

# BIOS 2.1

BIOS zum Genie I/II mit Option  
RAM 64.1, EG 64.3 oder  
Lubomir Soft Switch

HANDBUCH

*Günter Ermer*

Copyright 1983 by Schmidtke electronic Software  
written by Günter Ermer



BIOS 3.0 / Ergänzungen zum BIOS 2.1 - Handbuch  
=====

Ab sofort wird unser CP/M 2.2 für Genie-Rechner mit dem neuen BIOS 3.0 ausgeliefert. Dieses Handblatt erläutert Ihnen die Änderungen zum BIOS 2.1 bzw. 2.2, dessen Handbuch mitgeliefert wurde.

Diese neue CP/M - Version mit BIOS 3.0 läuft auf folgenden Systemen:

System	Bankinlogik
TRS 80 Model 1 *	LSB
Genie I/II	EG 64.1, EG 64.3, LSS 1 oder 80-Zeichen-Karte
Genie IIs	keine Bankinglogik erforderlich
Speedmaser	keine Bankinglogik erforderlich

Auf allen Systemen wird die 80-Zeichen-Karte unterstützt. Hierzu befindet sich ein SYS-File "SYS80.SYS" auf der Diskette.

Sicherheitskopien:  
-----

Sie sollten grundsätzlich nicht mit der Originaldiskette arbeiten ! Ihre Originaldiskette ist daher mit einem Autostart-Kommando versehen, das direkt nach dem Anstarten eine Sicherheitskopie durchführt. Bitte legen Sie dazu eine leere Diskette in das B-Laufwerk, Ihre Originaldiskette in das A-Laufwerk und betätigen den RESET - Taster.

Falls Sie kein 2tes Laufwerk besitzen oder falls Ihr B-Laufwerk eine andere Spurzahl hat (z.B. A-Laufw. 40Tr. und B-Laufw. 80Tr.), so halten Sie bitte beim BOOTEN die <NEW-LINE> bzw. <ENTER> - Taste gedrückt, die automatische Kopieroutine wird damit abgebrochen. Die <ENTER>-Taste muß solange gedrückt werden, bis sich Ihr System mit A.0 => meldet. Nun formatieren Sie eine neue Diskette mit dem FORMAT - Programm, stellen mit PD die neuen PDRIVE-Parameter (=Angabe der Laufwerkparameter) ein und kopieren schließlich mit PIP oder COPY. Mit SYSGEN übertragen Sie die CP/M-Systemspuren, die zum booten der neuen Diskette notwendig sind.

■ Danach legen Sie Ihre Original-Diskette an einen sicheren Ort und arbeiten nur noch mit der Kopie !!!!!!!

Am besten kopieren Sie sich direkt mehrere Arbeitsdisketten.

Der 80-Zeichenkarten-Treiber:  
-----

Um die 80-Zeichen-Karte zu aktivieren geben Sie bitte ein:

A.0 => COPY SYS.SYS=SYS80.SYS <ENTER>

Danach müssen Sie das CP/M neu booten (RESET-Taster betätigen). Ab sofort ist die 80-Zeichen-Karte aktiviert. Falls Sie auch mit 64-Zeichen Programmen arbeiten möchten, sollten Sie sich von beiden Diskettenversionen einen kleinen Vorrat anlegen. Sie können die 80-Z-Aktivierung auch wieder rückgängig machen, indem Sie in das B-Laufwerk eine 64-Z-Version einlegen und danach folgendes eingeben:  
A.0 => COPY SYS.SYS=B:SYS.SYS <ENTER> / (RESET zum booten)

## Neue Möglichkeiten bei aktivierter 80-Zeichen-Karte:

### Zusätzliche Bildschirmsteuercodes:

01 Hex - schaltet auf Inversschrift um  
02 Hex - schaltet auf Normalschrift zurück

### Erweiterte Tastaturroutine:

ctrl 4 - schaltet die 80-Z-Karte auf den nächst  
möglichen Zeichensatz um. Dadurch kann  
man zwischen allen Zeichensätzen hin  
und her schalten.

ctrl : - schaltet die 80-Z-Karte auf den Bild-  
Invers-Modus (dunkle Schrift auf hellem  
Grund). Nochmaliges Betätigen schaltet  
wieder auf Normal-Modus zurück.

## SYSGEN.COM :

Die derzeitige CP/M Version beinhaltet kein BOOTGEN Programm zum Kopieren der Bootspuren mehr, sondern ein SYSGEN Programm. Der Kopierschutz entfällt damit völlig.

Nach dem Anstarten des SYSGEN Programms fragt es als erstes nach einem PDRIVEDATENSATZ. Dort geben Sie den Datensatz an, den die neue Diskette erhalten soll. Wenn Sie hier mit "NEW LINE" antworten, bekommt die neue Diskette den Pdrivedatensatz der alten.

Danach fragt das Programm nach der QUELLDISKETTE. Hier geben sie ein Laufwerk an (A bis D; immer in Großbuchstaben) in dem eine Systemdiskette eingelegt ist. Deren BOOTSPUREN werden eingelesen und ab der Adresse 8000 Hex bis 9FFF Hex im Speicher abgelegt. Auch nachdem Sie das Sysgenprogramm wieder verlassen haben kann der Inhalt des obigen Speicherbereiches mit dem DDT bearbeitet werden. Sie haben ab sofort vollen Zugang zu den Bootspuren. Wenn Sie auf die Frage QUELLDISKETTE mit "NEW LINE" antworten, werden keine Daten von einer vorhandenen Diskette eingelesen, sondern es wird zur Generierung der neuen Systemdiskette der Inhalt des Speicherbereiches von 8000 Hex bis 9FFF Hex genommen.

Als letztes will das SYSGEN Programm ein ZIELLAUFWERK genannt bekommen. Hier können Sie ebenfalls A bis D (nur Großbuchstaben) angeben. Mit ctrl C (auch Break) oder mit F können Sie das Sysgen Programm an dieser Stelle verlassen, ohne neue Bootspuren generieren zu müssen.

## Update Service - BIOS 3.2 :

Ein weiteres Update ist zur Zeit in Arbeit. Das Bios 3.2 enthält u.a. die Einbindung des SIA-MODULs sowie weitere Verbesserungen einzelner Zusatzprogramme. Ihre Adresse liegt uns vor; wenn Sie Ihre CP/M - Registriertkarte ordnungsgemäß abgeschickt haben, erhalten Sie die neue Version sofort nach Fertigstellung KOSTENLOS unaufgefordert frei Haus zugesandt !

## Der Kopierschutz

Die Ihnen vorliegende CP/M Originaldiskette wurde mit einem Kopierschutz versehen. Das soll nicht heißen, daß Sie nicht in der Lage wären, Sicherheitskopien von Ihrer Originaldiskette zu erstellen. Für ein Betriebssystem wäre es auch undenkbar, wenn man zum Betreiben der verschiedenen Programmsysteme immer ein und dieselbe Diskette benötigen würde, um das System einsatzbereit zu machen. In der Regel ist es doch so, daß man fast jede Diskette mit einem Betriebssystem ausrüstet. Das ist auch mit diesem CP/M trotz Kopierschutz möglich.

Der Kopierschutz besteht darin, daß die Spuren 0 bis 2 der Originaldiskette, nur von dem Programm `BOOTGEN.COM` kopiert werden können. Die Systemspuren einer Tochterdiskette sind grundsätzlich nicht kopierbar! Und nur mit kopierten Systemspuren (0-2) ist eine CP/M Diskette bootfähig. Bootfähige Diskette können Sie also nur herstellen, wenn Sie im Besitz einer Originaldiskette sind. Wenn Sie jedoch bootfähige Tochterdisketten erstellt haben, sind diese grundsätzlich ohne Originaldiskette einsatzfähig.

Sie können selbstverständlich auch Dateien, Files und Programme von einer Tochterdiskette auf eine andere kopieren, ohne die Originaldiskette heranziehen zu müssen. Sie können also von sämtlichen Programmen und Dateien Sicherheitskopien herstellen.

Die einzige Einschränkung, der Sie unterliegen besteht darin, daß Sie nur dann bootfähige Tochterdiskette herstellen können, wenn Sie im Besitz einer Originaldiskette sind und das Codewort kennen, mit dem Sie das Programm `BOOTGEN.COM` zum Kopieren der Systemspuren (0-2) in Betrieb nehmen können.

Sollte Ihre Originaldiskette aus irgenwelchen Gründen einmal schadhaft werden, können Sie solange normal weiterarbeiten, wie Sie genügend bootfähige Tochterdisketten besitzen. In einem solchen Fall werden wir, gegen Einsendung Ihrer defekten Originaldiskette, um schnellstmöglichen Ersatz bemüht sein.

Es ist also immer ratsam einen Stoß funktionsfähige Tochterdisketten auf Lager zu haben.

Um eine komplette Kopie Ihrer Originaldiskette zu erstellen, müssen Sie zuerst eine leere Diskette formatieren und dann mit einem `BOOTER` ausstatten, der auf den Systemspuren (0-2) abgelegt werden muß.

Dazu lesen Sie sich die Abschnitte `FORMAT.COM` und `BOOTGEN.COM` in diesem Handbuch durch. Dann formatieren Sie sich mit dem Programm `FORMAT.COM` eine Diskette und kopieren mit dem Programm `BOOTGEN.COM` die Systemspuren auf die neue Diskette. Legen Sie jetzt die Originaldiskette in Laufwerk A und die neue in Laufwerk B. Geben Sie dann den Befehl `PIP B:=A:*.*` ein. Überprüfen Sie jedoch vorher die Einstellung der Laufwerkparameter mit dem Programm `PD.COM`.

Nach der Ausführung dieses Befehls besitzen Sie eine Kopie der Originaldiskette. Am besten fertigen Sie mehrere solcher Kopien an.

Wenn Sie das Handbuch durchlesen und die einzelnen Befehle gleich ausprobieren wollen, machen Sie das am besten mit einer Kopie Ihres CP/M Systems.

Inhaltsverzeichnis

	<u>Seite</u>	
1.	Kopierschutz	1
	Bootprobleme	4
	Übersicht	5
2.	MCP.COM und CCP.COM	3
3.	COPY.COM	6
4.	Die Bildschirmroutine	7
	Tastaturroutine	8
	Tastaturbelegung	9
	Besonderheiten der Tastaturbelegung	10
	Control-Zeichen	11
	Zeichen von 20 hex bis 3F hex	12
	Zeichen von 40 hex bis 5F hex	13
	Zeichen von 60 hex bis 7F hex	14
5.	MCP.SYS	15
	SYS.SYS	16
	FUNKTION.SYS	17
	PDRIVE.SYS	21
	FUNKTION.MAS	18
	PDRIVE.MAS	25
	Grundmodus	25
	P (Listen)	25
	L (Löschen)	25
	K (Kopieren)	26
	A (Anhängen)	26
	S (Beenden)	26
	E (Editieren)	26
	BIOS Byte	27
	Doppelspurverfahren	28
	Doppelsektorverfahren	28
	Logische Organisation	29
	Sektortabelle	30
6.	PD.COM	32
	SYSTEM.COM	37
	BOOTGEN.COM	36
7.	FORMAT.COM	34
	Genie I/II CP/M Formate für Single Density	34
	Genie I/II CP/M Formate für Double Density	35
	LFORMAT.COM	35
8.	Blocking und Deblocking	38
	Serielle Schnittstelle	40
9.	Die 17 Sprungvektoren des BIOS	41
	und deren Konditionen	42
	Speicherorganisation	43
	CP/M und Video Genie	44

MCP.COM und CCP.COM

In diesen zwei Dateien ist je ein CCP (Console Command Prozessor) abgespeichert. Die Datei CCP.COM enthält den originalen CCP von Digital Research.

Die Datei MCP.COM beinhalten einen etwas abgeänderten und erweiterten CCP. Bei der Auslieferung der Diskette sind die Dateien MCP.COM und MCP.SYS indentisch.

Wenn Sie den CCP wechseln wollen, brauchen Sie nur die entsprechende Datei in das File MCP.SYS zu kopieren. Also nach der Befehlsfolge:

```
COPY A:MCP.SYS=A:CCP.COM      ist der Original-CCP aktiv
COPY A:MCP.SYS=A:MCP.COM      ist der neue      CCP Aktiv
```

Nach der entsprechenden Kopieranweisung müssen Sie jedoch noch einen Warmstart herbeiführen, um den gewechselten CCP zu aktivieren.

Die Befehl des neuen CCP in MCP.COM sind dabei wie folgt:

REN	entspricht REN	beim Original-CCP
LIST	entspricht TYPE	beim Original-CCP
DIR	entspricht DIR	beim Original-CCP
KILL	entspricht ERA	beim Original-CCP
USER	entspricht USER	beim Original-CCP
SAVE	entspricht SAVE	beim Original-CCP

Zusätzlich kann man mit dem neuen CCP statt auf 16 Userbereiche auf 32 zurückgreifen.

Die wichtigste Änderung ist jedoch die Tatsache, daß der neue CCP resident im Speicher bleibt. Bei einem Rücksprung aus einem externen Programm erfolgt normalerweise (beim Original-CCP z.B.) ein Warmstart des Systems. Diesen Warmstart unterdrückt der neue CCP erfolgreich. Das Ergebnis ist im Durchschnitt ein schnelleres Arbeiten. Sollten Sie jedoch einmal einen echten Warmstart durchführen müssen, z.B. nach einem Diskettenwechsel, so können Sie das einfach, indem Sie ctrl.C auslösen während der CCP aktiv ist.

Durch dieses Verfahren geht jedoch ca. 3 K Bytes Speicherplatz verloren. Sollten Sie diesen Speicherplatz unbedingt benötigen, so geben Sie beim Anstarten eines externen Programms einfach ein Minuszeichen vor dem Programmnamen ein und der neue CCP blendet sich dann automatisch aus. Beim Rücksprung zum CCP erfolgt dann jedoch ein echter Warmstart. Zwei Beispiele zur Eingabe:

```
A.0 =>-ED TEXT.ASM
A.0 =>-B:FORMAT
```

In beiden Fällen blendet sich der neue CCP aus und stellt dann ca. 3 K Bytes zusätzlich an Speicherplatz zur Verfügung. Sollten beim Anstarten externer Programme mit dem neuen CCP Probleme auftauchen, daß etwa ein Programm in der TPA nicht laufen will, so starten Sie es mit dem Minuszeichen vor dem Filenamen an. Es kann zu solchen Problemen kommen, wenn Programme eigene Diskfehler-routinen ins BDOS einbauen. Daraufhin ist der neue CCP noch nicht ausreichend getestet!

Die Meldung des neuen CCP's ( A.0 => ) ist um die Usernummer erweitert worden.

Die SUBMIT und XSUB-Programme sind mit dem neuen CCP nicht lauffähig. Dafür ist in ihm der JOB-Befehl fest eingebaut. Er ermöglicht es Ihnen Textdateien mit dem ED aufzubauen und wie mit SUBMIT abzuarbeiten. Der Vorteil von JOB ist allerdings, daß keine Datei \$\$\$SUB aufgebaut werden muß. Der JOB Befehl funktioniert also auch auf schreibgeschützten Disketten. Außerdem läuft ein JOB schneller ab als eine SUBMIT Befehlsfolge. Auch können Sie sich mit dem JOB durch externe Programme wie STAT, M80, PIP, DDT... hindurchtasten. Ja Sie können sich sogar durch solche Programme durchtasten, die sich zur Ein- und Ausgabe direkt ans BIOS wenden und das BDOS umgehen. Bei einer solchen illegalen Verfahrensweise von TPA-Programmen kann es jedoch auch mit dem JOB-Befehl zu Problemen kommen. Das muß dann jeweils ausprobiert werden.

Während der Ausführung eines JOB Befehls können Sie den neuen CCP jedoch nicht durch Eingabe eines Minuszeichens ausblenden, da dieser ja die Steuerung des JOB's übernimmt. Sie können zwar innerhalb einer JOB-Textdatei ein Minuszeichen vor einem TPA-Programmnamen eingeben, es wird jedoch ignoriert.

Einen JOB können Sie durch Drücken der BREAK Taste abbrechen. Treten Fehlermeldungen während der Ausführung von CCP-Befehlen auf, so wird ein eventuell aktiver JOB automatisch beendet.

Eine Parameterübergabe in eine JOB-Textdatei wie beim SUBMIT Programm ist noch nicht möglich.

### Bootprobleme

Sollte es bei Ihrer Originaldiskette, oder bei von Ihnen erstellten Tochterdisketten zu Bootproblemen kommen, so liegt das in aller Regel an einer falsch eingestellten Laufwerksgeschwindigkeit. Meistens laufen solche Laufwerke dann zu langsam. Ein Testprogramm zum Testen der Laufwerks-geschwindigkeit ist in Vorbereitung. Sollten Sie also solche Probleme haben und kein entsprechendes Testprogramm besitzen, wenden Sie sich bitte an Ihren Fachhändler.

Entsprechende Testprogramme, die für den TRS 80 geschrieben wurden, sind auf dem Video Genie nur bedingt einsetzbar, da unterschiedliche Taktfrequenzen benutzt werden.

Eine gute Hilfe bietet da das LWT, welches Sie auf allen G-DOS Disketten finden.

Übersicht

Die Ihnen hier vorliegende Beschreibung soll kein CP/M-Handbuch sein. Sie bezieht sich nur auf die besonderen Optionen und Eigenschaften des BIOS 2.1 zum CP/M des Genie I/II. Sie werden sehen, daß eine solche Beschreibung unbedingt notwendig ist, da die Möglichkeiten des BIOS 2.1 den Rahmen des sonst üblichen bei weitem sprengen.

Auf der von uns gelieferten Diskette sind folgende Programme und Dateien zu finden:

MCP	.SYS	(Option)
FUNKTION	.SYS	(Option)
SYS	.SYS	(Option)
PDRIVE	.SYS	(Option)
PD	.COM	(Option)
SYSTEM	.COM	(Option)
COPY	.COM	(Option)
PIP	.COM	(Standard)
STAT	.COM	(Standard)
DDT	.COM	(Standard)
SUBMIT	.COM	(Standard)
XSUB	.COM	(Standard)
ED	.COM	(Standard)
ASM	.COM	(Standard)
LOAD	.COM	(Standard)
DUMP	.COM	(Standard)
DUMP	.ASM	(Standard)
FUNKTION	.MAS	(Option)
PDRIVE	.MAS	(Option)
MASTER	.COM	(Steuerprogramm)
MCP	.COM	(Option)
CCP	.COM	(Option)
FORMAT	.COM	(Option)
LFORMAT	.COM	(Option)
MOVCPM	.COM	(Standard)
DEBLOCK	.ASM	(Standard)
DISKDEF	.LIB	(Standard)
BOOTGEN	.COM	(Option)

Alle, die mit (Option) gekennzeichneten Programme und Dateien sind speziell für das Genie I/II geschrieben und nur auf diesem funktionsfähig. Die Übertragung auf andere CP/M - Computer ist entweder gänzlich unmöglich, mit Sicherheit aber völlig sinnlos. Alle diese Programme oder Dateien sind in diesem Handbuch beschrieben.

Die Beschreibung der mit (Standard) gekennzeichneten Programme werden Sie in diesem Handbuch vergeblich suchen. Ich muß Sie diesbezüglich auf zusätzlich gelieferte Handbücher verweisen. Wir empfehlen besonders "Vom Umgang mit CP/M" von Pohl.

COPY .COM

Das Programm COPY.COM ermöglicht es Ihnen Programme und Dateien von maximal ca. 48 K Bytes Länge mit einem Laufwerk auf verschiedenen Disketten zu kopieren. Die Eingabe der Zeichen \* und ? in Filenamen, also die Benutzung von Wildcards, ist nicht erlaubt.

Die Syntax des Befehls gleicht dem des PIP Programms. Zusätzlich können Sie dabei für die Quell- und Zieldiskette unterschiedliche Pdriveparameter angeben. Beispiele:

COPY B:=A:MCP.SYS                   Kopiert die Datei MCP.SYS von Diskette A nach Diskette B.

COPY A:=A:MCP.SYS                   Kopiert die Datei MCP.SYS in sich selbst. Nach der Ausführung dieses Befehls hat sich auf der Diskette in Laufwerk A nichts geändert. Sie werden nur dann zu einem Diskettenwechsel, für Quell- und Zieldiskette aufgefordert, wenn Sie mindestens einen Pdriveparametersatz angeben.

COPY A:=A:PIP.COM S40 DD,OSBORNE                   Dieser Befehl kopiert z.B. die Datei PIP.COM von der Quelldiskette in Laufwerk A, auf die Zieldiskette in Laufwerk A. Zusätzlich werden jedoch noch die Pdriveparameter OSBORNE für die Quelldiskette und S40 DD für die Zieldiskette angenommen, auch dann, wenn die eigentliche Systemdiskette die Parameter S40 besitzt. Außerdem werden Sie hier zu einem Diskettenwechsel aufgefordert.

Diese Syntax können Sie in allen Formen und Farben auf jedes Laufwerk und für jedes Format anwenden. Fehlt bei der Eingabe der Pdrivedatensatz für die Quell- und/oder Zieldiskette, so werden die Parameter der Systemdiskette für die verschiedenen Laufwerke angenommen.

Optionen wie im PIP Programm sind hier noch nicht implementiert.

Das Kopieren zwischen den verschiedenen Usernummern einer oder mehrerer Disketten ist in Vorbereitung.

Die Bildschirmroutine

Die Bildschirmroutine versteht insgesamt 11 Steuerzeichen. Mit Hilfe dieser Steuerzeichen kann man alle nötigen Funktionen erstellen. Nun die Steuerzeichen:

08 hex	( 8 dez)	:Cursor eine Stelle nach links
0A hex	(10 dez)	:Cursor eine Stelle nach unten (LF)
0B hex	(11 dez)	:Cursor eine Stelle nach oben
0C hex	(12 dez)	:Cursor eine Stelle nach rechts
0D hex	(13 dez)	:Cursor zum Anfang der Zeile (CR)
0E hex	(14 dez)	:Cursor an
0F hex	(15 dez)	:Cursor aus
18 hex	(24 dez)	:Löschen bis zum Ende der Zeile ab der Cursorposition
19 hex	(25 dez)	:wie 18 hex (24 dez), nur wird bis zum Ende des Bildschirms gelöscht
1A hex	(26 dez)	:Löschen des Bildschirms und Cursor nach links oben stellen
1E hex	(30 dez)	:Cursor nach links oben stellen ohne löschen von Zeichen

Zusätzlich existieren noch zwei Escape-Steuersequenzen zur X,Y Manipulation des Cursors.

ESC	=	Y+20H	X+20H	Mit dieser vierstelligen Esc-
1BH	3DH	20H bis	20H bis	Steuersequenz ist es möglich
		2FH	5FH	den Cursor beliebig auf dem
				Bildschirm zu positionieren.

Die Zeichenfolge 1BH 3DH 20H 20H stellt den Cursor dabei in die linke obere Ecke des Bildschirms. Nach der Sequenz 1BH 3DH 2FH 5FH steht der Cursor rechts unten.

ESC	?	Mit dieser zweistelligen ESC
1BH	3FH	Steuersequenz können Sie
		die aktuelle Position des

Cursors auf dem Bildschirm abfragen. Nach der Ausgabe des Fragezeichens über die BDOS Funktionen 2, 6 und 9 befindet sich nach dem Rücksprung aus der entsprechenden Funktion die Position im Registerpaar DE des Prozessors. Im D Register befindet sich die Y Position mit Y+20H und im E Register die X Position mit X+20H. Das heißt, mit den Registerinhalten 2020H in DE befindet sich der Cursor links oben am Bildschirm, mit 2F5FH dagegen befindet er sich rechts unten. Durch diese ESC-Steuersequenz werden weder Zeichen auf den Bildschirm gebracht, noch wird der Cursor bewegt.

Wenn Steuerzeichen zum Bildschirm ausgegeben werden, die nicht mit den oben genannten identisch sind, so werden sie einfach ignoriert. Es passiert dann schlicht nichts.

Alle ASCII-Zeichen, deren Code kleiner ist als 20 hex (32 dez) sind Steuerzeichen. 20 hex (32 dez) selbst ist das Leerzeichen, auch als SPACE bekannt, also kein Steuerzeichen.

Eine Besonderheit sei hier noch erwähnt: Stößt die Bildschirmroutine im Fluß der Zeichen auf unbekannte, zwei Byte lange ESC-Steuersequenzen, so werden auch dieses ignoriert und nicht ausgegeben. Dadurch werden diese ESC-Steuersequenzen ausgefiltert, die bei anderen CP/M Computern zu Reaktionen führen können. Sollten Sie also ein Programm, das solche Steuersequenzen benutzt, auf dem Genie I/II laufen lassen, so bleibt der Bildschirm durch das Ausfiltern trotzdem einigermaßen übersichtlich. In jedem Fall müssen Sie jedoch das Programm an den entsprechenden Stellen ändern, also anpassen.

## Die Tastaturroutine

Vorweg sei bemerkt, daß die Tastatur des Genie I/II für ein CP/M-System eigentlich zu wenig Tasten besitzt. Es fehlt zum Beispiel eine getrennte ctrl-Taste. Ebenso fehlen die Tasten für die deutschen Umlaute, oder für die Entsprechungen des amerikanischen Zeichensatzes. Deshalb besitzt die Tastaturroutine des BIOS 2.1 teilweise eine Vierfachbelegung der Tasten. Das braucht Sie jedoch nicht abzuschrecken, es ist lediglich etwas gewöhnungsbedürftig.

Damit ein schnelles Tippen des Benutzers nicht behindert wird, besitzt die Tastaturroutine eine Einzeltastenentprellung. Beim ersten Abfragen der Tastatur werden alle Tasten berücksichtigt. Bei den meisten Tastaturroutinen folgt jetzt die Totzeit, die nötig ist, damit keine Tasten durch Prellen mehrfach erkannt werden. Diese Totzeit liegt in der Größenordnung 10 bis 50 Millisekunden. Da liegt die Schwierigkeit: wartet man zu lange, werden schnell hintereinander gedrückte Tasten nicht erkannt, wartet man zu kurz, prellt die Tastatur.

Aus diesem Grund werden die Tasten im BIOS 2.1 einzeln entprellt. Nachdem eine Taste erkannt und dekodiert ist, wird sofort wieder die Tastatur abgefragt. Nur die eine gerade dekodierte Taste wird nicht beachtet. Während der Totzeit wird dann nur die eine Taste nicht abgefragt, wohl aber alle anderen. Der zeitliche Abstand der dann noch zwischen dem Drücken zweier verschiedener Tasten liegen muß, ist die Zeit, die die Tastaturroutine benötigt, um die Tastaturmatrix zweimal abzufragen. Sie beträgt ca. 150 - 200 Microsekunden.

Dem Hochgeschwindigkeitstipper sind also mit dieser Tastaturroutine keine Grenzen mehr gesetzt!

## Die Tastaturbelegung

An dieser Stelle treffen wir folgende Abmachung:  
 Wenn ein Zeichen durch " " eingerahmt ist, ist nicht der Code des Zeichens, sondern die Taste auf der Tastatur gemeint.

Beispiel:     A     bedeutet den Code 41 Hex (65 dez)  
               "A"   bedeutet das Drücken der zweiten Taste von links in der dritten Tastenreihe von oben

Wie schon gesagt, besitzt die Tastatur vier Ebenen:

- a) die Ebene der Kleinbuchstaben
- b) die Ebene der Großbuchstaben
- c) die Ebene der Controlzeichen
- d) die Ebene der Funktionstasten

Im Normalzustand liegt in der Tastatur die Ebene der Kleinbuchstaben vor. Die Großbuchstaben erreichen Sie durch Drücken der SHIFT-Taste. Die Controltaste (wird im folgenden durch ctrl abgekürzt) ist die Abwärtspfeiltaste. Durch Drücken dieser Taste liegt die Ebene der Controlzeichen vor. Sie liegen dann auf den Tasten "0", "A"-"Z" und "5"-"9". Dadurch sind alle 32 Steuerzeichen zugänglich. In der Regel finden unter CP/M jedoch nur die Steuerzeichen ctrl "A"-"Z" Verwendung. Die Ebene der Funktionstasten erreicht man wie folgt: Drücken Sie zuerst die SHIFT-Taste. Halten Sie diese fest und betätigen Sie gleichzeitig die SPACE-Taste (die große ganz unten). Halten Sie diese ebenfalls fest. Diese Tastenkombination wird im Folgenden durch fkt abgekürzt. Nun können Sie unter den Tasten "0" und "A"-"Z" die frei programmierbaren Funktionstasten erreichen. Näheres siehe unter FUNKTION.SYS in diesem Handbuch.

Die Tastatur besitzt drei Umschaltmöglichkeiten:

- ctrl "1" :Damit tauschen Sie die Kleinbuchstaben mit den Großbuchstaben. Danach erreichen Sie die Kleinbuchstaben mit SHIFT. Ein weiteres ctrl "1" macht diese Operation rückgängig
- ctrl "2" :Damit tauschen Sie in allen vier Ebenen die Taste "Y" und "Z" aus. Ein nochmaliges ctrl "2" macht auch diese Operation rückgängig.
- ctrl "3" :Nach dieser Tastenkombination liegt auf der Taste ";"        das ö bzw. Ö  
 auf der Taste "@"        das ü bzw. Ü  
 auf der Taste "CLEAR" das ä bzw. Ä und  
 auf der Taste ":"        das ß bzw. \*  
 Nach einem zweiten Drücken von ctrl "3" sind die Umlaute wieder verschwunden und die normale Tastaturbelegung taucht wieder auf.

Zur Voreinstellung der Umschaltmöglichkeiten siehe unter SYSTEM.COM in diesem Handbuch.

Die Besonderheiten der Tastaturbelegung:

An dieser Stelle sollen nur die Tastenbelegungen beschrieben werden, die sich nicht schon durch die Beschriftung der Tasten von selbst erklären. Wie oben beschrieben können Sie die Tastatur durch ctrl "3" mit Umlauten belegen. Dadurch verschwinden die Zeichen: ; + @ \ CLEAR :

Diese Zeichen sind deshalb immer noch an einer zweiten Stelle innerhalb der Tastaturbelegung zu finden:

```

;      liegt auf fkt  ";"
+      liegt auf ctrl ";"
@      liegt auf fkt  "@"
\      liegt auf ctrl "@"
CLEAR  liegt aut ctrl "CLEAR"
:      liegt auf SHIFT "0"

```

Diese Zeichen liegen auch dann an den oben beschriebenen Stellen, wenn die Umlaute weggeschaltet sind, also diese Zeichen auch normal zu erreichen sind.

Hier sei erwähnt, daß dieses BIOS eine echte CLEAR-Taste besitzt. Durch das Drücken dieser Taste löscht die Tastaturroutine des BIOS 2.1 den Bildschirm bevor der Rücksprung zum aufrufenden Programm stattfindet. Die CLEAR-Taste selbst besitzt dabei den Code 18 hex (24 dez). Sie erzeugt den gleichen Code wie ctrl "X", nur daß zusätzlich der Bildschirm gelöscht wird.

Die Taste "BREAK" erzeugt den Code 03 hex (3 dez). Darauf reagieren die meisten CP/M Programme mit einem Warmstart des Systems.

Die "NEW LINE" Taste ist mit dem Code 0D hex (13 dez) belegt (ebenso wie ctrl "M").

Die Taste ctrl "Linkspfeil" ist mit dem Code 18 hex (24 dez) belegt (ebenso wie ctrl "X"). Die Ausführung dieses Kommandos führt zum Löschen des aktiven Eingabepuffers (gleich wie SHIFT "Linkspfeil" im NEWDOS 80 V 2.0 oder im Level II Basic)

## Nun zur Belegung der Pfeiltasten:

```

"Hoch Pfeil" : 1B hex (27 dez) ESC
"Abwärts Pfeil" : Controltaste
"Links Pfeil" : 08 hex ( 8 dez) gleich wie ctrl "H"
"Rechts Pfeil" : 09 hex ( 9 dez) gleich wie ctrl "I"

SHIFT "Hoch Pfeil" : 0B hex (11 dez) gleich wie ctrl "K"
SHIFT "Abwärts Pfeil" : 0A hex (10 dez) gleich wie ctrl "J"
SHIFT "Links Pfeil" : 08 hex ( 8 dez) gleich wie ctrl "H"
SHIFT "Rechts Pfeil" : 09 hex ( 9 dez) gleich wie ctrl "I"

```

Aus der Zusammenstellung ist zu ersehen, daß unter SHIFT "Pfeiltasten" die normalen Cursorsteuerzeichen zu erreichen sind.

Tabelle der ASCII Zeichen und der TastenbelegungControl-Zeichen

Code hex dez	Zeichen	Erreichbarkeit des Zeichens bzw. Codes
00	0	ctrl "0"
01	1	ctrl "A"
02	2	ctrl "B"
03	3	ctrl "C" "BREAK"
04	4	ctrl "D"
05	5	ctrl "E"
06	6	ctrl "F"
07	7	ctrl "G"
08	8	ctrl "H" "Linkspfeil"
09	9	ctrl "I" "Rechtspfeil"
0A	10	ctrl "J" SHIFT "Abwärtspfeil"
0B	11	ctrl "K" SHIFT "Hochpfeil"
0C	12	ctrl "L"
0D	13	ctrl "M" "NEW LINE"
0E	14	ctrl "N"
0F	15	ctrl "O"
10	16	ctrl "P"
11	17	ctrl "Q"
12	18	ctrl "R"
13	19	ctrl "S"
14	20	ctrl "T"
15	21	ctrl "U"
16	22	ctrl "V"
17	23	ctrl "W"
18	24	ctrl "X" ctrl "Linkspfeil"; ctrl "CLEAR" ("CLEAR")
19	25	ctrl "Y"
1A	26	ctrl "Z"
1B	27	ctrl "5" "Hochpfeil"
1C	28	ctrl "6"
1D	29	ctrl "7"
1E	30	ctrl "8"
1F	31	ctrl "9"

Die eingeklammerten Tastenbelegungen sind nur dann erreichbar, wenn Sie die Tastatur ohne Umlaute betreiben. Ansonsten liegen an diesen Stellen der Tastatur die Umlaute.

Die Zeichen von 20 hex bis 3F hex

Code		Zeichen	Erreichbarkeit des Zeichens bzw. Codes	
hex	dez			
20	32		"SPACE"	
21	33	!	SHIFT "1"	
22	34	"	SHIFT "2"	
23	35	#	SHIFT "3"	
24	36	\$	SHIFT "4"	
25	37	%	SHIFT "5"	
26	38	&	SHIFT "6"	
27	39	'	SHIFT "7"	
28	40	(	SHIFT "8"	
29	41	)	SHIFT "9"	
2A	42	*	SHIFT ":",	
2B	43	+	(SHIFT ";")	ctrl ";"
2C	44	,	","	
2D	45	-	"-"	
2E	46	.	"."	
2F	47	/	"/"	
30	48	0	"0"	
31	49	1	"1"	
32	50	2	"2"	
33	51	3	"3"	
34	52	4	"4"	
35	53	5	"5"	
36	54	6	"6"	
37	55	7	"7"	
38	56	8	"8"	
39	57	9	"9"	
3A	58	:	(": ")	SHIFT "0"
3B	59	;	("; ")	fkt ";"
3C	60	<	SHIFT ", "	
3D	61	=	SHIFT "- "	
3E	62	>	SHIFT ". "	
3F	63	?	SHIFT "/ "	

Die eingeklammerten Tastenbelegungen sind nur dann erreichbar, wenn Sie die Tastatur ohne Umlaute betreiben.

Die Zeichen von 40 hex bis 5F hex

Code		Zeichen	Erreichbarkeit des Zeichens bzw. Codes
hex	dez		

40	64	@	("@") fkt "@"
41	65	A	* SHIFT "A"
42	66	B	* SHIFT "B"
43	67	C	* SHIFT "C"
44	68	D	* SHIFT "D"
45	69	E	* SHIFT "E"
46	70	F	* SHIFT "F"
47	71	G	* SHIFT "G"
48	72	H	* SHIFT "H"
49	73	I	* SHIFT "I"
4A	74	J	* SHIFT "J"
4B	75	K	* SHIFT "K"
4C	76	L	* SHIFT "L"
4D	77	M	* SHIFT "M"
4E	78	N	* SHIFT "N"
4F	79	O	* SHIFT "O"
50	80	P	* SHIFT "P"
51	81	Q	* SHIFT "Q"
52	82	R	* SHIFT "R"
53	83	S	* SHIFT "S"
54	84	T	* SHIFT "T"
55	85	U	* SHIFT "U"
56	86	V	* SHIFT "V"
57	87	W	* SHIFT "W"
58	88	X	* SHIFT "X"
59	89	Y	* SHIFT "Y"
5A	90	Z	* SHIFT "Z"
5B	91	Ä	* <u>SHIFT "CLEAR"</u>
5C	92	Ö	* <u>SHIFT ";</u>
5D	93	Ü	* <u>SHIFT "@"</u>
5E	94		ctrl "-"
5F	95		ctrl "."

Die eingeklammerten Tastenbelegungen sind nur dann erreichbar, wenn Sie die Tastatur ohne Umlaute betreiben.

Die unterstrichenen Tastenbelegungen sind nur dann erreichbar, wenn Sie die Tastatur mit Umlauten betreiben.

Die mit \* gekennzeichneten Zeichen können mit ctrl "i" mit der Ebene der Kleinbuchstaben getauscht werden.

Sollten Sie die Zeichen "eckige Klammer auf" und "eckige Klammer zu" benötigen, zum Beispiel beim PIP Programm um Optionen anzugeben, so finden Sie diese unter Ä bzw. Ü (Großbuchstaben). Die Umlaute sehen zwar nicht wie eckige Klammern aus, die Codes der Zeichen sind jedoch identisch!

Die Zeichen von 60 hex bis 7F hex

Code hex dez	Zeichen	Erreichbarkeit des Zeichens bzw. Codes
-----------------	---------	--

60 96	`	(SHIFT "a") ctrl "a"
61 97	a	* "A"
62 98	b	* "B"
63 99	c	* "C"
64 100	d	* "D"
65 101	e	* "E"
66 102	f	* "F"
67 103	g	* "G"
68 104	h	* "H"
69 105	i	* "I"
6A 106	j	* "J"
6B 107	k	* "K"
6C 108	l	* "L"
6D 109	m	* "M"
6E 110	n	* "N"
6F 111	o	* "O"
70 112	p	* "P"
71 113	q	* "Q"
72 114	r	* "R"
73 115	s	* "S"
74 116	t	* "T"
75 117	u	* "U"
76 118	v	* "V"
77 119	w	* "W"
78 120	x	* "X"
79 121	y	* "Y"
7A 122	z	* "Z"
7B 123	ä	* <u>"CLEAR"</u>
7C 124	ö	* <u>";"</u>
7D 125	ü	* <u>"@"</u>
7E 126	ß	* <u>";"</u>
7F 127	Del	ctrl "/"

Die eingeklammerten Tastenbelegungen sind nur dann erreichbar, wenn Sie die Tastatur ohne Umlaute betreiben.

Die unterstrichenen Tastenbelegungen sind nur dann erreichbar, wenn Sie die Tastatur mit Umlauten betreiben.

Die mit \* gekennzeichneten Zeichen können mit ctrl "1" mit der Ebene der Großbuchstaben getauscht werden.

MCP .SYS

Es gibt zwei Sorten von CP/M Disketten, die mit Betriebssystem und die ohne, also reine Datendisketten. Das Betriebssystem ist aufgeteilt in drei Programmteile:

dem BIOS (Basic Input Output System)  
dem BDOS (Basic Disk Operating System)  
und dem CCP (Console Command Prozessor)

Diese drei Programmteile stehen normalerweise auf den Systemspuren einer CP/M Diskette. Beim Genie I/II CP/M liegen diese drei Programmteile jedoch im normalen Benutzerbereich der jeweiligen Diskette. Das BIOS und das BDOS sind in der Datei SYS.SYS und der CCP in der Datei MCP.SYS abgespeichert. Auf den ersten drei Spuren (Spur 0, 1 und 2) Ihrer CP/M Diskette liegt lediglich ein intelligenter Booter, der das eigentliche Betriebssystem vom Benutzerbereich einliebt. Bei einem Neustart des Systems (Kalt- oder Warmstart) lädt das BIOS unter BDOS-Kontrolle den CCP ein.

Die Datei MCP.SYS muß immer auf der Diskette im Laufwerk A enthalten sein, sonst wird der nächste Kalt- oder Warmstart des Systems zum Fehlstart. Das BIOS meldet sich dann mit "kein MCP auf Laufwerk A" und stellt den Betrieb ein. Wenn Sie diese Meldung bekommen, hilft nur die Resettaste oder der Ein- /Ausschalter.

Dadurch, daß der CCP im File MCP.SYS vom CP/M erreicht werden kann, ist der Benutzer in der Lage diesen zu bearbeiten und sehr einfach durch neuere Console Command Prozessoren zu ersetzen. Man braucht den neuen "CCP" entsprechend dem neuen Arbeitsbereich, also direkt unter dem BDOS (Startadresse D607 hex), zu verschieben. Abgespeichert werden muß der neue "CCP" wie ein COM-File. Da das File MCP.SYS bei einem Neustart des Systems vom BIOS eingeladen wird, muß dieses natürlich wissen, wohin der Inhalt des Files MCP.SYS eingeladen werden soll. Dazu muß dem neuen "CCP" ein Record von 128 Bytes Länge vorangestellt werden. Davon sind nur die ersten vier Bytes interessant. Die ersten zwei (Low, High) geben die Anfangsadresse des neuen "CCP's" wieder, die zweiten zwei (Low, High) die Anzahl der Bytes, aus denen der neue "CCP" besteht. Die nächsten 124 Bytes dieses Records sind unbenutzt. Damit besitzt das BIOS alle Informationen, um den CCP richtig einzuladen. Danach springt das BIOS zur Anfangsadresse des CCP's, die zugleich Startadresse sein muß.

Nachdem Sie den neuen CCP so vorbereitet haben, brauchen Sie nur noch das File MCP.SYS umzubenennen und dem neuen "CCP" den Filenamen MCP.SYS zu geben. Nach dem nächsten Kalt- oder Warmstart fahren Sie dann den neuen "CCP".

SYS .SYS

In der Datei SYS.SYS schließlich sind das BDOS und das BIOS des Betriebssystems abgespeichert. Diese Datei wird bei jedem Kaltstart von dem intelligenten Booter, der auf den Systemspuren steht, eingeladen. In dieser Datei werden zusätzlich die Systemparameter (siehe unter SYSTEM.COM in diesem Handbuch) und die Pdriveparameter der Laufwerke B bis D abgelegt. Im Prinzip besteht diese Datei also aus drei Teilen:

1. den Systemparametern
2. den Pdriveparametern
3. dem Betriebssystemteilen BDOS und BIOS

Der erste Teile belegt dabei genau einen CP/M Sektor (128 Bytes) auf der Diskette. Der zweite Teil belegt dagegen genau 3 solcher Sektoren, entsprechend den drei Parametersätzen. Der Parametersatz für Laufwerk A befindet sich innerhalb des Booters auf den Systemspuren und wird von diesem beim Kaltstart zum Betriebssystem übergeben.

Die Lage des Betriebssystems innerhalb des normalen Benutzerbereiches der Diskette hat auch hier den Vorteil, daß man jederzeit problemlos ans Betriebssystem herrankommt. Außerdem sind später eventuell notwendige Updates denkbar einfach, Sie brauchen dann nur die neue Datei SYS.SYS auf Ihren Disketten abzuspeichern und schon ist das neuere Betriebssystem installiert.

Die Datei SYS.SYS muß genau wie die Datei MCP.SYS auf der Diskette in Laufwerk A enthalten sein, sonst bekommen Sie beim nächsten Bootversuch die Fehlermeldung: "Datei SYS.SYS nicht gefunden!". Das sollte normalerweise auch einleuchten, denn ohne die Datei SYS.SYS besitzt eine Genie I/II CP/M-Diskette kein CP/M.

FUNKTION.SYS

Unter diesem Filenamen sind auf der Diskette 26 frei programmierbare Funktionstasten gespeichert. Zusätzlich befindet sich an erster Stelle, also noch vor den eigentlichen Funktionstasten, die Definition des Autostartkommandos. Dieses Autostartkommando kann auf zweierlei Weisen ausgelöst werden. Einmal durch einen Kaltstart, nicht durch einen Warmstart, aber durch fkt "0"

Mit fkt ist folgende Tastenkombination gemeint: Betätigen Sie zuerst die SHIFT-Taste, halten diese fest und drücken zusätzlich die SPACE-Taste und halten auch diese fest. fkt "0" wäre dann: SHIFT (halten) SPACE (halten) "0"

Insgesamt stehen Ihnen die Funktionen fkt "A"-"Z" zur Verfügung.

Das File FUNKTION.SYS muß immer genau 1K Bytes lang sein, also 1024 Bytes, egal wie viele Tasten Sie definieren. Dieses File wird, genau wie MCP.SYS auch, bei jedem Kalt- oder Warmstart des Systems eingeladen. Die Funktionstasten werden dann im Bereich des BIOS abgelegt (genau F500 hex bis F8FF hex). Organisiert ist das File wie folgt:

- Das erste Byte dieses File muß immer 00 hex sein!
- Die erste Funktionstaste wird durch fkt "0" ausgelöst und representiert den Autostartbefehl.
- Das Ende jeder Funktionstaste symbolisiert ein 00 hex.
- Die zweite Funktionstaste korrespondiert mit fkt "A".
- Das Ende symbolisiert wieder ein 00 hex.

Sie erkennen, daß in dem File FUNKTION.SYS insgesamt 28 mal 00 hex auftauchen muß. Stehen diese Nullen direkt hintereinander, so sind alle Funktionstasten leer. Fehlt bei einem Kalt- oder Warmstart das File FUNKTION.SYS auf der Diskette in Laufwerk A, so wird der Funktionstastenbuffer des BIOS mit 28 Nullen gelöscht.

Nun folgt als Beispiel der Anfang eines möglichen File FUNKTION.SYS und dessen hexadezimale Auflösung:

```
PIP.COM\M BASI
C.COM\M DELETE
\ .....
```

```
00 50 49 50 2E 43 4F 4D 60 4D 00 00 42 41 53 49
43 2E 43 4F 4D 60 4D 00 00 44 45 4C 45 54 45 60
60 00 .. .. .. .. .. .. .. .. .. .. .. ..
```

In diesem Beispiel ist das Autokommando als PIP.COM mit automatischem <NEW LINE> definiert. Das heißt, es entfällt nach dem Auslösen dieser Funktion das Drücken der NEW LINE Taste. Sie werden aber sagen, daß in der Tabelle nirgendwo das Zeichen 0D hex (13 dez) gleich <NEW LINE> zu finden ist. Da haben Sie recht. Das BIOS berechnet den Code wie folgt: Stößt das BIOS während der Übertragung der Funktionstaste auf das Zeichen 60 hex (96 dez), so subtrahiert das BIOS vom nächsten Zeichen den Wert 40 hex (64 dez). Die Zeichen 60 hex (96 dez) selbst werden aber nicht ausgegeben.

In der oberen Tabelle finden Sie nach den ASCII Zeichen PIP.COM die Zeichenfolge 60 4D 00 hex. Das BIOS berechnet dann aus 60 4D den Wert 0D hex (13 dez), welcher dann übertragen wird. Dann folgt die 00. Daran sieht das BIOS, daß die Funktionstaste hier zu Ende ist. Die 00 selbst wird nicht übertragen.

An den Stellen, wo zwei Nullen hintereinander stehen, ist die Definition einer Funktionstaste übersprungen worden. Im obigen Beispiel finden Sie diesen Fall zweimal. Nicht definiert sind hier die Funktionstasten fkt "A" und fkt "C".

Am Ende der Funktionstaste fkt "D" finden Sie die Zeichenfolge 60 60 00 hex. Die beiden 60 hex führen zur Berechnung des Wertes 20 hex (32 dez), welches das Leerzeichen darstellt. Danach endet die Definition der Funktionstaste. Beim Ausführen wird deshalb folgender Text übertragen: DELETE. In Microsoft Basic funktioniert dieser Befehl nur dann, wenn zwischen dem Befehlsword und den Zeilennummern ein Leerzeichen steht, was hier gleich mit definiert worden ist. Sie werden fragen, warum kann man das Zeichen 20 hex (32 dez) nicht gleich im Klartext in den File FUNKTION.SYS einbauen? Das BIOS selbst hat nichts dagegen, nur der String-Input-Befehl des Basic Interpreters schneidet alle Leerzeichen am Ende der Eingabe ab. Und das Programm FUNKTION.MAS auf Ihrer CP/M Diskette, mit dem Sie ein Funktionstastenfile editieren und neu aufbauen können ist ein Basicprogramm.

Das ` ,daß oben öfter gebraucht wird (z.B. bei PIP.COM`M), ist nicht das ` über der Sieben sondern das ` über dem @.

## FUNKTION.MAS

Mit dem Programm FUNKTION.MAS sind Sie in der Lage einen File, das den Regeln unter FUNKTION.SYS genügt, zu editieren oder neu zu erstellen. Mit dem Funktionstastensatz, der mit der Systemdiskette geliefert wird, können Sie das Programm mit fkt "F" anstarten. Die Funktionstaste fkt "F" bewirkt ein automatisches Einladen des Steuerprogramms MASTER.COM, das Einladen des Programms FUNKTION.MAS und dessen Anstartung. Allein dieses Beispiel zeigt schon, wie leistungsfähig die Funktionstasten den Benutzer unterstützen können.

Nach der Ausführung der Funktiostaste meldet sich das Programm FUNKTION.MAS mit:

Soll ein alter File bearbeitet werden (J/N) ?

Wenn Sie an dieser Stelle N (wie nein) eingeben, erzeugt das Basicprogramm ein Leerfile, das Sie direkt anschließend programmieren können.

Wenn Sie hier mit J (wie ja) antworten, fragt das Programm mit:

### Alter File Namen ?

nach einem Filenamen eines bereits existierenden Funktions-  
tastensatzes. Sie können also zum Beispiel mit FUNKTION.SYS  
antworten. Das Programm lädt dann dieses File von der Diskette  
in den Ramspeicherbereich des Computers. Nachdem Ihr Computer  
das File eingeladen hat, kann es bis zu einer Minute dauern,  
bis er sich wieder meldet, da der eingeladene Funktionstasten-  
satz mit Hilfe von nicht ganz einfachen Stringoperationen in  
seine 27 Bestandteile zerlegt wird. Es dauert um so länger, je  
mehr und komplexer Sie die Funktionstasten programmiert haben.

Ob Sie nun ja oder nein antworten, das Programm meldet sich  
als nächstes im Grundmodus mit:

### Funktionstaste ?

Hier haben Sie drei verschiedene Möglichkeiten:

- a) Geben Sie / ein. Nun listet Ihnen das Programm die  
Inhalte aller Funktionstasten, einschließlich des Auto-  
Kommandos. Durch nochmaliges <NEW LINE> landen Sie wieder  
im Grundmodus.
- b) Geben Sie "0" oder "A"- "Z" ein. Dann können Sie die Inhalte  
der entsprechenden Funktionstasten programmieren.  
Beispiel:

#### Funktionstaste ? A

```
A:DIR`M
?
```

Wenn Sie hier nur <NEW LINE> drücken, springt das Programm  
ohne Änderung der Funktionstaste zum Grundmodus zurück. Das  
ist sehr nützlich, falls man sich mal vertippt. Ansonsten  
geben Sie hier einfach den neuen Inhalt der Funktionstaste  
ein.

Beispiel:

#### Funktionstaste ? A

```
A:DIR`M
? PIP.COM`M
```

Damit programmieren Sie fkt "A" mit dem Einladen und An-  
starten des Programms PIP.COM, wobei die Zeichenfolge `M  
die <NEW LINE> Taste symbolisiert (siehe auch unter  
FUNKTION.SYS in diesem Handbuch). Nach der Eingabe der  
neuen Funktion springt das Programm wieder in den Grund-  
modus zurück. Lassen Sie sich an dieser Stelle noch einmal  
alle Funktionstasten mit / listen und Sie werden sehen,  
daß unter fkt "A" jetzt "PIP.COM`M" steht.

- c) Als letzten Befehl im Grundmodus des Programms FUNKTION.MAS haben Sie die Möglichkeit \* einzugeben. Damit wird das Editieren oder Neuprogrammieren der Funktionstasten abgeschlossen und das Programm meldet sich mit:

Einen anderen File Namen (J/N) ?

Wenn Sie keinen alten Funktionstastensatz editiert, sondern einen neuen erstellt haben, können Sie hier nur mit J (wie ja) antworten. Das Programm fragt dann mit:

Neuer File Name ?

nach einem Filenamem, unter dem die neu programmierten Funktionstasten abgespeichert werden sollen.

Wenn Sie allerdings einen alten Funktionstastensatz editiert haben, zum Beispiel FUNKTION.SYS, so können Sie die Frage

Einen anderen File Namen (J/N) ?

auch mit N (wie nein) beantworten. Dann werden die neu überarbeiteten Funktionstasten wieder in der Datei abgespeichert, von der sie geladen wurden. Sie können allerdings auch einen alten Funktionstastensatz trotzdem unter einem neuen Filenamem abspeichern.

An dieser Stelle müssen Sie wieder bis zu einer Minute warten, bis das Programm die 27 Funktionstasten wieder zu einer Einheit zusammengerechnet hat und abspeichern kann.

Nach dem Abspeichern übergibt das Steuerprogramm MASTER.COM die Kontrolle automatisch an den jeweils aktiven CCP zurück.

Wie Sie vielleicht bemerkt haben, können Sie verschiedene Funktionstastensätze unter verschiedenen Filenamem auf der (System-) Diskette ablegen. Im Laufe der Zeit werden Sie sich sicher für jeden Anwendungszweck einen optimal angepaßten Funktionstastensatz zulegen. Einen zum Arbeiten unter BASIC, mit den BASIC Befehlswörtern, einen zum Arbeiten unter Assembler, mit den 8080 oder 280 Mnemonics, einen für Arbeiten mit dem CCP, mit dessen Befehlen oder festen Befehlsfolgen. Aktivieren können Sie einen neuen Funktionstastensatz, zum Beispiel mit:

```
PIP FUNKTION.SYS=BASIC.FKT
```

Mit diesem Befehl wird der Inhalt von BASIC.FKT, den Sie sich selber erstellen können, in das File FUNKTION.SYS kopiert. Da nach der Ausführung des Kopierbefehls ein Warmstart des CP/M's erfolgt, sind die Funktionstastensatz von BASIC.FKT sofort aktiv.

Ja, Sie können sogar in jedem Funktionstastensatz den PIP Befehl zum Wechseln auf einen oder mehrere andere Funktionstastensätze programmieren.

PDRIVE .SYS

In der Datei PDRIVE.SYS sind die Pdriveparameter zwischengespeichert. Ein Pdrivedatensatz besteht immer aus 128 Bytes. Er besitzt also genau die Länge eines "CP/M Sektors". In jedem dieser Datensätze sind alle Informationen enthalten, um das jeweilige Laufwerk hardwaremäßig zu beschreiben. Zusätzlich enthält ein solcher Datensatz alles über die logische Organisation einer bestimmten Diskette, die dann auf dem oben hardwaremäßig beschriebenen Laufwerk betrieben werden soll.

Nun zum Aufbau eines solchen Datensatzes. Die Bytes zwei, drei und vier eines jeden Datensatzes sind unbenutzt. Das erste und das fünfte Byte sind BIOS Bytes. Hier wird das jeweilige Laufwerk hardwaremäßig beschrieben.

Nun folgt ein 15 Bytes langer String, der den Namen des Datensatzes enthält. Unter diesem Namen finden die verschiedenen Programme einen bestimmten Datensatz aus der Masse der anderen heraus. Hier liegt auch der große Vorteil der Pdrivedatensätze: Hat man sich einmal der Mühe unterzogen einen Datensatz für eine bestimmte Diskette zu entwerfen und auszutesten, kann man diesen Datensatz immer wieder unter seinem Namen wiederfinden. Dieser Name muß mindestens ein Zeichen lang sein. Er darf alle ASCII-Zeichen und auch Leerzeichen enthalten. Ein Name, der kürzer ist als 15 Bytes, muß nach rechts mit Leerzeichen aufgefüllt werden. Dies erledigt das Programm PDRIVE.MAS automatisch.

Das nächste Byte (Byte 21 dez) muß immer das Zeichen \$ (24 hex 36 dez) sein. An diesem Byte erkennen die verschiedenen Programme, ob sie einen korrekten Datensatz bearbeiten, oder nicht.

Nun folgen 15 Bytes, die die logische Organisation einer Diskette wiedergeben. Diese 15 Bytes werden nur vom BDOS benutzt, nicht vom BIOS. Die Beschreibung dieser Bytes folgt unter PDRIVE.MAS in diesem Handbuch.

Die nächsten 92 Bytes sind für die Sektortabelle bestimmt. Es brauchen nicht alle 92 Bytes belegt zu sein, was eigentlich auch nie vorkommt. Trotzdem sind alle 92 Bytes für die Sektortabelle reserviert.

Die Datei PDRIVE.SYS ist demnach nur eine lose, ungeordnete Aneinanderreihung solcher 128 Bytes langer Datensätze. Die Anzahl der verschiedenen Datensätze bestimmt die Länge der Datei PDRIVE.SYS.

Die Bedeutung der einzelnen Bytes eines Pdrivedatensatzes in Tabellenform:

00 hex - 00 hex	BIOS Byte (erster Teil)
01 hex - 03 hex	unbenutzt
04 hex - 04 hex	BIOS Byte (zweiter Teil)
05 hex - 13 hex	Name des Datensatzes
14 hex - 14 hex	\$ (24 hex 36 dez)
15 hex - 23 hex	logische Organisation der Diskette
24 hex - 7F hex	Sekortabelle der Diskette

In der von uns gelieferten Datei PDRIVE.SYS sind folgende Datensätze enthalten:

0: S40	1: S40 DS
2: S40 DD	3: S40 DSDD
4: S40 DT	5: S40 DSdT
6: S40 DDDT	7: S40 DSDDDT
8: S80	9: S80 DS
10: S80 DD	11: S80 DSDD
12: 40	13: 40 DS
14: 40 DD	15: 40 DSDD
16: 40 DT	17: 40 DSdT
18: 40 DDDT	19: 40 DSDDDT
20: 80	21: 80 DS
22: 80 DD	23: 80 DSDD
24: FORMULA	25: FORMULA DT
26: OSBORNE	27: OSBORNE DT
28: OSBORNE DD	29: OSBORNE DDDT
30: GENIE III 40	31: GENIE III 40 DS
32: GENIE III	33: GENIE III DS
34: IBM 3740	35: OMIKRON
36: ALPHATRONIC	37: 80 DSDDSP

Die Tabelle der Datensätze enthält nur deren Namen, also jeweils die Bytes 05 hex bis 13 hex eines jeden Datensatzes. Diese Namen können, wie oben schon gesagt, beliebig gewählt werden. Trotzdem folgen sie einem Regelsystem, welches es Ihnen erlaubt, den jeweils richtigen Datensatz auszuwählen, ohne immer lange in der oben abgebildeten oder sonstigen Tabellen nachschauen zu müssen.

Die ersten 24 Datensätze (0 bis 23) sind extra für das Genie I/II CP/M entwickelt worden. Sie liefern maximalen Speicherplatz in Verbindung mit optimalen Zugriffszeiten. Das Regelsystem der Datensatznamen bezieht sich vorwiegend auf diese 24 Datensätze.

Das "S" am Anfang eines solchen Namens bedeutet soviel wie System. Das heißt, diese Datensätze sind für Disketten bestimmt, die den drei Spuren langen Booter des Betriebssystems auf den Spuren 0 bis 2 tragen. Das Directory solcher Disketten liegt also auf Spur 3. Die Datensätze ohne "S" am Anfang des Namens sind für reine Datendisketten bestimmt. Auf solchen Disketten liegt das Directory auf Spur 0. Sie können nicht in Laufwerk A betrieben werden.

Nach dem "S" kommt entweder eine "40" oder eine "80" im Datensatznamen. Sie stehen für mit 40 Track (Spuren) oder 80 Track (Spuren) beschriebene Disketten. Also alle Datensätze mit einer "40" im Namen sind für 40 Track Disketten bestimmt, die mit einer "80" für 80 Track Disketten.

Als letztes gibt es die Namenszusätze "DS", "DD" und "DT". Sie signalisieren, daß der Datensatz mit

- "DS" für Double Side formatierte Disketten bestimmt ist,
- "DD" für Double Density formatierte Disketten bestimmt ist und
- "DT" für Double Track betriebene Disketten, das heißt 40 Track Disketten in 80 Track Laufwerken, bestimmt ist.

Treten zwei oder drei dieser Namenszusätze gleichzeitig auf, so gilt die Reihenfolge "DS" vor "DD" vor "DT". Nach dieser grauen Theorie nun drei Beispiele:

- a) Sie haben eine Diskette von der Sie wissen, daß sie ein CP/M System enthält, also muß der richtige Datensatz mit "S" anfangen. Zusätzlich ist es eine 40 Track Diskette, demnach muß der Namen ein "40" enthalten. Und als letztes wurde die Diskette mit Double Density beschrieben, was ein "DD" notwendig werden läßt. Nun brauchen Sie nur noch die einzelnen Namensteile zusammenzustellen und es ergibt sich S40 DD . Sie brauchen also nur noch das Laufwerk, auf dem Sie die Diskette betreiben wollen, mit dem Datensatz, dessen Namen S40 DD heißt, zu programmieren. Dazu siehe unter PD.COM in diesem Handbuch.
- b) Zweites Beispiel: Sie wollen eine 40 Track Diskette mit CP/M System, die doppeltseitig mit Double Density beschrieben wurde, betreiben. Als zusätzliche Schwierigkeit nehmen wir an, Sie besäßen "nur" 80 Track Laufwerke. Als erstes gehört wieder ein "S" wie System zum gesuchten Namen. Als zweites benötigen wir eine "40" wie 40 Track. Ein "DS" wie doppeltseitig und ein "DD" wie Double Density gehören auch dazu. Und zuletzt noch das "DT" wie doppelt Track, weil Sie eine 40 Track Diskette auf ein 80 Track Laufwerk betreiben wollen. Nun hängen Sie wieder alles zusammen und es ergibt sich: S40 DSDDDT . Unter diesem Namen sind dann die Pdriveparameter abgespeichert, die Sie suchen.
- c) Als letztes Beispiel wähle ich eine 80 Track Datendiskette, die einseitig mit Single Density beschrieben wurde. Zu bemerken ist hier, daß Sie zum Lesen einer solchen Diskette auf jeden Fall ein 80 Track Laufwerk benötigen. Nun zum Datensatznamen; ein "S" darf nicht enthalten sein, da es eine reine Datendiskette ist. Ebenfalls treffen hier die Namenszusätze "DS", "DD" und "DT" nicht zu. Der gesuchte Datensatzname heißt in diesem Fall schlicht 80 .

Die oben aufgeführten Beispiele beziehen sich nur auf Disketten, die mit den speziell entwickelten Genie I/II Formaten beschrieben sind, also mit dem Programm FORMAT.COM der Systemdiskette. Für die Datensatznamen zu systemfremden Formaten, also Disketten anderer Computer gilt, daß nur solche Namenszusätze frei wählbar sind, die auch auf den entsprechenden Computern frei wählbar sind. Beispiele: Im Datensatz "FORMULA" ist bereits enthalten, daß die entsprechenden Disketten doppelseitig unter Double Density beschrieben sind. Als frei wählbarer Namenszusatz bleibt nur noch "DT" übrig. Bei den Formaten "OSBORNE" ist zusätzlich der Namenszusatz "DD" zulässig, da der Osborne Computer Disketten unter Single Density wie auch unter Double Density beschreibt. Beim Format "GENIE III" dagegen ist nur der Namenszusatz "DD" unerlaubt, da das Genie III unter CP/M immer nur Double Density fährt, also alle GENIE III Datensätze grundsätzlich Double Density sind.

Natürlich können nicht immer alle Formate benutzt werden. Besitzen Sie zum Beispiel ausschließlich einseitig schreibende Laufwerke, so können Sie keine Formate benutzen, die den Namenszusatz "DS" enthalten. Zusätzlich fallen dann die FORMULA Formate aus, da dieser Computer die Disketten immer doppelseitig beschreibt. Entsprechendes gilt für "DD" und "DT".

Wie Sie sicher schon bemerkt haben, können Sie mit diesem BIOS unter GENIE I/II auch Disketten fremder Computersysteme lesen und schreiben. Sie brauchen nur einen richtig erstellten Pdrivedatensatz. Mitgeliefert bekommen Sie die ausgetesteten Datensätze zum Betreiben von OSBORNE, FORMULA und GENIE III Disketten. Sie können quasi alle CP/M Diskettenformate mit diesem BIOS Ihrem Computer anpassen. Haben Sie beim Programmieren eines bestimmten Datensatzes Probleme, setzen Sie sich mit Ihrem Fachhändler in Verbindung. Zum Anpassen eines bestimmten Diskettenformates benötigen wir folgende Daten und Sachen:

- 1) Eine Diskette, des anzupassenden Computersystems; diese Diskette muß ein CP/M System beinhalten und auf dem anzupassenden Computer bootfähig sein. Auch sollten die Programme STAT.COM und PIP.COM auf der Diskette abgespeichert sein.
- 2) Zusätzlich benötigen wir den Ausdruck (abschreiben tut's auch) der Anweisung "STAT DSK:" , der auf dem anzupassenden Computer angefertigt werden muß.
- 3) Es reicht nicht aus, wenn Sie uns sagen, daß Sie gerne einen Pdrivedatensatz zum Betreiben einer Diskette des CP/M Computers XYZ ungelöst hätten. Wir können nicht die Systemdisketten aller CP/M Computer auf Lager haben.

Mit diesen Daten sind wir normalerweise in der Lage einen Pdrivedatensatz für die entsprechende Diskette zu erstellen.

PDRIVE .MAS

Mit dem Programm PDRIVE.MAS sind Sie in der Lage, ein File mit Pdrivedatensätzen, also zum Beispiel die Datei PDRIVE.SYS, zu editieren oder neu zu erstellen. Mit dem Funktionstastensatz, der auf der Systemdiskette geliefert wird, können Sie das Programm mit fkt "Q" anstarten. Der Funktionsablauf ist analog zu fkt "F" beim Anstarten von FUNKTION.MAS. Das Programm PDRIVE.MAS meldet sich dann als erstes mit:

Wollen Sie einen alten File bearbeiten ?

Hier können Sie analog zu FUNKTION.MAS wieder entweder J (wie ja) oder N (wie nein) antworten. Bei nein wird im Computer eine Leerdatei erzeugt. Bei ja fragt Sie der Computer mit

Alter Filename ?

nach dem Filenamen einer Pdrivedatei. Hier können Sie demnach mit PDRIVE.SYS antworten. Der Inhalt des kompletten Pdrive-datensatzes wird in den Computer eingeladen. Diese Datei darf maximal 100 Datensätze und damit 100 Records lang sein. Mehr kann das Programm PDRIVE.MAS (gleichzeitig) nicht verarbeiten.

Auf jeden Fall meldet sich das Programm PDRIVE.MAS als nächstes im Grundmodus mit:

Befehlseingabe ?

Hier stehen Ihnen sechs Befehle zur Verfügung. Ausgeführt werden diese Befehle durch Eingeben eines Buchstabens (siehe unten) und Drücken der <NEW LINE> Taste. Das Programm führt dann im Dialogstil durch die jeweilige Funktion.

- 1) P Mit diesem Befehl können Sie sich die Datensatznamen der einzelnen Pdrivedatensätze auf dem Bildschirm listen lassen. Da die Gesamtdatei mehr Datensätze enthalten kann, als gleichzeitig auf dem Bildschirm passen, werden sie immer nur in 12er Schritten gelistet. Nach jedem 12er Block müssen Sie einmal <NEW LINE> drücken, um sich durch die Datensätze durchzutasten. Die einzelnen Datensatznamen sind dabei von 0 an aufwärts durchnummeriert.
- 2) L Mit diesem Befehl können Sie Datensätze löschen. Dazu geben Sie nur die Nummer des betreffenden Datensatzes ein. Dieser wird dann gelöscht. Wenn Sie sich danach die Namen der Datensätze noch einmal mit "P" listen lassen, werden Sie feststellen, daß der Name des Datensatzes zwar fehlt, dessen frühere Nummer jedoch noch auftaucht. Sollten Sie versehentlich einen Datensatz löschen, können Sie ihn trotzdem noch mit "E" editieren und seinen Namen neu eingeben. Damit ist der Vorgang des Löschens rückgängig gemacht.

- 3) K Mit diesem Befehl können Sie Datensätze kopieren. Dazu geben Sie lediglich die Nummer des betreffenden Datensatzes ein. Das Programm erzeugt dann einen zusätzlichen Datensatzplatz am Ende der Datei. In diesen wird dann der Datensatz kopiert. Diese Operation führt der Computer vorläufig nur im Arbeitsspeicher aus. Dieser Befehl ist dann sehr nützlich, wenn Sie den ursprünglichen Datensatz, leicht modifiziert, zusätzlich benötigen. Er erspart Ihnen damit das neue Eingeben aller Parameter eines Datensatzes. Vergessen Sie nicht, auch den Namen des Datensatzes zu ändern! Denn auf Grund des Namens findet das Programm PD.COM einen Datensatz innerhalb der Gesamtdatei.
- 4) A Mit diesem Befehl können Sie eine zweite Pdrive-datendatei in den Arbeitsspeicher einladen. Damit koppeln Sie also zwei verschiedene Dateien zu einer. Die Datensätze werden am Ende angehängt und entsprechend numeriert. Achten Sie aber darauf, daß die Anzahl der einzelnen Datensätze die Zahl 100 nicht überschreitet.
- 5) S Mit dem Befehl S verlassen Sie den Grundmodus endgültig. Das Programm fragt Sie jetzt mit:

Einen anderen Filenamen (J/N)?

ob Sie einen neuen Filenamen benutzen wollen, um die Pdrivedatensätze, die sich momentan im Speicher befinden, auf der Diskette abzugelegen. Wenn Sie beim Anstarten des Programms keinen alten Datensatz eingeladen haben, können Sie diese Frage nur mit einem "J" beantworten. Das System fragt Sie dann mit:

Neuer Filenamen?

Nach einem Filenamen, unter dem die neuen Datensätze auf der Diskette verankert werden sollen. Existiert ein entsprechendes File noch nicht, so wird es vom Programm eröffnet. Nach dem Abspeichern der Datensätze springt das Programm wieder zum jeweils aktiven CCP zurück.

- 6) E Mit diesem Befehl können Sie einen beliebigen Pdrivedatensatz, unter Angabe seiner Nummer, editieren. Haben Sie beim Anstarten des Programms keine alte Pdrivedatei eingeladen, bleibt der Bildschirm beim P Befehl natürlich leer, Sie können trotzdem jeden noch nicht existierenden "Datensatz" editieren und damit einen neuen programmieren. Das gilt auch, wenn die Datensätze 0 bis 30 schon belegt sind. Geben Sie beim Editieren einfach die Nummer 31 ein und programmieren Sie einen neuen Datensatz. In beiden Fällen führt Sie das Programm durch insgesamt vier Bildschirmseiten, auf denen Sie Parameter editieren oder programmieren können.

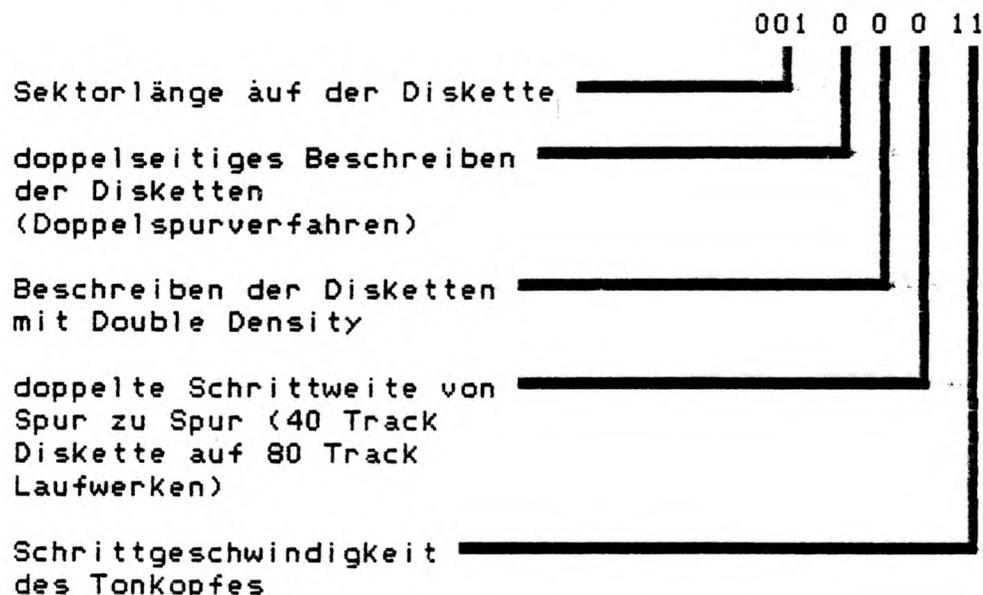
Editieren eines DatensatzesErste Bildschirmseite

über jeder der vier Bildschirmseiten steht der Name des betreffenden Datensatzes. Man weiß also immer, wobei man gerade ist. Auf der ersten Bildschirmseite werden die BIOS-Bytes programmiert. Als erstes wird gefragt, ob der Datensatz für ein 8 Zoll Laufwerk bestimmt ist:

8 Zoll Laufwerk:Dauerläufer \_\_\_\_\_:

Die Frage nach dem "Dauerläufer" bezieht sich auf den Antrieb des entsprechenden Laufwerkes. Läuft der Antriebsmotor immer, so sollten Sie hier "J" eingeben, denn es ergibt sich ein erheblicher Zeitvorteil. Das ist bei (fast) allen 8 Zoll Laufwerken der Fall. Wenn Sie die Einstellung dieser Parameter nicht ändern wollen, geben Sie einfach <NEW LINE> ein.

Der zweite Teil der BIOS Bytes besteht aus insgesamt 8 Einsen und Nullen. Die Bedeutung der einzelnen Bits wird am Beispiel des "Osborne" Datensatzes gezeigt:



Die hardwaremäßige Sektorlänge auf der Diskette stimmt bei Microcomputer CP/M Systemen in den seltensten Fällen mit der Länge eines "CP/M Sektors" überein. Ein "CP/M Sektor" ist bekanntlich immer 128 Bytes lang. Sehr oft werden dann 2, 4 oder 8 solcher 128 Bytes langen Sektoren zusammengefaßt und gleichzeitig in einem Sektor der Diskette untergebracht. Dies ist keine Funktion des eigentlichen CP/M's. Dafür haben die "Sektoren" immer eine Länge von 128 Bytes. Das ist im CP/M intern so festgelegt. Das Blocking und Deblocking erzeugt das jeweilige BIOS. Es sammelt dann beim Schreiben von Sektoren immer 2, 4 oder 8 "CP/M Sektoren" in einem Buffer des BIOS und führt erst dann eine Schreiboperation aus. Umgekehrt liest dann das BIOS einen langen Hardsektor in seinen Buffer ein und gibt ihn dann in 128 Bytes umfassenden Häppchen ans BDOS weiter.

Im BIOS 2.1 können Sie die hardwaremäßige Länge der Sektoren mit den linken drei Bits des BIOS-Byte, wie folgt, selbst bestimmen.

000	128 Bytes pro Hardsektor
001	256 Bytes pro Hardsektor
011	512 Bytes pro Hardsektor
111	1024 Bytes pro Hardsektor

Das vierte Bit von links des BIOS Byte gibt an, ob die Rückseite der Diskette mit dem Doppelspurverfahren angesprochen werden soll, oder nicht.

#### Das Doppelspurverfahren:

Bei diesem Verfahren organisiert das BIOS die Diskette so, daß sie für das CP/M aussieht, als wäre sie einseitig beschrieben und besäße doppelt so viele Spuren. Dabei liegen dann alle Spuren mit geraden Nummern (0,2,4,6...) auf der Vorderseite der Diskette und alle Spuren mit ungeraden Nummern (1,3,5,7...) auf der Rückseite der Diskette. Dieses Verfahren wird zum Beispiel beim FORMULA-Microcomputer angewendet. Bei diesem Format sind auch drei Spuren zu Systemspuren ernannt und damit dem CP/M entzogen, wie bei fast allen Microcomputern mit CP/M Betriebssystem. Beachten Sie dabei bitte, daß die Vorder- und Rückseite der ersten Spur und die Vorderseite der zweiten Spur je einzeln gezählt werden müssen. Dadurch beginnt das Directory bereits auf der Rückseite der zweiten und nicht auf der Vorderseite der vierten Spur.

#### Das Doppelsektorverfahren:

Wenn Sie Disketten mit diesem Verfahren ansteuern, dürfen Sie das vierte Bit von links des BIOS Byte nicht 1 setzen, obwohl Sie die Diskette doppelseitig beschreiben wollen. Die entsprechenden Parameter werden in der Sektortabelle gesetzt. Näheres dazu folgt. Für das CP/M sieht dann eine Diskette aus, als hätte sie pro Spur die doppelte Anzahl von Sektoren als eine gleiche einseitig beschriebene Diskette. Dieses Verfahren wird zum Beispiel beim GENIE III CP/M angewendet, um die Rückseite einer Diskette anzusprechen. Auch hier sind drei Spuren zu Systemspuren ernannt. Dabei belegen sie beim Doppelsektorverfahren wirklich die ersten drei Spuren und nicht eineinhalb wie beim FORMULA. Das Directory liegt dann wie erwartet auf der vierten Spur.

Das fünfte Bit von links des BIOS-Byte gibt an, ob die Diskette mit doppelter Schreibdichte beschrieben werden soll.

Das sechste Bit von links des BIOS-Byte muß gesetzt werden, wenn eine 40 Track Diskette auf einem 80 Track Laufwerk betrieben werden soll. Das BIOS läßt dann den Kopf des Laufwerks von Spur A bis Spur B einfach doppelt so viele Schritte ausführen wie sonst. Hier sei bemerkt, daß das Schreiben von 40 Track Disketten auf 80 Track Laufwerken in der Regel nicht einwandfrei funktioniert. Die betreffenden Stellen können zwar auf dem 80 Track Laufwerk wieder gelesen werden, meistens jedoch nicht mehr auf 40 Track Laufwerken. Sie sollten diese Möglichkeit am besten nur benutzen, um Programme und Daten in den Computer einzulesen.

Die letzten beiden Bits des BIOS Byte, Bit sieben und acht von links, geben die Schrittgeschwindigkeiten an, mit der das Floppyinterface das Laufwerk steuern sollen. Sie bedeuten dann folgendes:

00	5 ms von Spur zu Spur
01	10 ms von Spur zu Spur
10	20 ms von Spur zu Spur
11	40 ms von Spur zu Spur

Wenn Sie das BIOS-Byte nicht ändern wollen, drücken Sie einfach <NEW LINE>. Damit kommen Sie automatisch zur nächsten Bildschirmseite. Wenn Sie jedoch eines der Bits ändern wollen, müssen Sie immer alle acht eingeben, auch solche, die Sie nicht ändern wollen.

### Zweite Bildschirmseite

Auf der zweiten Bildschirmseite erfolgt die Programmierung der logischen Organisation der Diskette. Diese Parameter werden nur vom BDOS benutzt. Es sind 10 verschiedene Parameter, die insgesamt 15 Bytes eines Pdrivedatensatzes belegen. Da diese Parameter miteinander zusammenhängen, reicht die Eingabe von fünf aus. Die restlichen werden vom Programm PDRIVE.MAS berechnet und ergänzt. Nun die fünf notwendigen Parameter:

```
Sektoren pro Spur      :
Datenblocklänge       :
Kapazität der Diskette :
Zahl der Directory Blöcke:
Zahl der Systemspuren  :
```

Beim Übergang auf diese Bildschirmseite werden die aktuellen Inhalte dieser Parameter angezeigt. Programmieren Sie gerade einen neuen Datensatz, so werden zufällige Inhalte angezeigt, die aber keinen Sinn ergeben.

Jetzt können Sie hinter jeden alten Wert einen neuen eingeben. Wollen Sie einen der Parameter nicht ändern, so geben Sie nur <NEW LINE> ein. Wenn Sie einen der Parameter mit 0 programmieren wollen, so müssen Sie -1 eingeben. Null wird als "Keine Änderung" erkannt. Die -1 rechnet das Programm zu 0 um.

Sektoren pro Spur Hier müssen Sie die Zahl der "CP/M Sektoren" eingeben, die auf eine Spur passen. Wenn Sie eine Diskette doppelseitig mit dem Doppelsektorverfahren betreiben wollen, müssen Sie die Sektoren der Vorder- und Rückseite addieren und hier angeben. Diese Zahl ist identisch mit dem Parameter "Sektors/Track" des Programmausdrucks "STAT DSK:".

Datenblocklänge Hier folgt die Anzahl der Bytes, die mindestens durch einen Directoryeintrag belegt werden müssen. Die Bedeutung kommt der der Granule beim Newdos gleich. Eingeben dürfen Sie hier nur 1024, 2048, 4096, 8192 und 16384.

Kapazität der Diskette Hier muß die Anzahl von "CP/M Sektoren" angegeben werden, die auf die Diskette passen. Hat die Diskette Systemspuren, die nicht vom CP/M benutzt werden dürfen, müssen Sie deren Kapazität abziehen. Dieser Wert ist identisch mit dem Wert des Parameters "128 Bytes Record Capacity" des Programmausdrucks "STAT DSK:".

Zahl der Directory Blöcke Hier bestimmen Sie, wieviele Blöcke der unter Datenblocklänge bestimmen Länge das Directory umfassen soll. Sie müssen darauf achten, daß die Zahl der gesamten Directorybytes 8192 (gleich 8K Bytes) nicht überschreitet.

Zahl der Systemspuren Hier geben Sie die Anzahl der geschützten Spuren am Anfang der Diskette an. Beachten Sie, daß Sie beim Doppelspurverfahren mit n Systemspuren in Wirklichkeit nur n/2 Spuren schützen. Dieser Wert ist identisch mit dem Wert des Parameters "Reserved Tracks" des Programmausdrucks "STAT DSK:".

### Dritte Bildschirmseite

Auf der dritten Bildschirmseite können Sie die Sektortabelle programmieren oder editieren. Als erstes werden in sieben Zeilen insgesamt 92 Elemente angezeigt. In der Regel werden nicht alle wirklich benötigt. Die Zahl der relevanten Elemente schwankt bei den verschiedenen Pdrivedatensätzen zwischen 20 und 80.

Jetzt können Sie neue Werte für die einzelnen Elemente eingeben. Dabei korrespondieren die einzelnen Positionen mit den oberen. Probieren Sie das am besten einmal aus. Wollen Sie einen Wert nicht verändern, so drücken Sie einfach <NEW LINE>. Für 0 geben Sie -1 ein, da 0 als keine Änderung erkannt wird. Wollen Sie die Editierung der Sektortabelle abbrechen, so geben Sie -2 ein. Das Programm geht dann zur vierten Bildschirmseite über.

An dieser Stelle taucht vielleicht die Frage auf, wofür man eigentlich eine Sektortabelle benötigt? Mit Hilfe der Sektortabelle wird eine logische Sektornummer in eine physikalische Sektornummer umgesetzt. Dabei korrespondiert dann die logische Nummer 0 mit dem Inhalt der ersten Stelle der Sektortabelle, die logische Nummer 1 mit dem Inhalt der zweiten Stelle der Sektortabelle usw..

Würde keine Umsetzung der logischen in physikalische Sektoren stattfinden, würde das Einlesen einer Spur so ablaufen: Der Controller soll zuerst den Sektor 0 einlesen. Nachdem das geschehen ist, benötigt der Computer immer noch zusätzlich Zeit, um den Inhalt des Sektors an die richtige Adresse zu kopieren, oder sonstwie weiterzuverarbeiten. Als nächstes soll der Sektor 1 eingelesen werden. Da aber der Computer Zeit verbraucht hat, ist der Sektor 1 gerade eben unter dem Tonkopf vorbeigerauscht. Der Controller muß jetzt noch eine ganze Diskettenumdrehung warten, bis der Sektor 1 wieder am Tonkopf vorbei kommt. Nach diesem Verfahren benötigt der Computer mit Controller zum Lesen einer Spur mit 10 Sektoren insgesamt mindestens 11 Umdrehungen der Diskette und damit 2,2 Sekunden.

Setzt man jedoch die logischen Sektoren mit der Sektortabelle (0,4,8,2,6,1,5,9,3,7) in physikalische um, so läuft ein Einladen so ab: Wie oben wird als erstes der Sektor 0 eingeladen und bei der Weiterverarbeitung Zeit verbraucht. Dann soll der Sektor 1 eingeladen werden. Laut Sektortabelle wird dieser aber in die Nummer 4 umgewandelt. Der Controller soll also als nächstes den Sektor 4 einladen. Und siehe da, gerade in diesem Augenblick kommt der Sektor 4 am Tonkopf vorbei und kann sofort eingeladen werden. Dem CP/M wird der Inhalt dieses Sektors als Sektor 1 übergeben, obwohl er auf der Diskette die Nummer 4 trägt. Der Computer hat also in unserem Beispiel zum Verarbeiten eines Sektors soviel Zeit, wie drei Sektoren (1,2,3) benötigen um am Tonkopf vorbei zu kommen. Das sind ca. 60 Millisekunden. Um eine Spur mit 10 Sektoren zu lesen werden dann nur noch 3,8 Umdrehungen der Diskette benötigt, das sind 0,76 Sekunden. Die Zeitersparnis gegenüber dem ersten Beispiel beträgt rund 65 %. Damit ist das Vorhandensein einer Sektortabelle wohl ausreichend erklärt.

Die Sektortabelle des BIOS 2.1 erfüllt insgesamt drei Funktionen. Die erste, das sogenannte Interleaving mit dem entsprechenden Interleavingfaktor habe ich bereits erläutert. Die zweite ist das Umrechnen der ersten Nummer der logischen und der physikalischen Sektoren. Die logischen Sektoren zählen immer von 0 an aufwärts. Die physikalische Nummerierung dagegen kann mit 0 oder mit 1 beginnen. Fängt sie mit 0 an, muß innerhalb der Sektortabelle auch eine 0 stehen. Fängt sie jedoch mit 1 an, so wird es etwas komplizierter. In einem solchen Fall ist die Anzahl der "CP/M Sektoren", die in einen Hardsektor paßt die kleinste Zahl, die innerhalb der Sektortabelle auftauchen darf. Ein Beispiel:

Beim Osborne unter Double Density werden 1K-Bytes lange Sektoren auf die Diskette geschrieben. Die kleinste physikalische Nummer, die auf der Diskette steht ist nicht die 0 sondern die 1. Da aber in 1K-Bytes 8 "CP/M Sektoren" hineinpassen und damit gleichzeitig geschrieben oder gelesen werden, ist die kleinste Zahl, die innerhalb unserer Sektortabelle auftauchen, darf nicht die 0 sondern die 8. Da bei diesem Format insgesamt 40 "CP/M Sektoren" auf eine Spur passen, läuft die Nummerierung in der Sektortabelle von 8 bis 47. Schauen Sie sich das beim Format "OSBORNR DD" einmal an.

Die dritte Funktion, die die Sektortabelle im BIOS 2.1 erfüllen muß, ist die Realisierung des Doppelsektorverfahrens zum Ansteuern der Rückseite von Disketten. Dazu addieren Sie einfach die Zahl 100 zu den Sektoren, die auf der Rückseite liegen sollen. Diese 100 werden vom BIOS 2.1 beim normalen Betrieb wieder abgezogen. Nur werden solche Sektoren eben von der Rückseite der Disketten gelesen, oder auf ihr geschrieben. Dadurch können Sie die doppelte Anzahl von Elementen in die Sektortabelle eingeben. Genau eine Hälfte dieser Elemente muß dabei größer gleich 100 sein und die andere kleiner als 100. Einen Sektor, den Sie, zum Beispiel, mit 137 eingeben, wird auf der Diskette vom BIOS 2.1 unter der Nummer 37 gesucht. Ein Sektor mit der Nummer 37 würde ebenfalls unter der Nummer 37 auf der Diskette gesucht. Nur, der Sektor 37 liegt auf der Vorderseite der Diskette und der Sektor (137-100) auf der Rückseite.

Innerhalb einer Sektortabelle dürfen nie zwei gleiche Elemente stehen, da sonst das Blocking und Deblocking außer Funktion gesetzt wird. Datenverlust wäre die unmittelbare Folge! Dabei gelten die Sektorelemente 137 und 37 nicht als gleich, obwohl sie auf der Diskette die selbe Nummer tragen.

Wenn Sie die Diskette eines fremden Computers anpassen wollen, ist das Erstellen der richtigen Sektortabelle mit Sicherheit das Schwierigste. Dazu müssen Sie im BIOS des entsprechenden Computers, dessen Diskette Sie anpassen wollen, nachschauen. Dabei steht die Sektortabelle bei jedem BIOS an einer anderen Stelle im RAM Speicher. Ein Rezept kann ich Ihnen dafür deshalb leider nicht geben.

#### Vierte Bildschirmseite

Auf der vierten Bildschirmseite können Sie nur den Namen eines Pdrivedatensatzes ändern. Dazu geben Sie einfach einen neuen ein. Er darf nicht länger als maximal 15 Zeichen sein. Wenn Sie den Datensatznamen nicht ändern wollen, geben Sie nur <NEW LINE> ein.

PD \_\_\_\_\_ .COM

Bis jetzt haben wir nur gesehen, was ein Pdrivedatensatz darstellt und wie man ihn programmiert. Hier soll nun gezeigt werden, wie man einen Pdrivedatensatz aus der Datei PDRIVE.SYS entnimmt und ihn einem bestimmten Laufwerk zuordnet. Dazu dient das Programm PD.COM. Es antwortet nach dessen Anstartung mit:

Einstellung der Laufwerkparameter  
Welche Diskette wollen Sie bearbeiten:

Sie können mit dem Programm PD.COM auch die Pdrivedatensätze von Disketten bearbeiten, die in den Laufwerken B bis D eingelegt sind. An dieser Stelle des Programms geben Sie also das Laufwerk an, in dem die zu bearbeitende Diskette liegt. Nachdem Sie A,B,C oder D eingegeben haben, fährt das Programm wie folgt fort:

Die Parameter der Diskette in Laufwerk A.A:S40 DDB:S40 DDC:S40 DDD:S40Eingabe oder NEW LINE :

Wenn Sie die Parameter einer Diskette in Laufwerk B bis D abgefragt haben, so fehlt die Angabe zum Laufwerk A, da dessen Parameter nicht - wie die andern - in der Datei SYS.SYS steht, sondern im BOOTER auf den Systemspuren der Diskette. (Siehe dazu unter BOOTGEN.COM in diesem Handbuch.) Auf keinen Fall aber können Sie die A-Parameter einer Diskette ändern!

Durch Betätigen von <NEW LINE> ohne sonstige Angaben erfolgt ein Warmstart des CP/M Systems. Sie hätten sich dann mit diesem PD.COM Durchlauf nur die aktuellen Pdrivedatensätze listen lassen.

Angezeigt werden hier nur die Namen der aktuellen Datensätze und nicht deren Inhalte. Das ist in der Regel auch nicht nötig, da man weiß, welcher Name zu welchem Datensatz gehört. Man kann sich damit ein Bild machen, welche Diskette man in ein bestimmtes Laufwerk schieben darf und welche nicht.

Nun kommen wir zur Programmierung eines bestimmten Laufwerkes. Dazu geben Sie einfach an Stelle von <NEW LINE> einen Laufwerksnamen, gefolgt von einem Datensatznamen ein. Nun folgen einige Beispiele von gültigen Eingaben:

B:S40

C:OSBORNE

B:GENIE III DS

Die angegebenen Änderungen werden sofort auf der, in der Kopfzeile angegebenen Diskette, in der Datei SYS.SYS verankert.

Der Laufwerksbuchstabe muß immer in Großbuchstaben eingegeben werden. Beim Datensatznamen müssen Sie genau die Zeichen eingeben, wie bei dessen Programmierung mit PDRIVE.MAS. Stehen dabei im Namen Kleinbuchstaben, müssen diese auch hier eingegeben werden.

Nach der Eingabe sucht das Programm PD.COM in der Datei A:PDRIVE.SYS nach einem Pdrivedatensatz der genau den eben eingegebenen Namen trägt. Existiert kein solcher Datensatz, so erfolgt eine Fehlermeldung.

Verlassen können Sie das Programm PD.COM durch einfaches Betätigen der <NEW LINE> Taste. Wollen Sie jedoch, daß die eingegebenen Änderungen nicht nur auf der Diskette verankert werden, sondern daß sie sofort aktiviert werden, müssen Sie das Programm mit "X <NEW LINE>" verlassen.

Durch dieses Verfahren reduzieren sich die Eingaben zur Aktivierung eines bestehenden Pdrivedatensatzes, wie bei folgendem Beispiel, auf ein Minimum:

PD	(Einladen und Anstarten des Programms PD.COM)
A	(Auswählen der Diskette in Laufwerk A)
B:OSBORNE	(Der Datensatz mit dem Namen OSBORNE wird dem Laufwerk B zugeordnet und in der Datei SYS.SYS verankert.)
X <NEW LINE>	(Der neue Datensatz wird in den Arbeitsspeicher eingeladen und aktiviert.)

Nach diesen Eingaben ist Ihr Genie I/II in der Lage, im Laufwerk B Single Density beschriebene Disketten des OSBORNE Microcomputers zu lesen und zu schreiben. Genauso verfahren Sie natürlich bei den anderen Datensätzen. Sie müssen nur darauf achten, daß die Voraussetzungen erfüllt sind, die die einzelnen Datensätze an Ihre Diskettenstation stellen. Oder anders gesagt: Sie können keine FORMULA-Disketten lesen, wenn Sie keine Doppelkopflaufwerke haben, oder keine GENIE III Disketten, wenn Sie ausschließlich 40 Track Laufwerke besitzen.

FORMAT .COM

Mit dem Programm FORMAT.COM können Sie Disketten formatieren. Damit können Sie alle Formate erzeugen, die durch die ersten 24 Datensätze (0 bis 23) der Datei PDRIVE.SYS definiert werden. Es sind insgesamt 16 verschiedene Formatierungsvarianten.

Nach einem Durchlauf des Programms FORMAT.COM ergibt sich folgendes Bild:

```
Formatierungsroutine fuer Genie I/II
40 (0) oder 80 (1) Spuren          ? 0
SS (0) oder DS (1)                 ? 0
SD (0) oder DD (1)                 ? 0
Welche Diskette? Oder <NEW LINE> B
Welche Diskette? Oder <NEW LINE>
```

Nachdem Sie das Programm mit FORMAT <NEW LINE> angestartet haben, fragt es Sie zuerst nach der Anzahl der zu formatierenden Spuren. Geben Sie hier 0 für 40 Spuren und 1 für 80 Spuren ein. Danach wählen Sie aus, ob die Diskette zweiseitig formatiert werden soll. Geben Sie 0 ein für einseitiges Formatieren und 1 für zweiseitiges. Als letztes bestimmen Sie die Schreibdichte der neuen Diskette. Durch Eingabe von 0 wird sie mit Single Density und durch Eingabe von 1 mit Double Density beschrieben. Nun können Sie noch das Laufwerk auswählen, in dem Sie die neue Diskette formatieren lassen wollen. Geben Sie also A, B, C oder D ein. Haben Sie sich an dieser Stelle dazu entschlossen doch keine Formatierung vorzunehmen, betätigen Sie einfach die NEW LINE Taste. Es erfolgt dann ein Warmstart des Systems.

Nach dem Formatieren und Prüfen der neuen Diskette, können Sie mit den gleichen oben ausgewählten Parametern weitere Disketten in beliebigen Laufwerken formatieren. Dadurch sind Sie in der Lage, eine größere Anzahl neuer Disketten ohne große Umstände in einem Durchgang zu formatieren.

Die unterschiedlichen Diskettenformate sind dabei wie folgt organisiert:

Unter Single Density (einfacher Schreibdichte) sind die Hardsektoren des Genie I/II CP/M's 512 Bytes lang. Pro Spur werden dabei fünf Sektoren aufgezeichnet, Die Numerierung läuft dabei von 0 bis 4. Die Sektoren stehen entsprechend ihrer Numerierung hintereinander. Daß heißt, der Sektor 0 ist der erste einer Spur und der Sektor 4 der letzte. Dies ist nicht immer so. Beim GENIE III CP/M, zum Beispiel, stehen die Sektoren in der Reihenfolge der Sektortabelle hintereinander auf den einzelnen Spuren.

Wird beim Genie I/II CP/M eine Diskette einfacher Schreibdichte doppelseitig beschrieben, so läuft die Nummerierung auf der Rückseite der Diskette von 5 bis 9. Das BIOS 2.1 steuert dabei die Rückseite mit dem Doppelsektorverfahren an.

Unter Double Density (doppelter Schreibdichte) sind die Hardsektoren des Genie I/II CP/M's 1024 Bytes lang. Pro Spur werden dabei ebenfalls fünf Sektoren aufgezeichnet. Dadurch ergibt sich unter doppelter Schreibdichte genau eine Verdoppelung des Diskettenspeicherplatzes. Auch die Numerierung läuft von 0 bis 4 und auf der eventuell vorhandenen Rückseite von 5 bis 9.

Die Reihenfolge der Sektoren auf den Spuren entspricht dabei ebenfalls deren Numerierung.

Damit sind die Diskettenformate, die durch FORMAT.COM erzeugt werden können, 100%ig beschrieben.

#### LFORMAT .COM

Das Programm LFORMAT.COM formatiert 8 Zoll Disketten und zwar ausschließlich im IBM 3740 CP/M Standard-Format. Die einzige Möglichkeit den Programmablauf zu beeinflussen besteht darin, daß Sie das Laufwerk angeben müssen, in dem die zu formatierende Diskette eingelegt ist. Beachten Sie dabei bitte, daß 8 Zoll Disketten in der Regel schreibgeschützt sind, wenn die Schreibschutzkerbe nicht zugeklebt ist. Der Versuch eine schreibgeschützte Diskette zu formatieren führt zum Programmabsturz!

BOOTGEN .COM

Das Programm BOOTGEN.COM ist die Entsprechung des SYSGEN-Programms anderer CP/M Microcomputersysteme. Da bei dem Genie I/II CP/M auf den "Systemspuren" kein CP/M System mehr steht, sondern nur noch ein intelligenter BOOTER, heißt das Programm eben BOOTGEN.COM.

Es ermöglicht Ihnen den Inhalt der ersten drei Spuren Ihrer Originaldiskette auf eine beliebige andere Diskette zu kopieren.

Nach dem Anstarten des Programms BOOTGEN.COM meldet es sich mit:

Bootgenprogramm Serien Nr. \_\_\_\_\_.

Eingabe des Pdrivedatensatzes: S40

Wie Sie sehen, fragt Sie das Programm nach einem Pdrive-datensatz. Dieser Datensatz, hier als S40 angenommen, wird im BOOTER verankert. Er muß also zum Format der neuen Diskette passen, die Sie mit einem BOOTER ausstatten wollen.

Als nächstes müssen Sie sich durch die Eingabe eines Codewortes legitimieren:

Legitimation (Codewort) \_\_\_\_\_:

Dieses Codewort bekommen Sie nach Einsendung Ihrer Rücklaufkarte, die der Originaldiskette beim Kauf beiliegt. Ohne dieses Codewort ist das Programm BOOTGEN funktionsunfähig.

Originaldiskette oder <NEW LINE>:

ist der nächste Schritt in der Programmausführung. Hier geben Sie das Laufwerk an, in dem Sie Ihre Originaldiskette eingelegt haben. Nachdem der BOOTER von Ihrer Originaldiskette eingelesen und überprüft wurde, folgt:

Zieldiskette \_\_\_\_\_ oder <NEW LINE>:

An dieser Stelle geben Sie dann das Laufwerk an, in welches Sie die Diskette eingelegt haben, die Sie mit einem BOOTER ausstatten wollen. Beachten Sie, das diese Diskette vorher formatiert werden muß, da das Programm FORMAT.COM immer von Spur 0 an formatiert. Jetzt können Sie so viele Disketten mit einem BOOTER versehen wie Sie wollen, Sie brauchen nur immer wieder ein Laufwerk anzugeben.

Durch Eingabe von <NEW LINE> erfolgt nach einer Zwischenmeldung ein Kaltstart des Systems.

SYSTEM .COM

Mit diesem Programm können Sie insgesamt sieben Systemparameter voreinstellen. Beim Anstarten des Programms werden zuerst die derzeitigen Einstellungen wie folgt aufgelistet.

Einstellung der Systemparameter

Welche Diskette wollen Sie bearbeiten: A  
Ansprechzeit des Autorepeat (in Hex) : 80  
Wiederholzeit des Autorepeat (in Hex) : 18  
Umwandlung von LF zu 00 fuer den Drucker: J  
Tastatur mit Umlauten : N  
Tastatur auf Grossbuchstaben : J  
Y und Z vertauscht : N  
Ausgabe beim Autostart unterdrueckt : J

Wie bei den Programmen PD.COM und BOOTGEN.COM können Sie auch hier eine Diskette auswählen, die Sie bearbeiten wollen. Danach werden deren Systemparameter angezeigt. Und nun die Bedeutungen im einzelnen:

Ansprechzeit des Autorepeat (in Hex) : Hier können Sie die Zeit bestimmen, nach der beim Autorepeat die erste Wiederholung auftreten soll.

Wiederholzeit fuer Autorepeat (in Hex) : Hier bestimmen Sie die Zeit, die zwischen zwei Wiederholungen vergehen soll. Welche Werte Sie bei diesen beiden Parametern eingeben, liegt ganz bei Ihnen. Am besten probieren Sie einfach aus, was Ihren Tippgewohnheiten am nächsten kommt.

Umwandlung von LF zu 00 fuer den Drucker: Mit diesem Parameter entscheiden Sie, ob bei der Druckerausgabe das Steuerzeichen LF (0A hex 10 dez) zu 00 umgewandelt werden soll. CP/M selbst gibt immer die Zeichenfolge CR LF zum Drucker aus, um dessen Druckkopf an den Anfang der nächsten Zeile zu positionieren. Andere Betriebssysteme, wie zum Beispiel NEWDOS 80 V 2.0 geben nur ein CR (0D hex 13 dez) aus, um das Gleiche zu erreichen. Dann muß man jeweils den Drucker anpassen, meistens mit Minidipschaltern. Wechselt man zwischendurch öfters zwei solche Betriebssysteme, so kann das ständige Drucker Auf- und Zuschrauben auf Dauer lästig werden. Dann programmieren Sie diesen Parameter auf J (ja) und alle LF werden zu 00 und damit unwirksam. Bei verschiedenen Programmen kann das jedoch zu Problemen führen, wenn diese zwei- oder mehrmals über ein und

die selbe Zeile drucken. Zum Beispiel verfährt WORDSTAR so, um einen Fettdruck zu realisieren. Dann müssen Sie diesen Parameter auf N (nein) setzen und im Drucker umschalten.

Die nächsten drei Parameter dienen der Voreinstellung der Tastaturbelegung. Egal, wie Sie diese Parameter setzen, Sie können immer während des normalen Betriebs durch ctrl"1", ctrl"2" und ctrl"3" diese Einstellungen umschalten. Durch die Systemparameter geben Sie nur die erste Startbelegung an.

Ausgabe beim Autostart unterdrueckt : Wenn Sie diesen Parameter J (ja) setzen, wird die Ausgabe zum Bildschirm bis zum Ende des Autostarts unterdrückt. Sonst gibt das BIOS von Anfang an alles aus. Die Ausführung des Autostarts kann durch Drücken der <NEW LINE> Taste beim Kaltstart unterdrückt werden.

Wollen Sie die Systemparameter nur listen und nicht verändern, so geben Sie bei Ansprechzeit des Autorepeat (in Hex): einfach "BREAK" ein. Dann erfolgt ein Warmstart des Systems. Wenn Sie jedoch einen der Parameter ändern wollen, so müssen Sie alle sieben Werte neu eingeben, auch die, die Sie nicht ändern wollen. Danach werden die neuen Systemparameter auf der oben ausgewählten Diskette verankert.

### Blocking und Deblocking

Was versteht man darunter? Nun, wir haben schon gesagt, daß die Sektoren für den BDOS-Teil des CP/M's immer eine Länge von 128 Bytes besitzen. Das ist innerhalb des CP/M's einfach so festgelegt. Wenn man jedoch mehrere solcher "CP/M Sektoren" zu einem größeren zusammenfaßt und gemeinsam und gleichzeitig auf die Diskette schreibt, ergeben sich zwei Vorteile.

Einmal spart man an ID Feldern auf der Diskette. Jedem Sektor ist ein ID Feld vorangestellt. In diesem steht die Seite, auf der sich der Sektor befindet (Vorder- oder Rückseite), die Spurnummer, die Sektornummer und die Länge des Sektors. Diese ID Felder werden bei der Formatierung einer Diskette angelegt und bleiben ab diesem Zeitpunkt immer unverändert erhalten. Auch beim Schreiben eines Sektors werden diese ID Felder nicht verändert. Legt man also statt 18 Sektoren zu 128 Bytes, 5 Sektoren zu 512 Bytes an, so braucht man 13 ID Felder weniger und kann diesen Platz mit Sektoren belegen. Damit ergibt sich eine größere Speicherkapazität. Mit 18 kleinen Sektoren ergibt sich eine Speicherkapazität von 2,25 K Bytes pro Spur, mit 5 großen Sektoren dagegen 2,5 K Bytes.

Als zweites können fünf große Sektoren mit 4,2 Umdrehungen der Diskette eingelesen werden, bei 18 kleinen braucht man dazu ca. 6 bis 7 Umdrehungen. Es ergibt sich also zusätzlich ein Zeitvorteil.

Was passiert denn jetzt beim seriellen Lesen einer Spur? Wir gehen dabei davon aus, daß das BIOS 512 Bytes pro Sektor auf die Diskette schreibt. Damit würden genau vier "CP/M-Sektoren" in einen Hardsektor hineinpassen. Das BIOS bekommt dabei nun vom BDOS den Auftrag den "CP/M Sektor" 0 einzulesen. Dazu lädt das BIOS dann den ersten großen Hardsektor ein. Von diesem trennt dann das BIOS die ersten 25% ab und übergibt ihn dem BDOS als "CP/M Sektor" 0. Als nächstes soll dann der "CP/M Sektor" 1 eingelesen werden. Bei einer Hardsektorlänge von 512 Bytes entspricht das genau den zweiten 25% des schon eingelesenen Hardsektors.

Es braucht also an dieser Stelle keine weitere Diskettenleseoperation ausgeführt zu werden. Erst beim "CP/M Sektor" 4 muß ein neuer Hardsektor von der Diskette geholt werden.

Beim Schreiben läuft das ganze etwas komplizierter ab. Wenn das BDOS einen "CP/M-Sektor" auf die Diskette schreiben will, weiß das BIOS noch nicht, ob auch alle vier "CP/M Sektoren" des Hardsektors, in den der "CP/M Sektor" fällt, auf die Diskette geschrieben werden sollen. Dieser Fall tritt immer beim wahlfreien Dateizugriff auf. Wenn jetzt wirklich nur ein "CP/M Sektor" in den Bereich eines Hardsektors geschrieben werden soll, kann das BIOS ihn nicht einfach auf die Diskette schreiben, da damit die drei anderen "CP/M Sektoren" überschrieben würden und verloren wären. Das BIOS muß also vorher den Hardsektor einlesen, in den der zu schreibende "CP/M Sektor" fällt. Damit ist natürlich ein kleiner Zeitverlust verbunden. Nachdem der entsprechende Hardsektor eingelesen wurde, überträgt das BIOS den "CP/M-Sektor" an die richtige Stelle innerhalb des Buffers, in dem der Hardsektor steht. An dieser Stelle wird die eigentliche Schreiboperation auf die Diskette noch nicht ausgeführt, denn es könnte ja sein, daß das BDOS einen weiteren "CP/M-Sektor" überträgt, der ebenfalls innerhalb des gerade eingelesenen Hardsektors liegt. Dieser "CP/M-Sektor" kann dann direkt in den BIOS Sektorbuffer übertragen werden, ohne vorher eine Diskettenleseoperation ausführen zu müssen. Genauso kann es jetzt passieren, daß ein "CP/M Sektor" vom BDOS verlangt wird, der auch schon im BIOS Buffer enthalten ist. Und wieder laufen die Disketten nicht an wodurch schon wieder Zeit eingespart wird.

Es kann also durchaus vorkommen, daß mehrere "Diskettenoperationen" vom CP/M ausgeführt werden, ohne daß die Disketten wirklich einmal anlaufen, besonders unter doppelter Schreibdichte tritt dieser Effekt auf, da dann bei den Genie I/II CP/M Standardformaten 1024 Bytes lange Sektoren auf die Diskette geschrieben werden. Darin gehen immerhin acht "CP/M Sektoren" gleichzeitig.

Jetzt taucht natürlich die Frage auf, wann wird den eigentlich ein Hardsektor auf die Diskette geschrieben? Nun immer dann, wenn ein neuer Hardsektor in den Buffer eingelesen werden muß, wird dieser vorher auf die Diskette entleert. Zusätzlich wird der letzte zu schreibende Hardsektor immer bei einem Warmstart des Systems, beim Umpositionieren des Tonkopfes und beim Selektieren eines anderen Laufwerks durch das BDOS, auf die Diskette geschrieben. Außerdem löst das BIOS Unterprogramm "CONSOLE INPUT", also die Tastaturroutine, ein Schreiben des letzten Sektors aus, wenn dieses nötig ist. Warum das? Wenn dem nicht so wäre, könnte es vorkommen, daß ein Programm die Tastatur abfragt und gleichzeitig der letzte Hardsektor einer Folge von Diskettenschreibleseoperationen noch nicht zur Diskette übertragen wurde. Das liegt am Mechanismus des Blocking. Wenn jetzt an dieser Stelle ein Diskettenwechsel notwendig würde, oder Sie einfach fertig sind und Ihren Computer ausschalten wollen, wären 4 oder 8 "CP/M Sektoren" verloren, weil sie eben noch nicht zur Diskette übertragen wurden. Datenverlust wäre die Folge. Dadurch, daß das Schreiben des letzten Hardsektors durch das Aufrufen der Tastaturroutine ausgelöst wird, ist zu einem solchen Zeitpunkt der Sektorbuffer des BIOS immer leer und Sie können gefahrlos jeden Diskettenwechsel vornehmen.

Das Blocking und Deblocking läuft für den Benutzer völlig automatisch im Hintergrund ab. Das einzige, was Sie davon mitbekommen ist die Tatsache, daß nicht für jede "Diskettenoperation" auch wirklich die Disketten anlaufen müssen. Das sollten Sie sich immer vor Augen halten und nicht glauben, Ihr Programm wäre "abgestürzt" oder gar Ihr Computersystem defekt.

Bei den recht umfangreichen Tests sind durch den Blocking-Deblockingmechanismus noch nie irgendwelche Datenverluste aufgetreten. Ein Warmstart des Systems vor jedem Diskettenwechsel, um Datenverluste zu vermeiden, hat sich als unnötig erwiesen.

Abschließend kann man sagen, daß durch das Blocking und Deblocking, gegenüber System die jeden "CP/M Sektor" auch als Hardsektor auf die Diskette schreiben, erheblich Zeit eingespart wird. Zusätzlich gewinnt man noch an Diskettenspeicherplatz.

### Serielle Schnittstelle

Da es für das Video Genie die serielle Schnittstelle eigentlich nicht gibt, sind die entsprechenden Routinen im BIOS 2.1 nicht enthalten. Allerdings sind drei Vektoren vorbereitet, um in externen Programmen entsprechende serielle Treiber zu installieren. Die Adressen der Vektoren im BIOS 2.1 sind wie folgt:

E400H	Startadresse des BIOS 2.1
E433H	RUART
E435H	WUART
E437H	UARTST

Die drei Vektoren sind jeweils zwei Bytes lang. Die entsprechenden Routinen müssen dabei folgendermaßen aufgebaut sein:

RUART ließt ein Byte von der seriellen Schnittstelle ein und übergibt das Byte im Register A ans BIOS weiter

WUART gibt ein Byte über die serielle Schnittstelle aus. Das BIOS übergibt dabei das auszugebende Byte im Register C an die Treiberroutine.

UARTST überprüft die serielle Schnittstelle, ob ein Byte von der Routine RUART eingelesen werden kann. Im Register A muß dann 0 gesetzt werden, wenn kein Byte abgeholt werden kann und FFH wenn ja.

Das Basic Input/Output System (BIOS)

Hierbei handelt es sich um den hardwareabhängigen Teil des CP/M Betriebssystem. Es nimmt für jedes Computersystem die Anpassung an die spezifische Systemkonfiguration vor. Bedingung an das BIOS ist, daß es mit einer Sprungtabelle beginnt, die 17 Routinen bedient. Ob sich nun wirklich die entsprechenden Routinen dort befinden, ist solange egal, wie weder das BDOS selbst, noch Anwenderprogramme diese Routinen benutzen wollen.

Die 17 Sprungvektoren am Anfang des BIOS

- |                  |  |
|------------------|--|
| 1.COLD BOOT      | Nach dem Laden des Systems sorgt diese Routine für das Initialisieren von Systemparametern.                                |
| 2.WARM BOOT      | Hiermit wird der CCP nach einem <ctrl.C> neu geladen.  |
| 3.CONSOLE STATUS | Fragt die Konsole ab, ob eine Eingabe erfolgt ist.   |
| 4.CONSOLE INPUT  | Eingabe von der Konsole  |
| 5.CONSOLE OUTPUT | Ausgabe eines Bytes zur Bildschirmroutine  |
| 6.LIST OUTPUT    | Ausgabe eines Bytes zur Drucker-schnittstelle  |
| 7.PUNCH          | Serielle Schnittstelle (Ausgabe) nicht implementiert aber vorbereitet  |
| 8.READER         | Serielle Schnittstelle (Eingabe) nicht implementiert aber vorbereitet  |
| 9.HOME DISK      | Dieses Unterprogramm besteht im BIOS 2.1 nur noch aus einem Return-Befehl.   |
| 10.SELECT DISK   | Auswahl eines Laufwerkes   |
| 11.SET TRACK     | Eine bestimmte Spur der Diskette wird angewählt.   |
| 12.SET SECTOR    | Ein bestimmter Sektor wird angewählt.  |
| 13.SET DMA       | Bestimmt den Bufferbereich im USER-Ram, von bzw. in den bei der nächsten Diskettenoperation geschrieben oder gelesen wird. |
| 14.READ          | Einlesen von der Diskette in den DMA-Bereich, 128 Bytes.   |
| 15.WRITE         | Ausgeben auf Diskette vom DMA-Bereich, 128 Bytes.  |
| 16.LIST STATUS   | Abfrage ob der Drucker bereit ist  |
| 17.SECTOR TRANS  | Übersetzung der logischen in die physikalische Position eines Sektors auf der Diskette.                                    |

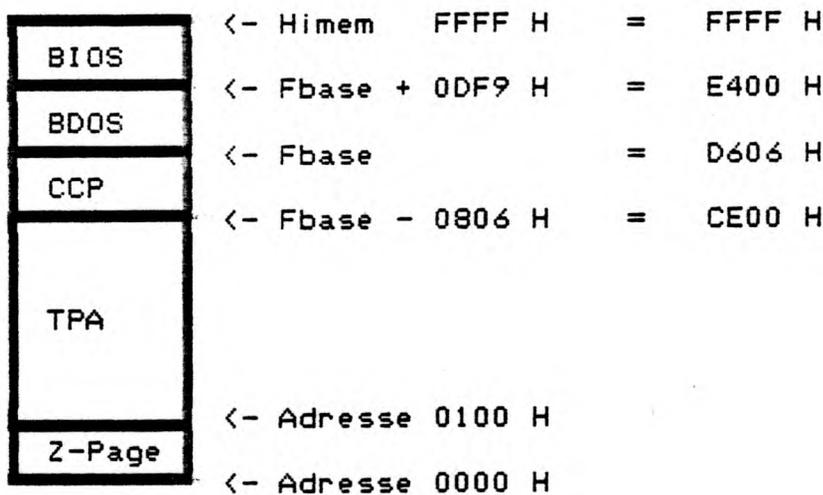
Die Einsprung- und Rücksprungkonditionen  
der 17 BIOS Funktionen

Routine	Einsprung- Konditionen	Rücksprung- Konditionen
COLD BOOT		
+ WARM BOOT		
CONSOLE STATUS		A=Status *
+ CONSOLE INPUT		A=Charakter
CONSOLE OUTPUT	C=Charakter	
LIST OUTPUT	C=Charakter	
PUNCH OUTPUT	C=Charakter	
+ READER INPUT		A=Charakter
- HOME DISK		
+ SELECT DISK	C=Drive	HL=DHA
+ SET TRACK	C=Track	
SET SECTOR	C=Sektor	
SET DMA	BC= DMA Adresse	
% READ DISK		A=Status *
% WRITE DISK		
LIST STATUS		A=Status *

\* Status:           00 = fertig (bereit)  
                  01 = Fehler  
                  FF = beschäftigt

**DHA: Disk Parameter Header Address:** Hier steht eine Tabelle von Adressen, die angeben, wo das BDOS verschiedene Werte und Variablen zwischenspeichern darf. Die DHA selber umfaßt dabei 16 Bytes. Jedes Laufwerk besitzt eine eigene DHA. Hier steht unter anderem auch, wo die eigentlichen Pdriveparameter abgespeichert sind. Das sind die 15 Bytes, die die logische Organisation der Diskette beschreiben und die Sie mit dem Programm PDRIVE.MAS Befehl "E" zweite Bildschirmseite programmieren können. In der DHA steht auch die Anfangsadresse der Sektortabelle.

- + Alle so gekennzeichneten Unterprogramme schreiben den letzten Hardsektor einer Diskettenschreiboperation auf die Diskette.
- % Alle so gekennzeichneten Unterprogramme schreiben ihn nur dann auf die Diskette, wenn es unbedingt sein muß, um weiterarbeiten zu können.
- HOME DISK löst keine Schreiboperation aus. Es besteht im BIOS 2.1 lediglich aus einem RET-Befehl.

Die Speicherorganisation

## Die Speicherbelegung im einzelnen:

- 0-2 Jump WARMBOOT
- 3 Das IOBYTE (beim Kaltstart mit 80H initialisiert)
- 4 Das momentan angesprochene Laufwerk (ab Null zählend)
- 5-7 Jump BDOS (dies ist der Einsprungpunkt für alle BDOS-Operationen, Byte 6 und 7 geben die FBASE wieder, FBASE-1 gibt die größte für Anwenderprogramme nutzbare RAM-Adresse an)
- 8 RST 1
- 10 RST 2
- 18 RST 3
- 20 RST 4
- 28 RST 5
- 30 RST 6
- 38 RST 7 wird vom DDT-Programm benutzt. Alle anderen Restartadressen sind momentan nicht benutzt.
- 5C-7C Vom CCP zur Speicherung des FCB (File Control Block) benutzt.
- 80-FF Vom CCP als File In-Outputbuffer benutzt (DMA-Buffer)
- 0100H Ab dieser Adresse beginnt die TPA (Transient Program Area). Hier, und nur hier, können Anwenderprogramme laufen. Dieser Speicherbereich kann maximal ausgedehnt werden bis zum Beginn des BDOS, also bis FBASE-1.
- FBASE-0806H Hier liegt der CCP (Console Command Prozessor) im Speicher. Er bildet die Grundkommandoebene. Sie wird automatisch nach jedem Warm- oder Kaltstart des Systems angesprungen. Der CCP versteht insgesamt sechs verschiedene Befehle.
- FBASE Anfang des BDOS (Basic Disk Operating System). Das BDOS organisiert den File I/O-Fluß zur und von der Diskette. Er verwaltet auch das Directory der Diskette.
- FBASE+0DF9H Hier beginnt das BIOS (Basic Input/Output System) und zwar mit den 17 Sprungvektoren zu den einzelnen Unterprogrammen. Das BIOS ist der einzige Programmteil, der zur Anpassung des CP/M's an jeden unterschiedlichen Computertyp neu geschrieben werden muß.
- FFFFH Ende des BIOS und Ende des Speicherplatzes.

CP/M und Video Genie

Das BDOS und der CCP sind auf allen CP/M Computern bis aufs letzte Bit gleich, eben Standard. Das BIOS enthält die Unterprogramme zur Steuerung der Tastatur, des Videointerfaces und die Routinen, um Sektoren auf die Diskette zu schreiben oder von ihr zu lesen.

Die Länge des BIOS selbst ist nicht vorgeschrieben. Kleine BIOS haben eine Länge von ein bis zwei K-Byte. Es gibt allerdings auch BIOS die zehn K-Byte und länger sind. Das hat zur Folge, daß jeder Computer dem CP/M eine andere Anzahl von Bytes als Arbeitsspeicher zur Verfügung stellt. Alle CP/M's sind jedoch Standard-CP/M's, solange die untere Grenze des Speicherplatzes die Adresse 0000H ist. Deshalb benötigt man entweder die Option EG-64 oder den LSS 1, da diese in der Lage sind, die Adressen 0000H bis 3FFFH auszublenden und an dieser Stelle 16 K-Byte zusätzliche Ram Speicherstellen einzublenden.

Hier liegt auch die einzige Schwierigkeit, da damit der Disk-Controller ( Adressen 37E0H bis 37FFH ), die Tastatur ( Adressen 3800H bis 3BFFH ) und das Video-Ram ( Adressen 3C00H bis 3FFFH ) ausgeblendet werden. Das BIOS muß also immer, bevor es auf eine dieser drei Einheiten zugreift, die Erweiterungsplatine von Mode 111 auf Mode 000 umschalten, da sonst auf die entsprechenden RAM Speicherstellen zugegriffen wird und nicht auf die Ein- oder Ausgabeeinheiten. Nach dem Ausführen von, zum Beispiel einer Tastaturabfrage, muß das BIOS natürlich wieder auf Mode 111 zurückschalten, bevor es per RET Befehl zum aufrufenden Programm zurückkehrt, damit wieder die vollen 64 K-Byte Ram zur Verfügung stehen.

Eine zusätzliche Schwierigkeit tritt durch die Möglichkeit des CP/M's auf, die DMA-Adresse durch die BIOS-Funktion 13 (SET DMA) einzustellen. Stellen wir uns einmal vor, wir legen diese Adresse innerhalb der ersten 16-K-Byte Ram (Adressen 0000H-3FFFH) des Arbeitsspeichers, die ja von der Erweiterungsplatine gesteuert werden. Nun wollen wir einen Diskettensektor in diesen Bereich laden und rufen dazu die BIOS-Funktion 14 (READ) auf. Das BIOS schaltet dann zuerst die Erweiterungsplatine von Mode 111 auf Mode 000 zurück, um die Floppystation ansteuern zu können. Wie oben schon erwähnt wird dadurch automatisch auch das Ram der Adressen 0000H-3FFFH ausgeblendet. Die Funktion READ würde zwar vom Bios richtig ausgeführt, aber die Daten würden nicht im Ram landen (wie auch), sondern zum Beispiel im Rom oder in der Tastatur, je nach dem, welche DMA-Adresse wir gewählt haben. So geht es also nicht! Das Problem zu lösen, ist ganz einfach. Wir definieren im BIOS, das ja ganz oben im Speicher liegt und durch die Umschaltungen der verschiedenen Modes der Erweiterungsplatine nicht beeinflußt wird, einen Zwischenspeicher, der genau einen Diskettensektor aufnehmen kann.

Die Ausführung der BIOS-Funktion 14 (READ) und entsprechend der BIOS-Funktion 15 (WRITE) läuft dann wie folgt ab:

1. Die BIOS Funktion 14 (15) wird aufgerufen.
2. Das BIOS schaltet die Erweiterungsplatine von Mode 111 auf Mode 000 um.
3. Der entsprechende Diskettensektor wird von der Diskette gelesen und in dem BIOS-Zwischenspeicher abgelegt. Dies ist deshalb immer problemlos möglich, da die oberen 48-K-Ram des Arbeitsspeicher nicht von der Erweiterungsplatine beeinflusst werden. Und das Bios liegt ja ganz oben in diesem Bereich.
4. Nun muß das BIOS die Erweiterung wieder auf Mode 111 zurückschalten. Der zu lesende (schreibende) Diskettensektor steht zu diesem Zeitpunkt schon im Ram des Computers, nämlich im Zwischenspeicher des BIOS. Alle Sektoren, die von der Diskette gelesen oder auf ihr geschrieben werden, müssen diesen Zwischenspeicher durchwandern, damit ein fehlerloser Betrieb des CP/M's unter allen Umständen gewährleistet ist.
5. Nun sind also wieder die vollen 64-K-Byte Ram ansprechbar, und der zwischengespeicherte Diskettensektor kann vom BIOS entsprechend der eingestellten DMA-Adresse umgeladen werden.
6. Nach der ausgeführten BIOS-Funktion wird durch einen RET Befehl wieder zum aufrufenden Programm zurückgesprungen.

Das hört sich alles sehr kompliziert an, ist es aber eigentlich nicht. Bei laufendem CP/M macht sich das immerwährende Umschalten gar nicht bemerkbar. Dafür läuft das alles viel zu schnell ab. Selbst das Umladen der Diskettensektoren im Ramspeicher ist gegenüber der eigentlichen Diskettenoperation kein Zeitfaktor, der ins Gewicht fällt.

Nun glauben Sie vielleicht, das BIOS fürs Genie I/II ist das komplizierteste, was es gibt. Weit gefehlt! Kaum ein CP/M-Computer hat alle Ein- und Ausgabegeräte auf dem Input-Output-bus des 8080, 280 Computers liegen. In einem solchen Fall bräuchte man während des Betriebs des CP/M nicht mehr laufend Speicherplatz auszublenden, um Zugang zu den Ein- und Ausgabegeräten zu haben, wie das beim Video Genie notwendig ist. Aber selbst diese Computersysteme kommen um ein Banking nicht herum, denn beim Einschalten des Computers fängt der 8080, 280 Prozessor (Kern eines jeden CP/M Computers) ab Adresse 0000H an, ein Programm auszuführen. Dort muß dann auf jeden Fall ein Rom liegen, durch dessen Programm das CP/M von der Diskette in den Arbeitsspeicher geladen wird. Danach wird auch bei diesen Systemen das Rom ausgeblendet, um volle 64-K-Ram zur Verfügung zu haben.

Sie sehen, kein CP/M Computer kommt um eine Bankinglogik herum. Die Steuerung dieser Logik, die beim Video Genie durch die Erweiterungsplatine repräsentiert wird, übernimmt bei allen CP/M Computern das BIOS.