

meter-Block auswerten (siehe Tabelle). Diese Methode hat diverse Vorzüge, aber leider auch Grenzen, und – hält man es für möglich? – es gibt hübsche Unterschiede zwischen PCDOS 3.2, MSDOS 3.2 und PCDOS 3.3.

Da das Anlegen eines Boot-Sektors mit vollständigem Disk-Parameter-Block bei älteren DOS-Versionen nicht vorgeschrieben war, sind alle drei der neueren DOS-Versionen aus Kompatibilitätsgründen in der Lage, Disketten ohne Boot-Sektor anhand ihres Media-Bytes zu erkennen. Allerdings können dabei immer nur die Formate erkannt werden, die vom Hersteller für den jeweiligen Rechner und seine Laufwerke vorgesehen sind. Im AT wird also stets bei F9h das 1,2-MB-Format, auf Rechnern der PS/2-Familie das 720-KB-Format gewählt.

Die neuen DOS-Versionen verweigern aber jegliche Mitarbeit, wenn weder Boot-Sektor noch Media-Byte zu finden sind (etwa alles 00). Dann setzt es eine Fehlermeldung, die nicht mal mehr den sektorweisen Zugriff mittels DEBUG zuläßt, obwohl dieser zumindest auf die ersten 8 Sektoren der Spur 0 immer möglich wäre, da das Format schließlich physikalisch und logisch lesbar bleibt.

Chaos per ST...

Hier kommt nun der Atari ST ins Spiel. An sich verfügt er über einen Boot-Sektor mit einer (weitgehend) korrekten und vollständigen Formatbeschreibung, jedoch fängt sein Boot-Sektor nicht mit dem Sprung (einer xx86-CPU von Intel) in den Bootstraploader an. Deshalb wird sein Boot-Sektor als nicht DOS-konform ignoriert und der Disk-Parameter-Block einfach nicht ausgewertet. Leider setzt der Formatierer des ST ein Media-Byte F7h, das es gemäß IBM-Manual gar nicht geben sollte, an den Anfang der FATs. Schlimmer noch, der ST schreibt dieses Byte sowohl beim ein- als auch beim zweiseitigen Format vor die FAT.

Selbst wenn man aber auf einer Atari-Disk herumpatchte, indem man F9h für '80 Spuren doppelseitig' vor die FAT schriebe und damit ein gültiges IBM-Format kennzeichnete – es hilft nichts. DOS würde bei

diesem Media-Byte nämlich eine andere FAT-Größe verwenden als der ST und folglich auch das Directory, das immer direkt im Anschluß an den letzten FAT-Sektor angelegt wird, an anderer Stelle vermuten – mit verheerenden Resultaten, versteht sich.

Eine brauchbare Abhilfe besteht darin, die ersten drei Bytes im Boot-Sektor der Atari-Disketten umzupatchen: ein JMP (EBh), ein beliebiges Byte (Sprungweite) und ein NOP (90h) machen die DOS-Versionen bereits glücklich – solange es nur um das doppelseitige Format des ST geht.

... und per DOS

PCDOS 3.2 und 3.3 halten nämlich noch eine besondere Schikane bereit: Das am weitesten verbreitete Atari-Format, nämlich das einseitige, behandeln diese DOS-Versionen total falsch. Sie werten auch nach einem Disk-Patch den Boot-Sektor einfach nicht korrekt aus.

Mit dem Programm Memmap zum Beispiel (in c't 9/87, 'Betriebssystem-Forscher') bringt man in Erfahrung, daß das einseitige ST-Format im DOS-internen Parameterblock penetrant mit einem statt mit zwei Sektoren pro Cluster geführt wird, außerdem wird das Media-Byte F8h (im DPB des ST-Boot-Sektors) gemäß IBM als Kennung für eine Festplatte interpretiert. Aber auch wenn man hier sonstwas einträgt, PCDOS bleibt stur und macht Murks. MSDOS 3.2 hingegen arbeitet korrekt, wie es die spärliche Dokumentation verspricht.

PCDOS 3.2 hat übrigens noch ein weiteres Problem offenbart, das aber möglicherweise nur auf ATs wegen der dort üblichen Auswertung des Disk-Change-Signals zum Tragen kommt. Bei Nicht-MF-Drives darf ja kein Disk-Change-Signal vorhanden sein (siehe Kasten 'Floppy-Anschluß an AT'), und somit kann der AT anhand dieses Signals auch nicht bemerken, daß eine neue Disk eingelegt wurde, deren Format möglicherweise anders ist. Und immer dann, wenn PCDOS 3.2 'auf die alte Tour', also nur anhand des Media-Bytes, das Format ermittelt, kann es von einem einmal ein-

geloggt Format nicht zu einem anderen wechseln, auch dann nicht, wenn bei der neuen Diskette wieder ein auswertbarer Boot-Sektor vorliegt. Der AT kommt erst nach einem Warmstart wieder zu sich.

Man kann sich also auch bei den neuen überarbeiteten DOS-Versionen nur darauf verlassen, daß die Formate korrekt unterstützt werden, die IBM verwendet. Und wenn Sie vielleicht nach einer Methode suchen, wie man einen PC ab DOS 3.2 ohne Fehlermeldung ins Nirwana schicken kann, so ändern Sie doch mal die Cluster-Größe im Boot-Sektor zum Beispiel auf 00. Ein Disk-Zugriff genügt, denn Plausibilitätsprüfungen waren wohl zur Zeit der DOS-Entwicklung noch nicht erfunden. . .

Nicht nur Atari

Da die Herum-Patcherei auf Disketten nicht jedermanns Glück ist, haben wir den Treiber so ausgelegt, daß er die Original-Atari-Disketten sowohl mit den alten (unter 3.2) als auch mit allen drei neuen DOS-Versionen lesen und beschreiben kann. Da eine Boot-Sektor-Auswertung ohne eben diese Disk-Patches nicht funktioniert, bleibt nur die Auswertung des Media-Bytes.

Und schon folgt das nächste Problem: da für beide ST-Formate dasselbe Byte F7h verwendet wird, ist keine selbsttätige Unterscheidung möglich. Um dennoch beide ST-Formate bearbeiten zu können, muß man den Treiber zweimal ins System einbinden (zweimal in CONFIG.SYS eintragen), jeweils mit anderen Parametern in der Aufrufzeile. Dann kann man also etwa unter Laufwerk D: das einseitige Format, unter E: das zweiseitige bearbeiten. Die Parameter sind in einem Kasten erläutert.

Unser Treiber ist aber noch aus anderen Gründen generell darauf eingerichtet, daß keine Boot-Sektor-Informationen benutzt werden; denn es gibt auch noch andere Fälle, in denen die neuen DOSse nicht weiterkommen: so enthält der Disk-Parameter-Block zum Beispiel keine Information über einen physikalischen Sektorversatz auf der Scheibe, und der DEC-Rainbow benutzt einen solchen.

Zwar sind die weiteren 80-Spur-Formate, die wir im Treiber explizit einstellen, jetzt möglicherweise für die neuen DOS-Versionen direkt lesbar (wir haben es nicht ausprobiert), aber Benutzer von DOS 3.1 und älteren Versionen können ohnehin nur über solche Treiber abweichende Formate bearbeiten. Deshalb haben wir diese Formate, die bereits im Treiber aus c't 2/87 unterstützt wurden, noch dringelassen.

Wie erwähnt gibt es aber diverse 80-Spur-Formate mit dem Media-Byte F9h, so daß dem Treiber explizit (über die Aufrufzeile) mitgeteilt werden muß, welches Format er bei diesem Media-Byte zugrunde legen soll. Wird kein Parameter angegeben, so stellt er sich auf das neue Format der IBM-Rechner ein. Dieses 720-KB-Format wird exakt gleich bei allen 80-Spur-Laufwerken mit normaler Schreibdicke benutzt, also sowohl auf 5,25- als auch 3,5-Zoll-Drives. Will man verschiedene F9h-Formate gleichzeitig unterstützen, so muß der Treiber wieder mehrfach eingebunden werden und diese Formate kann man nur unter verschiedenen Laufwerksbuchstaben ansprechen.

Um Mißverständnissen vorzubeugen: Es muß nicht für jedes im Treiber aufgeführte Format ein eigener Treiberaufruf erfolgen, sondern nur dann, wenn doppeldeutige Media-Bytes auftauchen, also bei F7h und F9h, und mehrere Formate mit diesen Media-Bytes unterschieden werden müssen. Im Normalfall können also mit einem Treiber sowohl ein Atari-, ein F9h-Format und alle 80-Spur-Formate mit unterschiedlichen Media-Bytes automatisch bedient werden.

Für 'Treiber-Neulinge' noch der folgende Hinweis: Ohne diverse Disk-Treiber im System korrespondieren die physikalischen Disk-Laufwerke 1 und 2 mit den logischen Laufwerken A: und B:, die Harddisk als drittes Drive wird unter C:, eine RAM-Disk dann als viertes Laufwerk unter D: angesprochen. Das sind aber nicht die einzigen Möglichkeiten, denn man kann auch ein und dasselbe physikalische Laufwerk über mehrere Treiber unter weiteren Laufwerksbuchstaben verwalten (DOS nimmt immer den nächsten freien Buchstaben).