

80-Spur-Drives kein Double-Stepping unterdrückt werden. Ebenso werden entsprechende Programmteile deaktiviert, wenn als Parameter '40' erkannt wurde.

Ob das Programm auf einem IBM AT gestartet wurde, erkennt es daran, daß der Wert an der vorletzten Speicherstelle im ROM (F000:FFFEh) FCh beträgt. Wird der Treiber im 80-Spur-Modus für ein MF-Laufwerk aufgerufen, ist die Double-Stepping-Unterdrückung aktiv. Die Umschaltung selbst wird direkt vor dem Einsprung in die eigentlichen Lese- und Schreibroutinen vorgenommen, die den INT 13h aufrufen. Nach den Lese- oder Schreiboperationen wird der

Wert in der Speicherstelle 0040:0090h wieder berichtigt.

Da wir diesmal auch etwas mehr Zeit zum Kommentieren hatten, lassen sich die Implementationsdetails der ausführlich beschriebenen Neuerungen recht gut aus dem Listing ansehen.

Trotz aller Überprüfungen hatte sich in die abgedruckte Urversion ein Fehler eingeschlichen. Bei der Fehlerbehandlung wurde nicht die korrekte DOS-Fehlernummer an das aufrufende Programm zurückgegeben, sondern immer der gleiche Fehlercode. Dies lag an mehreren JNZ anstatt JZ.

Die Praxis mit dem Ursprungstreiber zeigte noch ein anderes Problem auf, für das der Treiber

aber keine Schuld hat. Wie schon in der Kartei für den INT 13h (ROM-BIOS) beschrieben, kann es zu Übertragungsfehlern kommen, wenn die DMA-Page-Adresse bei Überschreitung der Seitengrenze nicht richtig aufdatiert wird.

Erst DOS 3.0 behandelt diesen Fehler überhaupt und versucht ihn zu umgehen, aber leider ungenügend. Bei uns trafen vor allem Nachfragen von Olivetti- und Commodore-PC-Benutzern diesbezüglich ein, bei denen anscheinend das sehr speziell angepaßte IO.SYS (Disk-BIOS) älterer MSDOS-Versionen Probleme machte. Doch dadurch, daß DOS 3.x wesentlich größer als seine Vorgänger ist, wird die

Anfangsadresse des freien Speichers so weit nach oben verschoben, daß die Wahrscheinlichkeit, auf einen Seitenüberlauf zu treffen, wesentlich verringert wird. So kann es aber immer noch vorkommen, daß bei bestimmten Konfigurationen die Ladeadresse für die Daten über den Treiber immer noch eine Adresse berührt.

Hier kann man nur über einen kleinen Trick Abhilfe schaffen: Man muß die Anfangsadresse des freien Speichers künstlich so legen, daß kein Seitenfehler erfolgt. Am einfachsten gelingt dies, indem man ein Dummy-Programm als Treiber einfügt und somit die Startadresse des freien Speichers nach oben verschiebt. (gr)

```

1: ;*****
2: ;*
3: ;*      Universeller PC-/MSDOS-Diskettentreiber
4: ;*
5: ;* Programmname      : unidrv.asm
6: ;* Ersteller         : Martin Ernst
7: ;* Erweitert         : Peter Köhlmann
8: ;* Erstellungsdatum  : 11.03.86
9: ;* letzte Änderung  : 16.08.87
10: ;* letzter Test am   : 02.09.87
11: ;* Versionsnummer   : 3.0
12: ;* Revisionsnummer  : 6.0
13: ;*
14: ;*****
15:
16:      page      72,132
17:      title     PC-/MSDOS-Universal-Treiber
18:
19:
20: false      equ      0
21: true       equ      not false
22: v20        equ      true      ; true für Rechner mit V20/30,
23: v20        equ      false     ; 80188/86 oder 80286
24:
25: length     equ      13h
26:
27: .186
28: ausgabe macro message
29:      mov      dx,offset message
30:      mov      ah,9
31:      int      21h
32:      endm
33:
34:
35: ; diese Definition ist der Einsprung in die Monitorfunktionen des c't 86
36:
37: monitorem segment at 0f000h
38:      org      0e000h
39: monitor   label far
40: monitorem ends
41:
42:
43: code      segment para public
44: assume cs:code,ds:code,es:code,ss:code
45:
46: ;*****
47:
48: diskdev: dw      -1,-1
49:          dw      0000100000000000b ;Bit 11 von Attribut-Word gesetzt
50:          dw      strategy
51:          dw      diskint
52:          dw      1
53:
54: ptrsav   dd      0 ;hier wird der Zeiger auf die Anforderungs-
55:                  ;kopfeile gesichert
56:
57:          db      3 dup(?)
58:
59:
60: disktbl: dw      diskinit ; "Init Device-Treiber"
61:          dw      mediachk ; "Check Media Type"
62:          dw      getbpb    ; "Get BIOS Parameter Block " des
63:                  ; selektierten "Media Type"
64:          dw      cmderr    ; "I/O-Control Input" (nicht implementiert)
65:          dw      diskred   ; "Read Data"
66:          dw      busexit   ; "Non Destructive Read Data" (n.i.)
67:          dw      exit      ; "Input Status"
68:          dw      exit      ; "Input Flush Buffer"
69:          dw      diskwr    ; "Write Data"
70:          dw      diskwrv   ; "Write Data" mit Verify-Funktion
71:          dw      exit      ; "Output Status"
72:          dw      exit      ; "Output Flush Buffer"
73:          dw      exit      ; "I/O-Control Output"
74:          dw      exit      ; "Open"
75:          dw      exit      ; "Close"
76:          dw      exit      ; "Removable Media" --> gleich nach "exit",

```

```

77:
78: ; kein "busy", da diskette
79: ;*****
80:
81: iodat     struc
82:      cmdlen db      ? ; Länge der Tabelle
83:      unit   db      ? ; Einheiten-Code
84:      cmd     db      ? ; Befehls-Code
85:      status  dw      ? ; Status der Operation
86:      db      8 dup (?)
87:      media  db      ? ; Media Descriptor Byte
88:      trans  dd      ? ; Transfer Adresse
89:      count  dw      ? ; Anzahl der Blöcke oder Character,
90:                  ; die übertragen werden sollen
91:      start  dw      ? ; erster Block, der transferiert werden soll
92:      drive  db      ? ; zugeordnete Drive-Bezeichnung
93:      iodat   ends
94:
95: ;*****
96:
97: bpbs      struc
98:      db      13 dup (?)
99:      bpb1   db      ? ; Struktur für die Rücklieferung von Werten
100:      bpb2   dw      ? ; über die Anforderungskopfeile
101:      bpb3   dw      ?
102:      bpb4   dw      ?
103:      bpb5   dw      ?
104:      bpbs   ends
105:
106: ;*****
107:
108: dbp       struc
109:      secsize dw      ? ; Sektorgröße (512 bytes)
110:      alloc   db      ? ; Anzahl Sektoren pro Cluster
111:      ressec  dw      ? ; Anzahl reservierte Sektoren
112:      fats    db      ? ; Anzahl der FATS
113:      maxdir  dw      ? ; Anzahl der Directory-Einträge
114:      sectors dw      ? ; Gesamtzahl der Sektoren
115:      mediaid db      ? ; Media Descriptor Byte
116:      fatsec  dw      ? ; Anzahl der Sektoren pro FAT
117:      sectrk  dw      ? ; Anzahl der Sektoren pro Track
118:                  ; (nicht Zylinder!)
119:      koepfe  dw      ? ; Anzahl der Köpfe (bzw. Seiten bei Floppy)
120:      hidden  dw      0 ; versteckte (hidden) Sektoren
121:      dbp     ends
122:
123:
124: ; IBM PS/2, 720 KByte, 80 Spuren
125: ibmps     dbp      <512,2,1,2,112,1440,0f9h,3,9,2>
126:
127: ; Philips YES
128: philips   dbp      <512,2,1,2,176,1440,0feh,3,9,2>
129:
130: ; DEC Rainbow (MS-PCDOS), 80 Spuren
131: decrain   dbp      <512,1,20,2,96,800,0fah,3,10,1>
132:
133: ; ct86, 80 Spuren, einseitig
134: ct86      dbp      <512,1,1,2,112,720,0fch,4,9,1>
135:
136: ; ct86 80 Spuren, doppelseitig
137: ct86      dbp      <512,2,1,2,224,1440,0fch,4,9,2>
138:
139: ; Atari ST (default: doppelseitig, 80 Spuren)
140: atari     dbp      <512,2,1,2,112,1440,0f7h,5,9,2>
141:
142: inittabl: dw      ibmps.secsize
143:
144: ;*****
145:
146: stratp proc far
147: ; Registerbelegung beim Aufruf:
148: ; bx = Offset der Anforderungszeile
149: ; es = Segment der Anforderungszeile
150:
151: strategy:
152:      mov      word ptr cs:[ptrsav],bx ; Adresse der Zeile wird

```