

Und genau das muß man tun, wenn man die beiden Atari-Formate mit unserem Treiber unterstützen will. Aber auch, wenn man das 720-KB-Format im AT auf dem ersten physikalischen Drive, das sonst unter A: angesprochen wird, bearbeiten will, muß man das über den Laufwerksbuchstaben tun, den der Treiber bei seinem Aufruf (beim Booten) anzeigt. Kurzum: Man sollte sich genau über den Unterschied von physikalischem Laufwerk, also der puren Hardware, die vom Floppy-Controller über das Select-Signal angesprochen wird, und den logischen Laufwerken mit seinen Buchstabenkennungen, die das DOS zur Verwaltung benutzt, im klaren sein.

Sonderfälle

Der Treiber läßt sich auch mit der Option '40' (für 40 Spuren)

betreiben. Dieser Fall ist eher historisch bedingt und soll es Anwendern einer auf 80-Spur-Betrieb gepatchten DOS-Version unterhalb 3.2 ermöglichen, weiterhin 40-Spur-Disketten lesen zu können. Für den in der letzten c't getesteten Toshiba T1000 mit seinem 80-Spur-DOS der Version 2.11 könnte hier Bedarf bestehen. Auch kann es hilfreich sein, einen solchen Treiber einzubinden, wenn ein Boot-Sektor mit fehlerhaften Parametern den direkten Zugriff über DOS ab 3.2 verhindert.

Eine weitere Option (A) ist ausschließlich Besitzern eines c't86 mit IFC-Karte vorbehalten. Nur diese können damit den Treiber benutzen, um das AT-Format zu bearbeiten. Bisher wurde dieser Betriebsfall beim c't86 mit einer gesonderten Treiberversion (ATDRV auf den Monitor-Disketten) gehand-

habt, die damit nun überflüssig ist.

Humor mit Format

So wunderbar viel unser Treiber auch kann – und auch besser kann als DOS –, er kann nicht formatieren. Hier kommt mal nicht der hinlänglich bekannte englische, sondern echter amerikanischer Humor zum Ausdruck. Der DOS-Dokumentation von Microsoft darf man entnehmen: Ab DOS 3.2 können Disk-Treiber selbständig formatieren. Was daran komisch ist?

Wenn man die DOS-Dokumentation nun geflissentlich liest, stellt man fest, daß es eine vielversprechende neue DOS-Funktion 44h (IOCTL) gibt, die Treiber beispielsweise auch zum Formatieren befähigen soll. Der Haken dabei ist allerdings, daß der Treiber die über Funktion

44h gebotenen Möglichkeiten alle (Schreiben, Lesen und so weiter) beherrschen muß, da man im Attribut-Wort nur festlegen kann, ob er's ganz kann oder gar nicht. Insgesamt ist der Aufwand nicht unbeträchtlich, den man treiben darf.

Aber jetzt zum eigentlichen Scherz. In den Unterlagen heißt es, im Listing des RAM-Disk-Treibers finde man ein schönes Beispiel, wie Disk-Treiber zu schreiben seien. Das ist richtig, nur wird dort nirgends DOS-Funktion 44h unterstützt, der Treiber ist auch als nicht IOCTL-fähig gekennzeichnet. Na ja, und das Formatieren kann man an RAM-Disks eh nicht nachvollziehen, da dies dort nicht stattfindet.

Also schaut man mit dem Debugger nach, wie es denn wohl DRIVER.SYS macht – denn der ist ja offiziell als Formatier-

Treiberaufrufe im Detail

Der neue Treiber ist zwar noch vielseitiger als sein Vorgänger, seine vielfältigen Möglichkeiten werden allerdings auch über zum Teil recht aufwendige Parameter-Angaben beim Aufruf erkaufte. In der allgemeinen Form erfolgt die Einbindung durch eine Zeile in CONFIG.SYS wie

```
device=laufwerk\pfad\name.sys (parameter)
```

'laufwerk:', 'pfad' und 'name' geben dabei an, wo und unter welchem Namen die Treiberdatei beim Booten gesucht werden muß. Etwas komplizierter ist die Angabe von (parameter), die in allgemeiner Form folgendermaßen aufgeschlüsselt ist:

```
tracks phys.drive [opt1] [opt2]
```

Wird für 'tracks' keine der beiden möglichen Spurzahlen '40' oder '80' angegeben, so stellt der Treiber default-mäßig '40' ein. Diese Option ist 'historisch' bedingt und nur sinnvoll für Anwender von auf reinen 80-Spur-Betrieb gepatchten DOS-Versionen unterhalb 3.2. Es sei auch explizit darauf hingewiesen, daß die Angabe '40' kein Double-Stepping auf 80-Spur-Drives erzeugt. Bei der Angabe '80' bedient der Treiber grundsätzlich nur 80-Spur-Formate, bei '40' grundsätzlich nur 40-Spur-Formate.

Bei Aufruf mit '80' für ein MF-Laufwerk in einem AT schaltet der Treiber das vom ROM-BIOS verursachte Double-Stepping ab. Damit sind alle 80-Spur-Formate schreib-/lesbar, die vom Treiber unterstützt werden (siehe auch Listing), genau wie beim Betrieb mit 'normalen' 80-Spur-Laufwerken über diesen Treiber.

'phys.drive' teilt dem Treiber mit, für welches physikalische Laufwerk er verantwortlich ist (gezählt wird ab 0 fürs erste Laufwerk). Diesen Parameter und die Spurzahl sollte man immer angeben, denn wenn die beiden fehlen (aber auch wenn die Parameter insgesamt falsch sind), wählt der Treiber stets die Einstellung '40 Spuren, Laufwerk 1'.

Da mehrere verschiedene Formate mit dem Media-Byte F9h gekennzeichnet werden, muß man mit N (Nixdorf), O (Olivetti) und S (Siemens) als Option (opt1) das jeweils zu unterstützende Format bei diesem Media-Byte explizit vorgeben. Keine Option oder '1' heißt, daß der Default-Wert für das IBM 720-KB-Format von IBM (siehe auch Tabelle) genommen wird.

Das gleiche Problem, nämlich die fehlende Unterscheidungsmöglichkeit durch dasselbe Media-Byte F7h zu Beginn der FAT bei ein- und doppelseitigen Disketten, tritt beim Atari ST auf. Um hier eine eindeutige Wahl treffen zu können, kann als letzte Angabe (opt2)

in der Zeile '1' oder '2' stehen: für einseitig oder zweiseitig. Ohne Parameter gilt die Einstellung 'zweiseitig'.

Die Option 'A' ist nur für den c't86 mit der IFC-Karte gedacht und ermöglicht den Betrieb mit MF-Laufwerken im High-Density-Modus (1.2 MByte). c't86-User brauchen also keine separate Version ATDRV.SYS mehr. Der Aufruf lautet dann

```
device=drv.sys 80 phys.drive A
```

Das folgende Beispiel zeigt eine mögliche CONFIG.SYS-Datei für einen AT, der als erstes physikalisches Laufwerk ein MF-Drive eingebaut hat, als zweites ein 3,5-Zoll-Laufwerk mit 80 Spuren (normale Schreibdichte). Angestrebt wird, auf dem MF-Laufwerk 5,25-Zoll-Disketten im Nixdorf-Format schreiben und lesen sowie mit 720 KByte (IBM-Format) schreiben, lesen und formatieren zu können. Auf dem 3,5-Zoll-Laufwerk sollen PS/2-kompatible Scheiben beschrieben, gelesen und formatiert werden können. Außerdem soll es noch möglich sein, sowohl einseitige als auch doppelseitige Atari-ST-Disketten lesen und schreiben zu können. Folgendes müßte dann in der Datei CONFIG.SYS stehen:

```
device=driver.sys /D:0/F:2
device=driver.sys /D:1/F:2
device=drv.sys 80 0
device=drv.sys 80 0 N
device=drv.sys 80 1 1
device=drv.sys 80 1 2
```

Die ersten beiden Treiber (DRIVER.SYS wird ab PC-/MSDOS 3.2 mitgeliefert) werden nur zum Formatieren benötigt, der erste für das MF-Laufwerk, der zweite für den 3,5-Zöller. (Beim Formatieren auf dem MF-Laufwerk muß vorher stets noch die Status-Zelle umgesetzt werden, siehe Haupttext.) Da in ATs normalerweise eine Harddisk den Laufwerksbuchstaben C: belegt, bekommen die beiden Treiber die Buchstaben D: und E: zugewiesen.

Unser eigentlicher Schreib-/Lesetreiber heißt im Beispiel DRV.SYS und wird einmal fürs MF-Laufwerk (Buchstabe F:) ohne weitere Optionen aufgerufen und unterstützt daher beim Media-Byte F9h das 720-KB-Format von IBM. Beim zweiten Aufruf (Buchstabe G:) bietet er statt dessen die Möglichkeit, das Nixdorf-Format zu verarbeiten.

Die Treiber für das 3,5-Zoll-Drive stellen sich bei F9h auf das IBM-Format (720 KB, PS/2), beim Media-Byte F7h jedoch einmal auf Ataris einseitiges (Buchstabe H:), zum andern auf Ataris zweiseitiges (Buchstabe I:) Format ein. So flexibel DOS mit seinem Treiberkonzept auch ist, um speziellen Anforderungen der Praxis gerecht zu werden, muß man einigen Aufwand und auch Umstand treiben.