS mit "No file(s)" die Suche beendet.

10.1 Der SAVE- und VERIFY-Befehl

Mit Hilfe des SAVE-Befehls lassen sich im Speicher des SPECTRUM befindliche Programme auf Disk bringen. Das Programm muß dazu einen Namen bekommen, der in Anführungsstriche gesetzt wird.

Bei einem BASIC-Programm braucht kein Typ angegeben werden, allerdings kann man eine Zeilennummer hinter dem Befehlswort LINE zur Realisierung eines Auto-Starts anfügen! Ist keine Zeilennummer angegeben, dann wird das entsprechende Programm immer bei Zeile 1 gestartet. Beispiele dazu:

```
SAVE "Tasword" LINE 15      Auto-Start erfolgt bei Zeile 15!  
SAVE "Grafedit" LINE"      " " " " 1!  
SAVE "b:Design"           Kein Auto-Start bei LOAD!
```

Für ein Maschinen-Code-Programm ist dagegen die Angabe des File-Typs erforderlich, außerdem wird noch die Startadresse und die zu speichernde Byteanzahl benötigt! Auch dafür Beispiele:

```
SAVE "Discount"CODE 47800,955  
SAVE "Text3"CODE 32000,4000
```

Mit dem VERIFY-Befehl läßt sich vergleichen, ob der gespeicherte File auf der Disk exakt mit dem Speicherinhalt übereinstimmt. Er kann sowohl bei BASIC, CODE- und DATA-Files verwendet werden. Beispiele dafür:

```
VERIFY "b:Design"  
VERIFY "Discount"CODE 47800,955  
VERIFY "Money"DATA m()
```

Ist keine genaue Übereinstimmung gefunden worden, erscheint die Fehlermeldung 'Verify Error' auf dem Bildschirm!

10.2 Der LOAD- und RUN-Befehl

Wurde ein BASIC-Programm mit Zeilennummerangabe für Auto-Start wie z.B.:

```
SAVE "Interest" LINE 25
```

abgespeichert, dann wird immer automatisch gestartet, unabhängig davon, ob es mit LOAD oder RUN geladen wird!

War dagegen bei dem BASIC-Programm kein Auto-Start programmiert, dann wird es mit LOAD nur geladen und kann anschließend z.B. gelistet werden, während RUN lädt und das eigentlich nicht selbststartende Programm bei Zeile 1 startet. Beispiele:

```
LOAD "Interest"           Programm wird nur geladen, kein Start!  
RUN "Design"             Programm wird geladen, Start bei Zeile 1!
```

Wie sicherlich erwartet, gibt das TRDOS auch eine Fehlermeldung aus (Out of memory), falls mal zu wenig Platz für das gewünschte Programm im Speicher vorhanden sein sollte! Das kann z.B. vorkommen, wenn für RAMTOP ein zu niedriger Wert angegeben wurde.

Geben Sie nur LOAD oder RUN ohne zusätzlichen File-Namen ein, versucht das TRDOS, ein BASIC-Programm mit dem Namen "boot" zu laden und zu starten!

Wie in den obigen Beispielen zu sehen, ist die Syntax die gleiche, wie im SPECTRUM-BASIC. So kann man auch hier ein CODE-Programm einfach an die gleiche Adresse laden, die beim Speichern angegeben wurde, indem man eingibt:

```
LOAD "Grafik2"CODE
```

Natürlich kann ein CODE-Programm ebenso an eine andere als die ursprünglich festgelegte Adresse geladen werden, indem man z.B. eintippt:

```
LOAD "Margin-CODE 51000
```

Soll ein CODE-Programm mit RUN gestartet werden, dann muß allerdings die Auto-Startadresse der Startadresse des Programms entsprechen. Auch dazu wieder ein Beispiel:

RUN -Discount"CODE 47800

Vorausgesetzt, der Code ist überhaupt lauffähig ab Adr. 47800, dann wird er nach dem Laden sofort aufgerufen und ausgeführt!

Eine Alternative ist die Verwendung eines mehrzeiligen Laders, der z.B. mit

```
SAVE "Dcload" LINE 10
```

auf Disk gebracht wird:

```
10 CLEAR 47799: RANDOMIZE USR 15619: REM: LOAD "Discount"CODE 47800  
20 RANDOMIZE USR 47838
```

Zum Programmstart, wird einfach **RUN "Dcload"** eingegeben, dann wird zuerst das kurze BASIC-Programm geladen und gestartet, wodurch der CODE-Teil nachgeladen und in Zeile 20 aufgerufen wird bei Adresse 47838! Dieser Umweg ist also immer erforderlich, wenn die Aufrufadresse (hier 47838) des MC-Programms von der Anfangsadresse (hier 47800) abweicht! Mehrere Zeilen sind leider erforderlich, weil der TRDOS-Befehl bekanntlich immer der letzte in einer Zeile sein muß!

10.3 Der MERGE-Befehl

Das TRDOS-MERGE entspricht exakt dem MERGE-Befehl im SOS. Es wird auch mit den gleichen Tasten (E-Modus, Symbol Shift, T) aufgerufen. Der MERGE-Befehl fügt zu einem schon im Speicher befindlichem BASIC-Programm ein weiteres von der Disk zu, wobei doppelt vorkommende Zeilennummern überschrieben und neue eingefügt werden. Beispiele für die MERGE-Syntax:

```
MERGE "Subrout"  
MERGE "Funk2"
```

11) PROGRAMME VON KASSETTE AUF DISK ÜBERTRAGEN

Wenn Sie ein für Kassette selbst geschriebenes Programm auf Disk übertragen wollen, dürfte das keine Probleme bereiten. Sie laden es ganz normal aus dem SOS heraus von Kassette und unterbrechen es gegebenenfalls mit <BREAK> oder <STOP>. Ist es erst einmal im Speicher, können Sie es z.B. per Direkteingabe auf Disk bringen:

```
RANDOMIZE USR 15619: REM: SAVE "xxx"      oder  
RANDOMIZE USR 15619: REM: SAVE "xyz"CODE ssss,1111
```

Viele Programme lassen sich auch mit dem auf der Utility-Disk befindlichen Hilfsprogramms "tapecopy" kopieren (siehe Anhang)!

Natürlich läßt es sich nicht vermeiden, daß die zu kopierenden Programme manchmal etwas umgeschrieben werden müssen, damit sie unter TRDOS lauffähig werden! In schweren Fällen, oder wo eine Anpassung unmöglich ist, hilft oft auch der Magie Button weiter!

11.1 Programm-Anpassung

Sehr häufig bestehen Programme aus einer Mischung von BASIC- und CODE-Teilen, um z.B. die Abarbeitungsgeschwindigkeit zu erhöhen. Der BASIC-Teil kann ein simpler CODE-Lader sein aber auch ein komplexer Mittler zwischen dem MC-Teil und dem Benutzer. In diesen Fällen ist die Anpassung gewöhnlich recht einfach. Andere Programme erfordern unter Umständen intensives 'hacken', um sie unter TRDOS zum Laufen zu bringen! Wird's sehr schwierig, muß man als letzte Rettung den erwähnten Magie Button benutzen.

Normalerweise muß man Zugang zu allen SAVE- und LOAD-Eefehlen des BASIC-Programms haben, wenn es konvertiert werden soll. Deshalb ist der erste Schritt dazu, das Programm zu listen!

Die anzuwendende Technik ist zwangsläufig vom jeweiligen Programm abhängig. Vielfach sind nur wenige LOAD- und SAVE-Befehle vorhanden, sodaß nach schneller Durchsicht die zu ändernden Befehle lokalisiert werden können. Manche Programme benutzen jedoch mehrere SAVE- und LOAD-Möglichkeiten, dann sucht man es am Besten zeilenweise nach den entsprechenden Befehlen durch und fügt davor das

```
RANDOMIZE USR 15619: REM: ein!
```

Wir müssen aber auch noch zwei weitere Punkte beachten! Erstens muß der TRDOS-Befehl bekanntlich der letzte innerhalb der Programmzeile sein und zweitens muß eventuell das gewünschte Laufwerk angegeben werden. Der erste Punkt macht manchmal leichte Änderungen der Zeilen-Nummerierung und des -Inhalts notwendig. Das folgende Beispiel verdeutlicht dies:

Existierendes Programm (Kassetten-Version):

```
500 IF X=5 THEN INPUT "Name "; NS: SAVE N$ DATA C$ ( ) : VERIFY N$ DATA C$ ( ) :  
GOSUB 700  
505 IF X=V THEN GOSUB 800: INPUT "Titel ";T$: SAVE T$: SAVE T$ CODE 50000,575  
506..... Rest des Programms
```

Angenommen, die Files sollen mit Laufwerk 'A' gespeichert werden, dann sieht die Anpassung folgendermaßen aus:

An gepaßtes Programm (Disk-Version) :

```
500 IF X=5 THEN INPUT "Name ";N$: RANDOMIZE USR 15619: REM: SAVE N$ DATA C$ ( )  
502 GOSUB 700  
  
504 IF X=V THEN GOSUB 800: INPUT "Titel "|TS: RANDOMIZE USR 15619: REM: SAVE T$  
  
505 RANDOMIZE USR 15619: REM: SAVE T$ CODE 50000,575  
506 ..... Rest des Programms
```

Die Änderungen der Zeilennummern variieren natürlich von Programm zu Programm. Hier waren die nicht benutzten Zeilen 502 und 504 geeignet, die erforderlichen TRDOS-Befehle in separate Zeilen einzubauen. Gelegentlich kann es auch notwendig werden, durch Änderung eigentlich nicht direkt von TRDOS-Befehlen betroffener Zeilennummern, den benötigten Platz zu schaffen! Allerdings muß man immer den gesamten Programmablauf im Auge behalten, denn es ist nicht ungewöhnlich, daß die nun verwendete Zeilennummer aus einem anderen Teil des Programms angesprungen wird.

Zusammengefaßt gelten also folgende Regeln für Anpassungen!

- (1) TRDOS-Befehle müssen stets als letzte in einer Zeile stehen!
- (2) Nur ein 'RANDOMIZE USR 15619: REM:' muß allen existierenden SAVE- und LOAD-Befehlen (oder anderen TRDOS-Befehlen) vorangestellt werden!
- (3) Wird mehr als ein Laufwerk benutzt, muß sichergestellt werden, daß das richtige angesprochen wird!
- (4) Alle Files müssen einen Namen haben!
- (5) Prüfen Sie den Programmablauf sorgfältig, wenn Zeilennummern geändert oder zugefügt werden müssen!

11.2 Der 'Magic-Button'

Der 'Magic-Button' (magischer Knopf) ist auf der Rückseite des BETA-Disk-Interface angebracht (siehe Bild 1). Sein Zweck besteht darin, das Speichern von Kassetten-Programmen auf Disk ohne vorherige Anpassung zu ermöglichen!

Es gibt nämlich viele Programme, die sich nur sehr schwer oder auch gar nicht für Disk-Betrieb umbauen und starten lassen. Dazu gehören u.a. Programme mit ausgefeiltem LIST-Schutz oder solche, die Daten-Files speichern und Laden für den Gebrauch im Hauptprogramm.

Weil es dann unmöglich sein kann, TRDOS-Befehle einzuarbeiten, ist eine andere Vorgehensweise erforderlich!

Dazu wird das Programm geladen und ein oder mehrere Data-Files geschaffen. Anstatt dazu jedoch die im Programm enthaltenen SAVE-Anweisungen zu benutzen, wird dabei der Magic-Button verwendet.

Damit wird eine spezielle Routine im TRDOS aufgerufen, die einfach, den gesamten Speicherinhalt ab Adresse 16384 (Screenanfang) bis 65535 (Speicherende) auf Disk bringt! Folglich wird beim Laden eines solchen Files auch das komplette Datenpaket (49152 Bytes = 192 Blöcke) in den Speicher geholt, statt des eigentlich vielleicht viel kürzeren Programms!

Der Nachteil, den gesamten Speicherinhalt laden zu müssen statt individuell angepaßter Files, wird wettgemacht durch die hohe Geschwindigkeit mit der TRDOS-Befehle ausgeführt werden!

Eine Formatierte leere Disk sollte für diese Zweck bereit gehalten werden. Diese **muß** unbedingt in Laufwerk A stecken!

Vorausgesetzt, diese Bedingungen sind erfüllt und das TRDOS ist eingeschaltet, dann muß folgendermaßen vorgegangen werden:

- (1) Zurück ins BASIC durch Eingabe von
RETURN <ENTER>
- (2) Systemschalter in Stellung 'AUS' bringen
- (3) Speicher löschen mit
PRINT USR 0 <ENTER>
- (4) Programm ganz normal von Kassette laden und starten
- (5) Wenn es gestartet hat, kurz den MAGIC BUTTON drücken und sofort wieder loslassen

Der Speicherinhalt des Computers wird nun als File(s) auf Disk abgelegt. Allerdings gibt es folgende Unterschiede:

SPECTRUM & PLUS : Es wird nur ein File geschaffen, der den Inhalt des kompletten 48 kB des RAM umfaßt. Ein Auflisten mit CAT zeigt dann, daß der Name "@" ist und der Filetyp CODE!

SPECTRUM 128 : Bis zu 7 Files werden gespeichert, abhängig davon, ob die 'Bänke' der oberen 64 kB RAM benutzt wurden. Dazu ein Beispiel, für den Katalog nach Verwendung des MAGIC BUTTON:

```
Title: Test
7 File(s)
0 Del. File
A: @ <C> 192: @7 <C> 64
A: @6 <C> 64: @4 <C> 64
A: @3 <C> 64: @1 <C> 64
A: @8 <C> 1
```

Der gemeinsame Name dieser Files ist also "@"!

Um einen solchen File wieder zu laden wird bei allen SPECTRUM-Computern an Stelle des Befehls RUN oder LOAD das Kommando GOTO (Taste 'G') benutzt:

GOTO "@"CODE

Das Programm kann natürlich auch umbenannt oder kopiert und mit GOTO- aufgerufen werden mit neuem Namen. Auch dazu Beispiele:

```
NEW "Game1","@"CODE COPY
"b:Game1","a:@"CODE
```

Um "Game1" nun aufzurufen, wobei die Disk wieder in Laufwerk A stecken muß, wird folgendes eingegeben:

GOTO "Game1"CODE

SPECTRUM 128 : Es ist natürlich auch möglich, alle erzeugten Files einzeln umzubenennen und zu kopieren. Aber dafür gibt es auf der Utility-Disk auch ein hilfreiches Programm, "magig" genannt! Sehen Sie im Anhang nach. Dies Hilfsprogramm kann auch zum Löschen von per MAGIC-BUTTON erzeugten File-Gruppen herangezogen werden!

Bevor man ein MAGIC-BUTTON-Programm speichert, muß man sicherstellen, daß die verwendete Disk nicht schon Files mit den Namen "@" oder "@1" usw. enthält!

Einige Programme verwenden auch nicht dem Standart entsprechende Tastaturabfrage-Routinen. Dann kann es notwendig werden, als erstes Zeichen des Namens ein "\$" zu verwenden. Beispiel:

GOTO "\$Game1"CODE

Wurde bei der Überprüfung des Programms vorm Kopieren schon festgestellt, daß dieses Problem auftritt, dann kann das erforderliche "\$" gleich berücksichtigt werden!

NEW "\$Game2","@"CODE

GOTO "\$Game2*CODE

Während MAGIC-BUTTON-Files geladen werden, erscheinen auf dem Bildschirm einige wirre Linien, die oft wie Morsezeichen aussehen. Das hängt damit zusammen, daß bei dieser speziellen SAVE- und LOAD-Methode die TRDOS-Systemvariablen vorübergehend im Screen abgelegt werden. Wenn diese Striche verschwinden, dann startet das Programm an der Stelle, an der es unterbrochen wurde!

12) FILES MIT DATA-FELDERN

Der normale (SOS) Befehl zum Laden oder Speichern eines numerischen Datenfeldes (Array) z.B. unter dem Namen "Money" lautet bekanntlich:

```
LOAD "Money" DATA M()                oder  
SAVE "Money" DATA M()
```

Für ein String-Array wird nur das '\$'-Zeichen angehängt:

```
LOAD "Money" DATA m$()
```

Wie bei den meisten TRDOS-Befehlen wird natürlich auch hier die SOS-Syntax übernommen, jedoch ergänzt durch den TRDOS-Aufruf und eventuell einer Laufwerksangabe! Datenfelder werden allerdings kaum jemals direkt aus dem TRDOS heraus angesprochen!

Üblicherweise werden Arrays geladen oder gespeichert aus laufenden Programmen. Die Syntax für den Gebrauch in einem BASIC-Programm sieht also folgendermaßen aus:

```
RANDOMIZEUSR 15619: REM: LOAD "Money" M()                oder  
RANDOMIZEUSR 15619: REM: SAVE "Money" M()
```

Beachten Sie bitte, daß dies nur gilt, wenn Laufwerk 'A' als Hauptlaufwerk (Default drive) eingestellt ist! Besitzen Sie jedoch mehrere Laufwerke, so ist es oft zweckmäßig, die Daten-Diskette immer ins gleiche, z.B. das 'B'-Laufwerk, einzulegen. Im Hauptlaufwerk 'A' kann dann die Disk mit dem Hauptprogramm stecken bleiben. Beim Zugriff auf die Daten-Disk braucht also immer nur vorübergehend auf's Laufwerk 'B' umgeschaltet zu werden, z.B. mit:

```
RANDOMIZEUSR 15619: REM: LOAD "b:Money" DATA M()
```

Selbstverständlich muß bei allen sonstigen Befehlen wie COPY, ERASE, NEW usw. stets der File-Typ, hier also DATA, mit angegeben werden, dann sonst sucht das TRDOS nach einem BASIC-File gleichen Namens! Beispiel für korrekte Syntax:

```
COPY "a:Money","B:Money" DATA
```

13) SEQUENTIELLE UND RANDOM ACCESS FILES

13.1 Allgemeine Beschreibung

Im letzten Abschnitt wurde der Umgang mit normalen DATA-Array-Files erklärt. Das TRDOS unterstützt jedoch zwei weitere Typen, die Sequentiellen- und Random-Access-Files! Um diese benutzen zu können muß grundsätzlich zuerst ein Strom (Stream) geöffnet werden. Beim SPECTRUM sind bis zu 16 Streams möglich, allerdings werden Stream 0-3 schon vom System selbst belegt. Somit stehen noch Stream 4-15 für's TRDOS zur Verfügung! Beim Öffnen eines Streams für Sequentielle- oder Random-Access-Files werden außerdem noch 336 Bytes zusätzlicher Speicherplatz reserviert.

Sowohl Numerische-, als auch String-Variablen lassen sich in diesen Daten-Files speichern, wobei Numerische- jedoch automatisch vom Rechner in String-Variablen verwandelt werden. Als Ende-Markierung setzt das System bei den jeweiligen String ein 'Carriage Return', also den ASCII-Code 13!

Ein Sequentieller File speichert die Daten seriell, das heißt alle direkt hintereinander, vergleichbar der Speicherung auf einem Kassettenband. Will man einen solchen File bis zum Ende lesen, ist es erforderlich, damit am Anfang zu beginnen. Ein Eintrag endet dann jeweils mit Code 13, der nächste folgt unmittelbar dahinter und so weiter, wobei jeder unterschiedlich lang sein darf!

Dagegen werden beim Random-Access- (Wahlfreier Zugriff) File die Daten in jeweils gleich langen nummerierten Blöcken gespeichert. Ein Eintrag in solch einem File kann folglich gespeichert oder gelesen werden, indem man seine Nummer angibt! Daher geschieht ein Zugriff auf den letzten Eintrag genau so schnell wie auf den ersten.

Bezüglich der Laufwerkswahl, Kopieren, Löschen, Namen ändern usw. gibt es keine Unterschiede zur Befehlssyntax bei den anderen File-Typen. Die einzige Neuerung ist die Kennzeichnung der Sequentiellen- bzw. Random-Access-Files durch eine andere Typ-Bezeichnung, nämlich '#' an Stelle von 'CODE' oder 'DATA'. Dazu folgende Beispiele:

```
COPY "b:Phone","aSPhone"#  
NEW "Neuname","Altname"#  
ERASE "Phonlist"#
```

Genau wie die DATA-Arrays werden die Sequentiellen- und Random-Access-Files selten aus dem TRDOS direkt angesprochen. Deshalb werden die Beispiele auch nur für eine Anwendung in BASIC-Programmen beschrieben.

13.2 Sequentielle Files

Ein Sequentieller File kann jeweils nur zum Schreiben oder Lesen geöffnet werden, niemals jedoch für beides zugleich! Zum Öffnen eines Files zum Schreiben gilt folgende Syntax:

```
OPEN Stream-Nummer, "Filename",W
```

Ist der Stream geöffnet, können mit Hilfe des BASIC- Befehlswortes 'PRINT' Daten in den gewünschten Stream geschrieben werden:

```
10 LET dos=15619  
20 RANDOMIZE USR dos: REM: OPEN #4,"Test",W  
30 PRINT #4; "Dies ist eine Test-Zeile!"  
40 RANDOMIZE USR dos: REM: CLOSE #4
```

Hier wurde zur Abwechslung mal die Variable 'dos' an Stelle von '15619' verwendet! Der String in Zeile 30 wird nun im File "Test" auf Disk gespeichert. Der Eintrag kann natürlich auch durch eine Variable ersetzt werden, in die der Text vorher geschrieben wird. Würde der File nicht geschlossen, wären die eingegebenen Daten verloren! Jeder geschlossene File kann erneut zum Schreiben, Lesen oder Ändern geöffnet werden und unter neuem Namen wieder gespeichert werden.

Zum Öffnen eines Files zum Lesen gilt folgende Syntax:

```
OPEN Stream-Nummer, "Filename",R
```

Ist der Stream geöffnet, können mit Hilfe des Befehlswortes 'INPUT' Daten über den gewählten Stream gelesen werden:

```
10 LET dos=15619  
20 RANDOMIZE USR dos: REM: OPEN #7,"Test",R  
30 INPUT #45 a$  
40 RANDOMIZE USR dos: REM: CLOSE #7
```

Der Daten-File "Test" wird nun von Disk in Variable a\$ geladen!

13.3 Random Access Files

Ein Random-Access-File ist ein Datensatz, der eine Anzahl von 0 aufwärts nummerierter Einträge enthält. Die Länge eines Eintrags (max. jeweils 254 Bytes) wird beim Öffnen des Files festgelegt!

Der Eintrag wird als Einzel-String abgespeichert. Jeder kürzere, als die gewählte Länge, wird linksbündig gespeichert mit einem ASCII-Code 13 als Abschluß! Die Rest-Bytes bleiben ungenutzt. Ist der eingegebene String aber länger als zuvor gewählt, dann wird er in Abschnitte der eingestellten Länge zerlegt und abgespeichert ohne die Endemarkierung 'Carriage Return'.

Beim ersten Öffnen des Random-Access-File, reserviert das TRDOS automatisch 16 Sektoren (4 kB) auf Disk, für File-Längen > 4 kB jedoch entsprechend mehr Platz! Zum öffnen eines Random-Access-File es gilt folgende Syntax:

OPEN Stream-Nummer, "Filename" RND, Eintragslänge

Zum Schreiben in den File gilt folgende Befehlsstruktur (das RANDOMIZE ... ist bei PRINT und INPUT nicht erforderlich!):

PRINT Stream-Nummer; Eintragsnummer, Variablen-Name

Ein praktisches Beispiel dazu, bei dem ein in der Variablen 'd\$' geschriebener Text auf Disk gebracht wird:

```
100 RANDOMIZE USR 15619: REM: OPEN #8, "Adressen"RND,100
110 PRINT #8; 72, d$
120 RANDOMIZE USR 15619: REM: CLOSE #8
```

Damit wird Stream 8 geöffnet für den File "Adressen", in dem jeder Eintrag 100 Zeichen lang sein soll. In Zeile 110 wird dann der Inhalt der String-Variablen 'd\$' als Eintrag Nummer 72 im File abgelegt!

Hier gilt natürlich das Gleiche wie bei den Sequentiellen Files, es kann nur in einem geöffneten File gelesen oder geschrieben werden! Um Daten aus dem File zu Lesen gilt folgende Syntax:

INPUT Stream-Nummer; (Eintragsnummer), Variablen-Name

Wieder ein kurzes Beispiel zur Demonstration:

```
100 RANDOMIZE USR 15619: REM: OPEN #12, "Test"RND, 20
110 INPUT a$
120 PRINT #12; 15, a$
130 ...
570 INPUT #12; (15), a$
580 PRINT a$
590 ...
990 RANDOMIZE USR 15619: REM: CLOSE #12
999 STOP
```

Dies ist zwar kein vollständiges Programm, aber es veranschaulicht im Wesentlichen die Vorgehensweise. In Zeile 110 wird der Eintrag eingetippt und in Zeile 120 gespeichert. Zeile 570 liest den Eintrag mit der Nummer 15 (beachten Sie, daß es sich dabei um den 16. Eintrag handelt, weil ab 0 gezählt wird) und lädt ihn in die String-Variablen 'a\$'. Mit Zeile 580 erscheint der Eintrag auf dem Bildschirm. Nach Beendigung aller File-Operationen muß der Stream natürlich wieder wie in Zeile 990 gezeigt, geschlossen werden!

Die Daten des Files sind verloren, wenn der Rechner abgeschaltet wurde, bevor der Stream geschlossen ist!

Es ist auch erlaubt, einen Eintrag aus mehreren Variablen zusammensetzen. Das Programm besitzt die Fähigkeit, diese für eine Schreib-Operation zusammenzufassen. Genauso muß dann eventuell bei Lesen der Eintrag wieder in mehrere einzelne Stringvariablen zerstückelt werden!

14) DIREKTES LESEN UND SCHREIBEN IN SEKTOREN

Zusätzlich zu den 3 in den letzten Kapiteln schon beschriebenen File-Typen bietet das TRDOS noch die Möglichkeit, Sektoren (zu je 256 Bytes) auf der Disk direkt zu beschreiben und zu lesen.

14.1 Der PEEK-Befehl

Der PEEK-Befehl erlaubt das Lesen jedes beliebigen Teilstücks eines Files auf Disk und dessen Übertragung ins RAM! Es kann nur ein Sektor auf einmal transferiert werden in jeden Bereich des Speichers. Die Befehls-Syntax sieht folgendermaßen aus:

PEEK "Filename" RAM-Adresse, Sektornummer

Dazu ein konkretes Beispiel:

PEEK "Test"30023,5

Hier wird der 5. Sektor des Files "Test" gelesen und sein Inhalt in den Speicher ab Adresse 30023 übertragen, wo er dann für eine Bearbeitung zugänglich ist. Es empfiehlt sich, RAMTOP vorher mit CLEAR xxxxx auf einen Wert unterhalb der Ladeadresse zu setzen!

14.2 Der POKE-Befehl

Mit dem POKE-Befehl läßt sich ab jeder Speicheradresse ein 256 Bytes langer Bereich in einen beliebigen Sektor eines Files auf Disk kopieren. Die Befehls-Struktur ähnelt dem PEEK-Befehl!

POKE "Filename" RAM-Adresse, Sektornummer

Praktisch sieht's so aus:

POKE "b:Test"30023,10

Hier wird eine Sektorlänge Daten ab Adr. 30023 in Sektor 10 des Files "Test" auf Disk im Laufwerk B kopiert! Da der Befehl den Inhalt von Files ändert, bitte sehr vorsichtig damit umgehen!

15) PROGRAMMIERUNG DER TRDOS-BEFEHLE IN MASCHINENSPRACHE

Der Einbau von TRDOS-Routinen in MC-Programme ist relativ einfach. Drei Elemente gehören dazu:

- (1) Die Nachbildung des BASIC-TRDOS-Befehls in Maschinensprache
- (2) Die MC-Routine, um diesen zu übergeben
- (3) Eine MC-Routine, die' das System in den ursprünglichen Ausgangszustand vorm Aufruf versetzt und den Befehl ausführt

Der verwendete RAM-Bereich ist natürlich dem jeweiligen Programm entsprechend, auszuwählen. Im Beispiel benutzen wir für 'SAVE' Adresse 49000 und 49500 für 'LOAD', außerdem Adresse 50000 für die Routine, die beides aufruft! Element 1 ist also bei Adresse 49000 bzw. 49500 zu finden und die Elemente 2 und 3 ab 50000.

Beispiel:

Adresse	Code	Basic	Kommentar
49000	234	REM	Die Codes entsprechend Anhang A des SPECTRUM-Handbuchs!
49001	58	:	
49002	248	SAVE	
49003	34		
49004	69	E	
49005	120	x	
49006	97	a	
49007	109	m	Filename hier: "Example"
49008	112	p	
49009	108	l	
49010	101	e	
49011	34		
49012	13	ENTER	Abschluß immer mit ENTER!

Für den LOAD-Befehl, der in unserem Beispiel bei Adresse 49500 beginnen soll, wird das Gleiche eingegeben, nur bei 49502 wird dort an Stelle von 248 (SAVE) 239, der Code für LOAD eingesetzt.

Beide Routinen können überall ohne Änderung in den Speicher geschrieben werden, denn sie sind frei verschiebbar! Dagegen ist die Initialisierungs-Routine, die wir bei Adresse 50000 beginnen lassen, nicht ohne Anpassung überall lauffähig. Es müssen die eventuell verschobenen Adressen der obigen SAVE- und LOAD- Routinen neu angegeben werden! Dazu eignet sich am Besten ein Assembler, deshalb werden hier nur die Z-80 Mnemonics angegeben:

```

CHADD EGU 23645      ;Adresse der Systemvariablen CHADD
ORG xxxxx           ;xxxxx = Anfangsadr. der SAVE-Routine
LD HL, (CHADD)      ;Wert aus CHADD ins HL-Register laden
LD (TEMP), HL       ;HL-Inhalt bei Adr. von TEMP ablegen
LD HL, 49000        ;Adresse der SAVE-Routine
LD (CHADD), HL      ;CHADD zeigt nun auf unsere Routine
CALL 15619          ;TRDOS mit Hilfe von CHADD aufrufen
JP BACK            ;Zurück zur Programmadr. von der die
                   ;ganze Routine aufgerufen wurde
                   ;
LD HL, (CHADD)      ;Hier beginnt die LOAD-Routine
LD (TEMP), HL       ;Befehlsfolgen wie bei SAVE
LD HL, 49500        ;Auch diese Adr. ändern, wenn LOAD-
LD (CHADD), HL      ;Routine verschoben wurde
CALL 15619          ;TRDOS aufrufen
BACK LD HL, (TEMP)  ;Wert aus TEMP zurückholen und
LD (CHADD), HL      ;damit CHADD auf Ausgangswert setzen
RET                ;Rücksprung hinter die Aufrufadresse
                   ;
TEMP                ;Platzhalter für Zwischenspeicherung
                   ;des ursprünglichen CHADD-Wertes

```

Die ganze Routine zum Aufruf der TRDOS-Befehle SAVE und LOAD beansprucht einschließlich der Rücksprungroutine nur 47 Bytes!

Anmerkung:

Diese Methode ist nicht die eleganteste, um TRDOS-Befehle in MC- Programme einzubauen. TRDOS enthält 24 sogenannte Funktionen zur Nutzung vieler Befehle. Eine Beschreibung führt hier zu weit und die Anwendung setzt gute Kenntnisse in MC-Programmierung voraus. Wen's interessiert, der kann mir schreiben oder meine Serie im USER-CLUB-Wuppertal-Info nachlesen!

16) FEHLER-MELDUNGEN

16.1 Direkt-Meldungen

Wenn Sie im TRDOS fehlerfreie Befehle eingeben, werden diese sofort ausgeführt. Handelte es sich nicht um einen TRDOS-Befehl (wie LOAD, RUN, FORMAT), wird er einfach ignoriert! Trat aber ein Syntax-Fehler oder ein Fehler während der Ausführung auf, dann wird eine Fehlermeldung ausgegeben. Die wichtigsten Meldungen und wann sie auftreten, wird im Folgenden erklärt!

(1) No disk

Im Laufwerk befindet sich keine oder eine unformatierte Disk, eventuell ist auch die Laufwerksklappe nicht verriegelt! Unter der Fehlermeldung erscheint hinter dem Prompt 'A>' wieder der eingegebene Befehl, so daß nach Korrektur ein Druck auf die ENTER-Taste genügt, um den Befehl erneut abarbeiten zu lassen.

(2) No file(s)

Das TRDOS kann den File auf der Disk nicht finden! Auch wenn der gesuchte Filename falsch eingetippt wurde oder ein falscher File-Typ angegeben wurde, kommt diese Meldung. Beispiele dazu:

```

LOAD "Test" CODE      statt      LOAD "Test"
ERASE "Dummy1" DATA statt      ERASE "Dummy2" DATA

```


Um die Fehlermeldung zu nutzen, verwendet man eine Variable im TRDOS-Befehl. Diese enthält dann nach Abschluß des Befehls den Fehlercode. Dazu wieder zwei Beispiele:

Beispiel 1: **LET A=USR 15619: REM: CAT**

Die Variable A enthält anschließend den Fehlercode, wie sich mit 'PRINT A' leicht überprüfen läßt!

Beispiel 2:

```
10 CLEAR 65367
20 LET err = USR 15619: REM: LOAD "Copy"CODE
25 REM Der Fehlercode findet sich hier in der Variablen err!
30 IF err = 1 THEN CLS: PRINT AT 10,1; "FILE ""Copy ""CODE IST NICHT AUF DER
DISK":STOP
```

17

) WISSENSWERTES ÜBER DAS TRDOS

17.1 Das TRDOS beansprucht 112 Bytes RAM für Systemvariablen

17.2 Ohne BETA-Interface beginnt nutzbarer Speicher bei:
(a) Adresse 23755 ohne zusätzliches INTERFACE 1 und
(b) Adresse 23813 mit angeschlossenem INTERFACE 1!

Mit BETA-Interface beginnt nutzbarer Speicher bei:
(a) Adresse 23867 ohne zusätzliches INTERFACE 1 und
(b) Adresse 23925 mit angeschlossenem INTERFACE 1!

Um ein Programm zu Speichern, Laden oder zu Starten, welches den RAM-Bereich von 23755 bis ca. 23925 unbedingt braucht, sollte man den MAGIC BUTTON (siehe Abschnitt 11.3) verwenden!

17.3 Das TRDOS speichert Programme in Sektoren auf Disk (Abschnitt 1.2). Ist es länger als ein Sektor (256 Bytes), wird es in entsprechend viele aufgeteilt, bis das ganze Programm gespeichert ist.

Um zum Beispiel 522 Bytes zu Speichern, werden 3 Sektoren benötigt. Vom dritten werden allerdings nur 10 Bytes tatsächlich belegt. Beim Laden werden später auch nur diese 10 Bytes geladen, die restlichen 246 bleiben unberücksichtigt, damit keine Daten im Speicher davon überschrieben werden!

17.4 Zusätzlich zu den schon erwähnten 112 Bytes für zusätzliche Systemvariablen benötigt TRDOS weitere 256 Bytes RAM bei allen Zugriffen auf Disk. Dieser Zwischenspeicher wird bei Bedarf auch vergrößert.

1. Bei Ausführung eines TRDOS-Befehls wird ein eventuell vorhandenes BASIC-Programm zum Schaffen des Zwischenspeichers hochgeschoben im RAM. Nach Beendigung wird es zurück an den ursprünglichen Platz kopiert. Das alles geschieht unsichtbar, unbemerkt und in Bruchteilen einer Sekunde!

17.5 Der MOVE-Befehl beansprucht mindestens 4 kB des SPECTRUM-RAM's als Zwischenspeicher. Dieser Befehl sollte nicht verwendet werden, solange sich noch ein Programm im Speicher befindet, also ist es erforderlich ihn vorher zu löschen! Dazu kann aus dem SOS einfach RANDOMIZE USR 0 eingegeben werden oder einfacher, der Systemschalter kurz in Reset-Position und zurück in seine Ausgangsstellung gebracht werden.

17.6 Bis zu maximal 128 Files lassen sich auf einer Disk unterbringen, dann ist der Katalog voll!

18) ANSCHLUSSBELEGUNG DES LAUFWERK-STECKVERBINDERS

Der Laufwerks-Steckverbinder (siehe Diagramm unten) ist Shugart-kompatibel ausgelegt! Die Anschlüsse sind wie folgt belegt:

Pin-Nr.	Pin-Nr.
Masse ----- 1	2 --- Nicht belegt
Masse ----- 3	4 --- Nicht belegt
Masse ----- 5	6 --- Laufwerk D anwählen
Masse ----- 7	8 --- Index-Loch Impuls
Masse ----- 9	10 --- Laufwerk A anwählen
Masse -----11	12 --- Laufwerk B anwählen
Masse -----13	14 --- Laufwerk C anwählen
Masse -----15	16 --- Kopf laden (Motor an)
Masse -----17	18 --- Richtung bei Spurwechsel
Masse -----19	20 --- Schrittsignal (Step)
Masse -----21	22 --- Datenleitung für Schreiben
Masse -----23	24 --- Schreibfreigabe-Signal
Masse -----25	26 --- Spur 0 Erkennung
Masse -----27	28 --- Schreibschutz-Signal
Masse -----29	30 --- Datenleitung für Lesen
Masse -----31	32 --- Signal für Seitenwahl
Masse -----33	34 --- Nicht belegt

Die Anschlußbelegung der Pins des Steckers ist von rechts auf das Interface (siehe Bild 1) gesehen wie folgt:

2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33

Es werden zusammen mit dem BETA-Interface 3 Hilfsprogramme (Utilities) geliefert und zwar folgende:

tapecopy - Zum Übertragen von Kassettenprogrammen auf Disk

magic - Umbenennen, Kopieren und Löschen von per MAGIC- BUTTON gespeicherten File-Gruppen

doctor - Disketten-Monitor-Programm

Um ein Programm zu starten, braucht die Utility-Disk nur ins Laufwerk gesteckt und **RUN** oder **RUN "boot"** eingegeben werden!

Das BASIC-Programm "boot" bringt nun ein Menu über daß Sie eins der Programme anwählen können. Alternativ dazu kann man jedes Programm auch direkt aufrufen mit:

```

RUN "tapecopy"           oder
RUN "magic"             oder
RUN "doctor"

```

Auf alle Fälle sollte die Utility-Disk sofort aus dem Laufwerk entfernt werden, nachdem das gewünschte Programm geladen ist. Sicherheitshalber niemals den Schreibschutz entfernen!

A: TAPECOPY

Es handelt sich um ein relativ einfaches Kopierprogramm, daß für's Kopieren gut geschützter kommerzieller Programme nicht geeignet ist! Um eigene Programme und Files von Kassette auf Disk zu bringen ist es jedoch ganz brauchbar. Es kopiert BASIC-, CODE-, ARRAY- und DATA-Files, vorausgesetzt sie sind nicht länger als max. 40000 Bytes (beim SPECTRUM & PLUS). Nach dem Laden von "tapecopy" erscheint folgendes Menu:

**TAPE TO DISK COPIER VER 5.0
TECHNOLOGY RESEARCH LTD**

**Data type:
Filename:
Code length:
Start line:
Basic length:**
**PRESS <BREAK> TO STOP
START TAPE**

Wenn "tapecopy" geladen ist, Utility-Disk entfernen, denn das Programm von Kassette wird über's gleiche Laufwerk gespeichert!

Nach Austausch der Disk, ist nur noch erforderlich, das Band wie gewohnt zu Starten. Alles weitere geschieht automatisch! Sie werden am Schluß aufgefordert, das Band anzuhalten. Im selben Moment wird der File auch schon auf die Disk kopiert unter dem gleichen Namen wie auf Kassette. Anschließend werden Sie gebeten, das Band erneut zu Starten für den nächsten Transfer! Taucht der gleiche Name und Typ ein zweites Mal auf, wird die erste Version auf Disk überschrieben! Also aufpassen!

Es können folgende Meldungen beim Kopieren erscheinen:

START TAPE: **Wiedergabe-Taste am Rekorder drücken!**

STOP TAPE: **Rekorder anhalten, bis Programm kopiert ist!**

TAPE LOAD ERROR: **Fehlermeldung des SOS. Zurückspulen und neuen Versuch machen!**

NOT ENOUGH MEMORY: **Das Programm ist länger als 40000 Bytes. Um es trotzdem kopieren zu können, notieren Sie bitte die Header-Daten und drücken Sie <BREAK>! Dann Speicher mit RANDOMIZE USR 0 löschen und Programm normal laden und nun mit Hilfe der Header-Daten wie beschrieben auf Disk speichern (siehe Kapitel 10.11!**

B: MAGIC

Wurde der MAGIC BUTTON benutzt, um ein SPECTRUM 128-Programm auf Disk zu bringen, wird eine ganze Gruppe (bis zu 7) Files gespeichert. Es ist sehr mühsam, diese von Hand nacheinander umzubenennen oder zu kopieren. Dieses Menu-gesteuerte Programm erleichtert Ihnen das Umbenennen, Löschen und Kopieren sowohl mit einem Einzel-, als auch mit Doppel-Laufwerk!

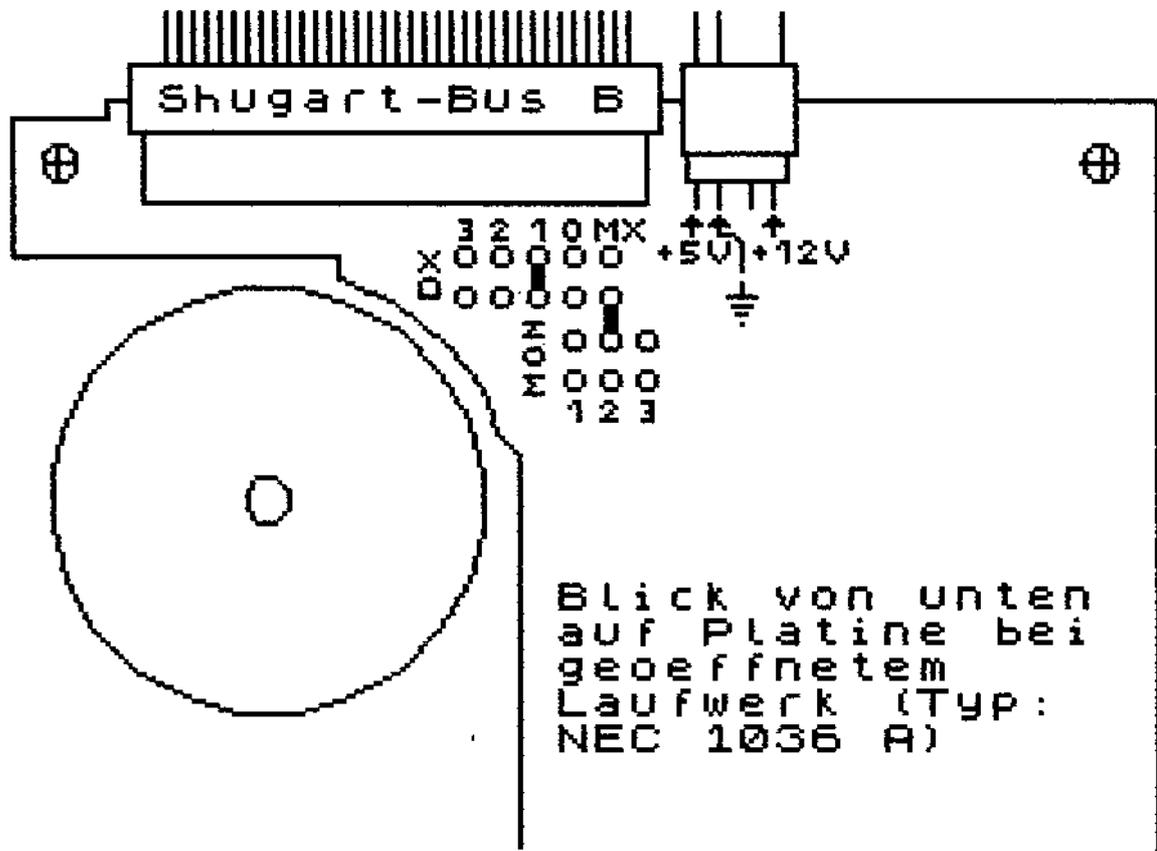
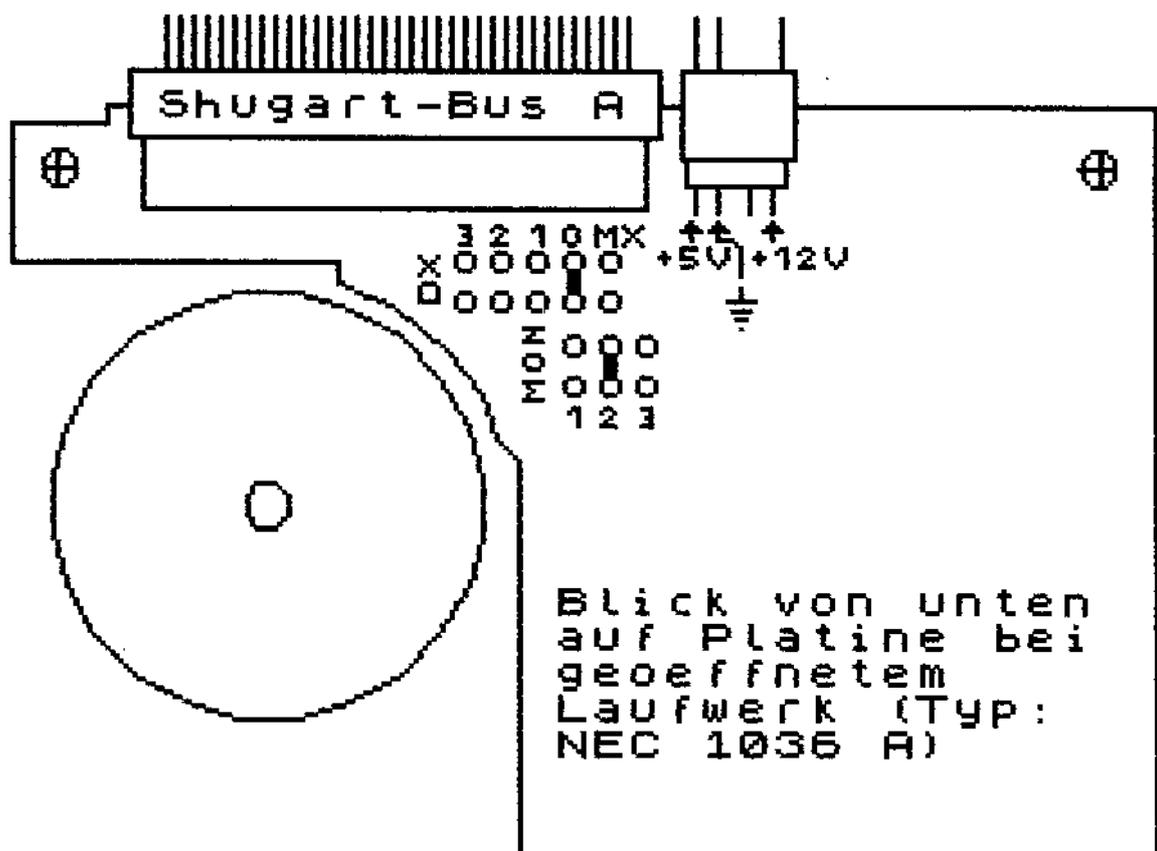
C: DOCTOR

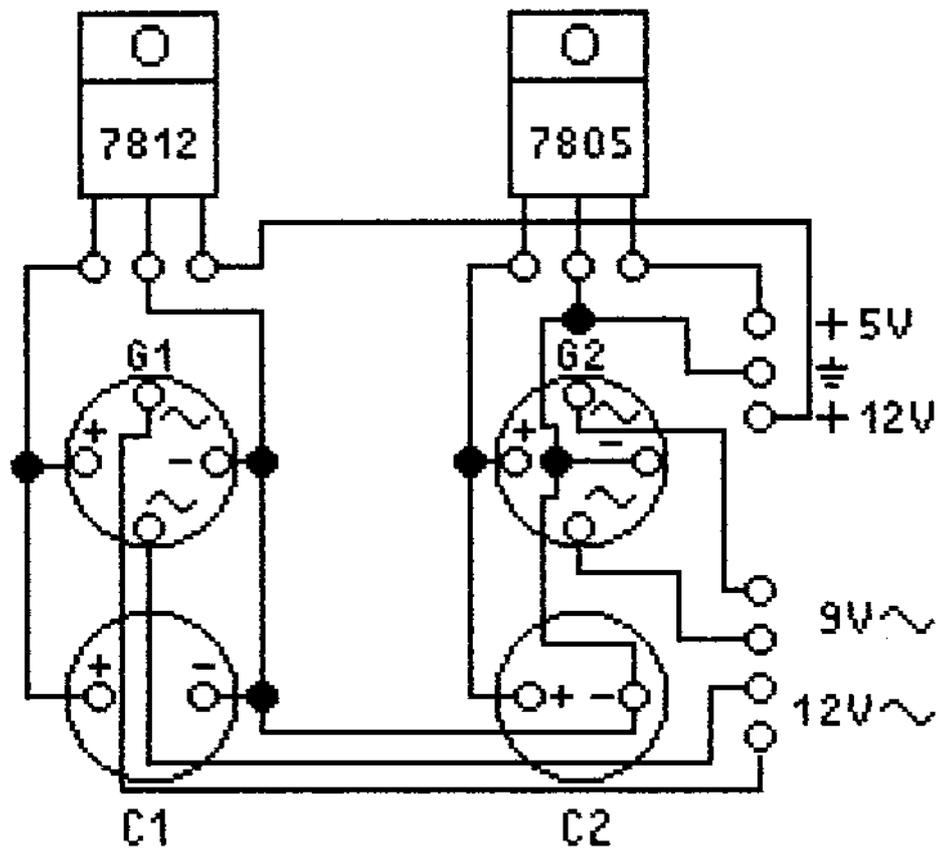
Dieses Programm arbeitet auch Menugesteuert. Sie können damit Sektoren auf einer Disk lesen, inspizieren und modifizieren! Es lassen sich verschiedene Laufwerke (A,B,C,D) einstellen und Sektoren-Nummern von 0-15, sowie Track-Nummern von 0-159! Danach kann der gewünschte Sektor gelesen und im RAM gespeichert werden. Die Sektorgröße beträgt bekanntlich 256 Bytes!

Der gespeicherte Sektorenhalt läßt sich nun anzeigen und auch editieren. Dazu verlangt das Programm die Eingabe einer Startadresse im Bereich von 0 bis 255 in Dezimal oder Hexadezimal (dann mit nachgestelltem 'h' oder 'H')! Wenn der Inhalt gelistet wird, kann jederzeit durch Drücken der <ENTER>- Taste abgebrochen werden. Falls gewünscht, kann der geladene Sektor aus dem Speicher, zurück auf die Disk geschrieben werden.

Der 'Change Disk'-Befehl teilt dem System mit, daß eine andere Disk vielleicht anderen Formats, eingelegt wurde. Es wird empfohlen, diesen Befehl nach jedem Disk-Wechsel aufzurufen!

Weiterhin gibt es einen Befehl, mit dem sich eine Disk verifizieren läßt, das heißt, sie kann auf fehlerhafte Sektoren durchsucht werden!





Netzteil für Disketten-Laufwerk Typ NEC 1036 A (von Micro P)