

In die Geheimnisse der Floppy eingetaucht

Formatieren einer Diskette ist für jeden Floppy-Besitzer das erste, was er mit ihr macht. Was beim Formatieren passiert und weshalb die Floppy so nervig rattert, warum es so lange dauert und wie es schneller geht, erfahren Sie in folgendem Artikel.

Teil 6

Wie jedem Floppy-Besitzer bekannt ist, muß eine Diskette vor dem ersten Speichern von Daten formatiert werden. Wie eine Diskette nach einem solchen Formatiervorgang aussieht, wurde schon besprochen.

Uns soll nun interessieren, was während des Formatierens so alles in der Floppy passiert und warum die 1541 so lange für einen eigentlich sehr einfachen Vorgang benötigt.

Zur Wiederholung: Beim Formatieren werden vom DOS alle wichtigen Markierungen auf die Diskette gebracht und außerdem sämtliche Sektoren in ihrer späteren Form angelegt.

Der Vorgang des Formatierens verwendet zu seiner Ausführung einen uns schon bekannten Jobcode, nämlich \$E0.

Bevor das DOS den eigentlichen Formatiervorgang startet, wird ab \$0600 (also im Puffer 3) ein Sprungbefehl abgelegt: JMP \$FAC7.

Dieser Sprungbefehl ist eine Art Vektor, der im RAM liegt und somit verändert werden kann. Er bietet dem Benutzer die Möglichkeit, eine eigene Routine einzubauen, die dann bei jedem Trackwechsel angesprochen wird, um so einige wirksame Manipulationen an der Formatierung vorzunehmen, indem zum Beispiel Werte in der Zero-page verändert werden, doch

dazu später. Üblicherweise zeigt dieser Vektor direkt auf eine Jobroutine, die für das Formatieren zuständig ist. Diese Routine wird nun vom Hauptprogramm mit dem Jobcode \$E0, der in Speicherstelle \$03 geschrieben wird, aufgerufen.

Formatieren in der Jobschleife

Am Anfang der Jobroutine steht nun die Abfrage, ob schon mindestens ein Track formatiert wurde oder ob dieser Einsprung der allererste ist. Ist dieser Einsprung der erste, so werden alle Parameter für den Stepmotor gesetzt; danach erfolgt ein Rücksprung in die Jobschleife. Hier wird der Tonkopf nun 45 (!) Tracks zurückgefahren, was sich in jenem charakteristischen Rattern der Floppystation äußert.

Nun, können Sie sagen, es würde auch reichen, wenn der Kopf nur 35 oder 40 Spuren zurücktransportiert würde. In der Tat ist der Wert 45 sehr hoch. Man muß aber bedenken, daß es passieren kann, daß der Schreib-/Lesekopf der Floppy durch eine defekte Diskette oder einen Programmierfehler zu weit nach innen gefahren und beispielsweise auf Track 42 am Anschlag gelandet ist, daß ein Zurückfahren um 40 Tracks einfach nicht ausreicht, um den Tonkopf richtig zu positionieren.

Der Wert von 45 Tracks enthält also eine Sicherheitsreserve, die ein Positionieren auf Spur 1 mit Sicherheit ermöglicht.

Wurde der Kopf also auf Track 1 positioniert, so erfolgt erneut ein Einsprung in die Formatieroutine; eine Speicherstelle zeigt jetzt an, daß der Kopf auf Track 1 positioniert wurde und das Formatieren starten kann.

Jetzt wird noch geprüft, ob auf die nächste Spur umgeschaltet werden soll, da die aktuelle bereits formatiert wurde (wenn ja, erfolgt wieder ein Einsprung in die Jobschleife, um das Nötige zu tun).

Diese Abfragen am Anfang der Formatieroutine scheinen umständlich und überflüssig zu sein; das Gefühl täuscht jedoch. Wir dürfen ja nicht vergessen, daß die Routine immer nur jeweils einen Track formatiert und danach zur Jobschleife zurückkehrt, damit der Tonkopf weitergeführt werden kann. Wir haben also gewissermaßen eine Endlosschleife, die nur durch die Feststellung, daß Spur 35 fertig formatiert wurde, beendet wird.

Ausmessen einer Spur

Jetzt haben wir aber endlich alle Voraussetzungen zum Formatieren eines Tracks erfüllt und wollen an die Arbeit gehen. Der Abschnitt, der jetzt besprochen wird, ist übrigens für die

langwierige Formatierung verantwortlich und sorgt für die ausgedehnten Wartezeiten.

Bevor die SYNC-Markierungen und Sektoren auf eine Spur geschrieben werden, wird diese Spur vom DOS »ausgemessen«.

Das Betriebssystem der 1541 »weiß« im Normalfall genau, wieviele Bytes für die SYNC-Markierungen und Sektoren einer Spur benötigt, beziehungsweise verbraucht werden.

Jetzt ist es aber so, daß die Sektoren nicht genau auf jede Spur abgemessen sind; vielmehr hat die Diskette pro Spur eine etwas höhere Kapazität, als eigentlich benötigt wird. Aus dieser Tatsache folgt natürlich, daß zwischen den einzelnen Sektoren »Leerstellen« entstehen, die keine Daten enthalten.

Da jetzt aber die Länge der Tracks von außen (Track 1) nach innen (Track 35) kontinuierlich abnimmt, werden diese Leerstellen immer kleiner; wir haben also unterschiedliche Anzahlen von »Leerbytes« zwischen den Sektoren.

Das DOS ist nun bestrebt, die Sektoren jeder Spur möglichst symmetrisch anzuordnen, also immer den gleichen Abstand zwischen zwei Sektoren eines Tracks zu haben. Bild 1 zeigt, was passiert, wenn keine vorherige Ausmessung stattfindet.

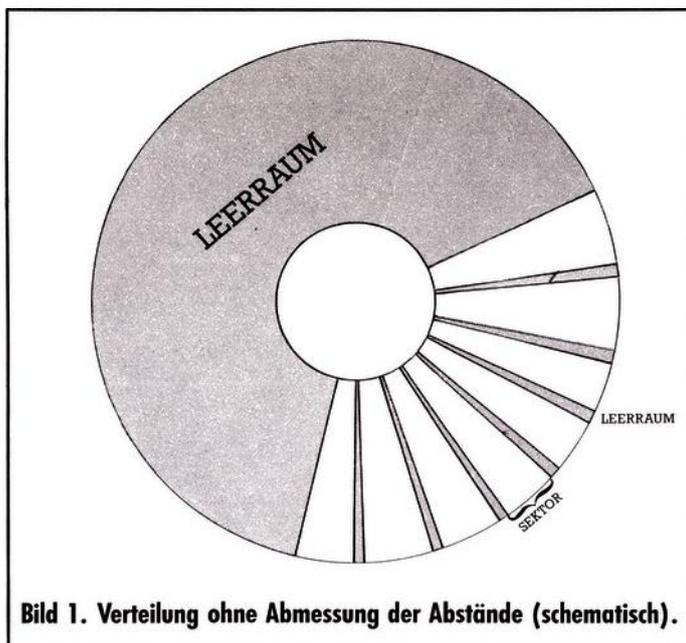


Bild 1. Verteilung ohne Abmessung der Abstände (schematisch).

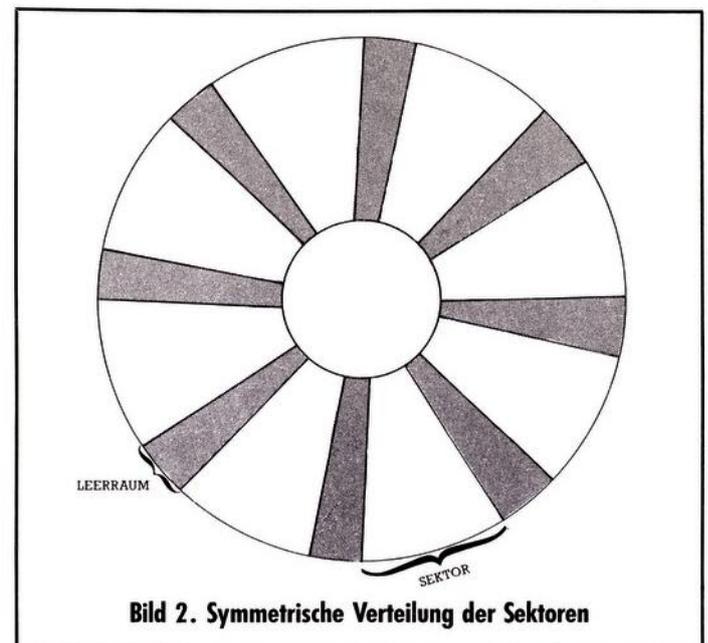


Bild 2. Symmetrische Verteilung der Sektoren

Um das Ziel einer »symmetrisch« formatierten Diskette zu erreichen, stellt das DOS durch einige komplizierte Schreib- und Lesevorgänge das Verhältnis zwischen benötigtem und vorhandenem Platz einer Spur fest. Aus diesem Verhältnis kann nun anhand einer einfachen Rechnung festgestellt werden, wieviel Platz zwischen den einzelnen Sektoren freigelassen werden muß.

Nachdem diese komplizierte Vermessung stattgefunden hat, die mehrere Diskettenumdrehungen und damit Zeit erfordert, beginnt nun das eigentliche Formatieren der Diskette, das mit allem Drum und Dran normalerweise nicht mehr als eine 1/3 Sekunde für eine Track benötigt.

Das Anlegen der Sektoren im Puffer

Bevor geschrieben werden kann, müssen die Sektoren erst einmal im Pufferspeicher der 1541 hergestellt werden. Da sich die einzelnen Sektoren nur durch deren Header unterscheiden, reicht das Anlegen der Blockheader; die Inhalte der Datenblöcke sind bei jedem Sektor gleich und bestehen aus dem schon bekannten Muster \$4B gefolgt von 255 \$01-Bytes.

Die Blockheader werden alle in einem Pufferspeicher (\$0300-\$03FF) abgelegt; der Inhalt der Datenblöcke steht ab \$0500 bis \$05FF.

Schreiben eines Tracks auf Diskette

So, alle Vorarbeiten wären jetzt abgeschlossen. Wir können mit dem Schreiben auf Diskette beginnen. Zuerst wird der Disk-Controller auf Schreibmodus gestellt und die Spur der Diskette gelöscht.

Der gesamte Spurinhalte wird nun während einer einzigen Diskettenumdrehung (1/3 Sekunde) auf die Diskette gebracht, wobei zuerst die SYNC-Markierung für den Blockheader, danach der Blockheader selbst geschrieben werden. Nach einer Lücke von 9 Byte folgt die SYNC-Markierung des Datenblocks mit den zugehörigen Datenbyte. Den Abschluß eines Sektors bildet der schon erwähnte »Leerraum«, der aus der vorher errechneten Anzahl von Byte besteht.

Zur Sicherheit erfolgt nach dem Schreiben eine Verify-Routine, die auf eventuelle Disketten- oder Schreibfehler kontrolliert und bei deren Auftreten einen »24, READ ERROR« ausgibt.

Mit dieser letzten Maßnahme ist eine Spur einer Diskette fertig formatiert worden, und es wird auf Erreichen der Spur 35 abgefragt. Wurde Spur 35 formatiert, so werden alle Flags für das Formatieren zurückgesetzt, die Jobschleife verlassen und ins Hauptprogramm zurückgekehrt.

Formatieren ohne ID

Im Hauptprogramm wird nun auf Track 18 positioniert. Die BAM der Diskette wird hergestellt und in Block 18,0 abgelegt. Anschließend wird noch der erste Directory-Block (18,1) mit Nullen vollgefüllt und ebenfalls abgespeichert, womit das Formatieren abgeschlossen wäre.

Formatiert man eine Diskette nur kurz, das heißt ohne Angabe einer ID beim N-Befehl, so werden alle anfänglichen Schritte weggelassen. Es wird in diesem Fall nur auf das richtige Formatkennzeichen in der BAM (\$41/65/A) kontrolliert und danach der eben beschriebene Vorgang auf Track 18 durchgeführt.

Formatieren mit »Variationen«

Nun wäre unser Floppykurs natürlich kein Floppykurs, wenn wir unsere neu gewonnenen theoretischen Kenntnisse nicht sofort in die Praxis umsetzen wollten.

In der Tat kann man mit Hilfe der Formatieroutine im DOS einige nette »Scherze« auf eine Diskette bringen, die entweder dem Spieltrieb oder dem Softwarewortschutz dienen können.

Ich habe vorhin schon erwähnt, daß die Formatieroutine jeweils über einen Sprungbefehl bei \$0600 im RAM der Floppy aufgerufen wird.

Diese Adresse wird bei jedem neuen Track angesprungen und bietet so die Möglichkeit, Tracks zu erzeugen, die in ihrem Aufbau voneinander abweichen, wenn entsprechende Eingriffe vorgenommen werden.

Diese Möglichkeit eines Eingriffes wollen wir an dieser Stelle aber gar nicht erst weiter diskutieren, da es ziemlich aussichtslos ist, hier ohne dokumentiertes DOS-Listing an die Arbeit zu gehen.

Daß wir kein DOS Listing besitzen, soll aber noch lange nicht heißen, daß wir nicht in der Lage sind, auf anderem Weg, Eingriffe in die Formatierung vorzunehmen. Wenn wir nicht effektiv mit der fest eingebauten Routine zusammenarbeiten können, dann schreiben wir uns eben ein vollständig eigenes Programm, das im RAM der Floppy abgelegt wird und uns für Abänderungen unendlich viele Möglichkeiten bietet.

Formatierung »selbst gebaut«

Sehen Sie sich einmal Listing 1 an. Ich habe hier ein Formatiersystem entwickelt, das einfacher und schneller arbeitet als die DOS-Routine und trotzdem ein paar zusätzliche Möglichkeiten bietet.

Da das Gesamtprinzip aber fast 100prozentig mit der im DOS eingebauten Routine übereinstimmt, können Sie sich anhand des Source-Code-Listings einmal die »praktische Ausführung« einer Formatieroutine ansehen.

Um Ihnen die Eingabe des Programms zu erleichtern, habe ich einen DATA-Lader als Listing 2 beigefügt, wobei ich Ihnen empfehlen möchte, diesen gleich einmal einzutippen.

Das Programm wird nur aktiviert, wenn alle DATAs richtig eingetippt wurden. Haben Sie alles richtig gemacht, so steht nach der Ausführung des Laders ein Maschinenprogramm am Basic-Anfang, dem eine Basic-Zeile beigefügt ist. Das Programm sollten Sie sich jetzt mit SAVE auf eine Diskette speichern und danach mit RUN starten.

Nach einer winzigen Verzögerung erscheint die READY-Meldung und der Cursor wieder. Das Formatierungsprogramm wurde jetzt in den Bereich ab \$C000 (49152) geschoben und der SAVE-Vektor abgeändert.

Tippen Sie jetzt einfach den Befehl SAVE — ohne Anführungszeichen und Filenamem — ein und drücken Sie RETURN. Es erscheint nun die Startmeldung des Formatprogrammes. Sie können jetzt einen Namen für eine Diskette eingeben (maximal 16 Zeichen werden angenommen). Danach erwartet der Computer eine zweistellige ID. Schließlich, und das ist das besondere an diesem Programm, können Sie noch den ersten und letzten zu formatierenden Track eingeben. Diese Eingabe muß hexadezimal erfolgen und erlaubt einen Bereich von \$01 bis \$FF.

Achtung! Wird eine Zahl größer als \$29 (41) eingegeben, wird es in der Regel kritisch. Der Kopfist dann nämlich am oberen Anschlagpunkt angelangt.

Etwas ist noch zu beachten: Ein Nachformatieren einer Spur auf einer gefüllten Diskette ist mit dem Programm ohne Änderung nicht möglich, da das Directory auf jeden Fall neu geschrieben wird. Wird die Diskette nicht vollständig formatiert, so ist darauf zu achten, daß die gleiche ID eingegeben wird, wie sie schon für die übrige Diskette Gültigkeit hat, da es sonst einen »29, DISK ID MISMATCH ERROR« gibt.

Wollen Sie dennoch einen Einzeltrack neu formatieren, ohne das Directory zu zerstören, so können Sie das durch eine einfache Änderung im Floppy-Programm erreichen. Sie gehen in Listing 1a an die Adresse \$06B5. Den Befehl JSR \$EE40 und das nachfolgende RTS ersetzen Sie durch lauter NOPs.

Eine Änderung des Directory-Track unterbleibt jetzt, sofern Sie die Tracknummern zur Formatierung entsprechend wählen, da dieser Befehl die Routine zum kurzen Formatieren im DOS aufgerufen hätte.

In jedem Fall gilt aber: Bei Formatieren von Einzeltracks müssen diese die gleiche ID wie die übrige Diskette erhalten.

Eine weitere Möglichkeit dient der Schonung des Laufwerks. Wenn Sie sich das Floppy-Programm noch einmal betrachten, dann finden Sie bei Adresse \$0696 den Befehl an den Disk-Controller, einen BUMP auszuführen. Wenn Sie hier das \$C0 durch ein \$00 ersetzen, dann unterbleibt dieses Anschlagen des Tonkopfes am Anfang des Formatierens. Diese Maßnahme ist immer dann nützlich, wenn mehrere Disketten hintereinander formatiert werden sollen.

Zur Zeitdauer ist noch zu sagen, daß das Programm für eine Diskette zirka 30 Sekunden benötigt und damit um einiges schneller ist als das Programm im DOS der 1541. Warum das so ist, wollen wir gleich erfahren.

Geschwindigkeit; aber wie?

In meinem Formatierprogramm wurde die Berechnung der Lücke zwischen zwei Sektoren weggelassen. Wir können nämlich davon ausgehen, daß diese Lücken auf jeder Diskette in etwa gleich sind. Aus diesem Grund verwende ich einfach einen Erfahrungswert für die Länge der Lücke, der zusätzlich noch einen Sicherheitsbereich enthält. Diesen Wert sehen Sie in Listing 1a an der Adresse \$05DF.

Wenn Sie mit dem Programm Disketten formatieren, werden Sie feststellen, daß die Datensicherheit auch weiterhin voll gewährleistet ist.

Im Gegensatz zu anderen schnellen Formatierprogrammen habe ich aber nicht auf ein Verify verzichtet, da das Formatieren die einzige Möglichkeit bietet, defekte Disketten rechtzeitig zu erkennen, ohne daß dabei wichtige Daten verlorengehen. Einmal ganz davon abgesehen, macht das Verifizieren außerdem nur einen sehr kleinen Teil am Geschwindigkeitsverlust aus, so daß die Sicherheit vor einigen Sekunden Zeitgewinn Vorrang haben sollte.

Wollen Sie die Zeit dennoch einmal ohne Verify messen, so »klemmen« sie den Rest der Formatierungsroutine ab \$05FD ganz einfach ab, indem Sie an dieser Stelle nach JMP \$FE00 ein JMP \$FD9E einfügen. Eine weitere Verbesserung gegenüber dem DOS 2.6 der 1541 hat eigentlich mehr kosmetischen Charakter.

Es geht hier um den Leerinhalt von Datenblöcken, nachdem eine Diskette neu formatiert wurde. Den Inhalt werden Sie höchstwahrscheinlich schon kennen: Es steht am Anfang des Datenblocks ein \$4B gefolgt von 255 \$01-Bytes. ▶

Floppyprogramm zum Disk-Format-System
1985 by KOSS

```

0500 ea nop
0501 a5 0a lda #0a Tracknummer aus Jobspeicher
0503 c9 24 cmp #24 größer als 35?
0505 90 07 bcc #050a nein
0507 a9 12 lda #12 ja: 18 als Anzahl der Sektoren
0509 85 43 sta #43 festlegen
050b 4c 13 05 jmp #0513
050e 20 4b f2 jsr #f24b Anzahl der Sektoren holen
0511 85 43 sta #43 und merken
0513 a9 00 lda #00
0515 85 1b sta #1b Sektorzähler setzen
0517 a0 00 ldy #00
0519 a2 20 ldx #20
051b a5 39 lda #39
051d 99 00 03 sta #0300,y Kennzeichen #08 für Blockheader
0520 c8 iny
0521 c8 iny
0522 a5 1b lda #1b Sektornummer
0524 99 00 03 sta #0300,y
0527 c8 iny
0528 a5 0a lda #0a Tracknummer
052a 99 00 03 sta #0300,y
052d c8 iny
052e a5 13 lda #13 ID 2
0530 99 00 03 sta #0300,y
0533 c8 iny
0534 a5 12 lda #12 ID 1
0536 99 00 03 sta #0300,y
0539 c8 iny
053a a9 0f lda #0f
053c 99 00 03 sta #0300,y Lücke lassen
053f c8 iny
0540 99 00 03 sta #0300,y
0543 c8 iny
0544 a9 00 lda #00
0546 59 fa 02 eor #02fa,y Prüfsumme bilden
0549 59 fb 02 eor #02fb,y
054c 59 fc 02 eor #02fc,y
054f 59 fd 02 eor #02fd,y
0552 99 f9 02 sta #02f9,y und abspeichern
0555 e6 1b inc #1b
0557 a5 1b lda #1b Sektorzähler erhöhen
0559 c5 43 cmp #43 schon Maximalzahl?
055b 90 be bcc #051b nein, weitermachen
055d a9 03 lda #03
055f 85 31 sta #31
0561 98 tya
0562 48 pha
0563 8a txa
0564 9d 00 07 sta #0700,x Datenblock mit #00 füllen
0567 e8 inx
0568 d0 fa bne #0564
056a 20 30 fe jsr #fe30
056d 68 pla
056e a3 tay
056f 88 dey
0570 20 e5 fd jsr #fde5
0573 20 f5 fd jsr #fdf5
0576 a9 07 lda #07
0578 85 31 sta #31
057a 20 e9 f5 jsr #f5e9 Prüfsumme für Datenblock
057d 85 3a sta #3a abspeichern
057f 20 bf f7 jsr #f78f
0582 a9 00 lda #00 Sektorzähler setzen
0584 85 32 sta #32
0586 20 0e fa jsr #fe0e Track löschen
0589 a9 ff lda #ff
058b 8d 01 1c sta #1c01 SYNC schreiben
058e a2 05 ldx #05
0590 50 fe bvc #0590
0592 b8 clv
0593 ca dex
0594 d0 fa bne #0590
0596 a2 0a ldx #0a
0598 a4 32 ldy #32
059a 50 fe bvc #059a Blockheader schreiben
059c b8 clv
059d b9 00 03 lda #0300,y
05a0 8d 01 1c sta #1c01
05a3 c8 iny
05a4 ca dex
05a5 d0 f3 bne #059a
05a7 a2 09 ldx #09
05a9 50 fe bvc #05a9 Lücke von 9 Bytes lassen
05ab b8 clv
05ac a9 55 lda #55
05ae 8d 01 1c sta #1c01
05b1 ca dex
05b2 d0 f5 bne #05a9
05b4 a9 ff lda #ff
05b6 a2 05 ldx #05
05b8 50 fe bvc #05b8 SYNC-Markierung für Datenblock
05ba b8 clv
05bb 8d 01 1c sta #1c01
05be ca dex
05bf d0 f7 bne #05b8
05c1 a2 0b ldx #0b
05c3 50 fe bvc #05c3
05c5 b8 clv
05c6 bd 03 01 lda #0100,x
05c9 8d 01 1c sta #1c01
05cc e8 inx
05cd d0 f4 bne #05c3
05cf a2 00 ldy #00
05d1 50 fe bvc #05d1 Datenblock schreiben
05d3 b8 clv
05d4 b1 30 lda (#30),y

```

```

05d6 8d 01 1c sta #1c01
05d9 c8 iny
05da d0 f5 bne #05d1
05dc a9 55 lda #55
05de a2 08 ldx #08
05e0 50 fe bvc #05e0 Lücke nach Sektor mit fester
05e2 b8 clv Länge von 8 Bytes schreiben
05e3 8d 01 1c sta #1c01
05e6 ca dex
05e7 d0 f7 bne #05e0
05e9 a5 32 lda #32
05eb 18 clc
05ec 69 0a adc #0a
05ee 85 32 sta #32
05f0 c6 1b dec #1b
05f2 d0 95 bne #0589 schon alle Sektoren?
05f4 50 fe bvc #05f4 nein, weitermachen
05f6 b8 clv
05f7 50 fe bvc #05f7
05f9 b8 clv
05fa 20 02 fe jsr #fe00 auf Lesen umschalten
05fd a5 c8 lda #c8 200 Leseversuche
05ff 85 1f sta #1f
0601 a9 00 lda #00
0603 85 30 sta #30
0605 a9 03 lda #03
0607 85 31 sta #31
0609 a5 43 lda #43
060b 85 1b sta #1b Sektorzähler
060d 20 56 f5 jsr #f556 auf SYNC-Signal warten
0610 a2 0a ldx #0a
0612 a0 00 ldy #00
0614 50 fe bvc #0614
0616 b8 clv
0617 ad 01 1c lda #1c01
061a d1 30 cmp (#30),y
061c d0 0e bne #062c
061e c8 iny
061f ca dex
0620 d0 f2 bne #0614
0622 18 clc
0623 a5 30 lda #30
0625 69 0a adc #0a
0627 85 30 sta #30
0629 4c 35 06 jmp #0635
062c c6 1f dec #1f
062e d0 d1 bne #0601
0630 a9 06 lda #06
0632 4c d3 fd jmp #fdd3
0635 20 56 f5 jsr #f556
0638 a0 bb ldy #bb
063a 50 fe bvc #063a
063c b8 clv
063d ad 01 1c lda #1c01
0640 d9 00 01 cmp #0100,y
0643 d0 e7 bne #062c
0645 c8 iny
0646 d0 f2 bne #063a
0648 a2 fc ldx #fc
064a 50 fe bvc #064a
064c b8 clv
064d ad 01 1c lda #1c01
0650 d9 00 07 cmp #0700,y
0653 d0 d7 bne #062c
0655 c8 iny
0656 ca dex
0657 d0 f1 bne #064a
0659 c6 1b dec #1b
065b d0 b0 bne #060d
065d 4c 9e fd jmp #fd9e
0660 a0 00 ldy #00
0662 b9 e0 06 lda #06e0,y
0665 99 00 02 sta #0200,y
0668 c8 iny
0669 cc df 06 cpy #06df
066c 90 f4 bcc #0662
066e ad df 06 lda #06df
0671 8d 74 02 sta #0274
0674 ad de 06 lda #06de
0677 8d 7b 02 sta #027b
067a a9 00 lda #00
067c 85 7f sta #7f
067e 20 00 c1 jsr #c100
0681 ac 7b 02 ldy #027b
0684 b9 00 02 lda #0200,y
0687 85 12 sta #12
0689 b9 01 02 lda #0201,y
068c 85 13 sta #13
068e 20 07 d3 jsr #d307
0691 a9 1a lda #1a
0693 8d 05 1c sta #1c05
0696 a9 c0 lda #c0
0698 85 00 sta #00
069a a5 00 lda #00
069c 30 fc bmi #069a
069e ae dc 06 ldx #06dc
06a1 86 0a stx #0a
06a3 a9 e0 lda #e0
06a5 85 02 sta #02
06a7 a5 02 lda #02
06a9 30 fc bmi #06a7
06ab c9 02 cmp #02
06ad b0 0c bcs #06bb
06af e8 inx
06b0 ec dd 06 cpx #06dd
06b3 90 ec bcc #06a1
06b5 20 40 ee jsr #ee40
06b8 60 rts

```

Lücke nach Sektor mit fester Länge von 8 Bytes schreiben



schon alle Sektoren? nein, weitermachen

auf Lesen umschalten 200 Leseversuche

Sektorzähler auf SYNC-Signal warten

Daten vergleichen

Zähler vermindern

24, READ ERROR SYNC-Signal abwarten

Listing 1a.
Das Floppy-Programm zum
Format-System

Datenblock testen

Ende; zur Jobschleife

Start des Floppyprogramms

Disknamen übernehmen

Länge der Zeile setzen

Kommaposition setzen

Drive 0 setzen LED am Laufwerk an

ID 1 holen

id 2 holen

alle Kanäle schließen

Timer setzen BUMP anfordern

auf Ausführung warten erste Tracknummer

Track formatieren

auf Ende warten Fehler aufgetreten? verzweige, wenn ja

schon Zieltrack formatiert? weiter, wenn nein Directory herstellen Ende

```

06b9 ea nop
06ba ea nop
06bb a2 02 ldx #02
06bd 4c 0a e6 jmp #e60a Diskstatus ausgeben; Ende

Computerprogramm zum Disk-Format-System
(c) 1985 by KOSS

.. c200 a2 00 ldx #00
.. c202 20 87 c2 jsr #c287 Titel und erste Frage ausgeben
.. c205 a0 00 ldy #00
.. c207 20 cf ff jsr #ffcfcf Eingabe holen
.. c20a c9 0d cmp #0d
.. c20c f0 08 beq #c216
.. c20e 99 e0 c1 sta #c1e0,y Namen abspeichern
.. c211 c8 iny
.. c212 c0 10 cpy #10 schon 16 Zeichen ?
.. c214 90 f1 bcc #c207 weiter, wenn nein
.. c216 a9 2c lda #2c Komma hinter den Namen setzen
.. c218 99 e0 c1 sta #c1e0,y
.. c21b c8 iny
.. c21c 8c de c1 sty #c1de
.. c21f a2 47 ldx #47
.. c221 20 87 c2 jsr #c287 Frage nach Disk-ID
.. c224 a2 00 ldx #00
.. c226 20 cf ff jsr #ffcfcf Eingabe abwarten
.. c229 c9 0d cmp #0d
.. c22b f0 09 beq #c236
.. c22d 99 e0 c1 sta #c1e0,y ID ebenfalls abspeichern
.. c230 c8 iny
.. c231 e8 inx
.. c232 e0 02 cpx #02
.. c234 90 f0 bcc #c226
.. c236 8c df c1 sty #c1df
.. c239 a2 53 ldx #53
.. c23b 20 87 c2 jsr #c287
.. c23e 20 cf ff jsr #ffcfcf 'FROM TRACK:#' ausgeben
.. c241 85 fa sta #fa
.. c243 20 cf ff jsr #ffcfcf
.. c246 85 fb sta #fb
.. c248 a9 00 lda #00
.. c24a 85 d0 sta #d0
.. c24c a2 62 ldx #62
.. c24e 20 87 c2 jsr #c287
.. c251 20 cf ff jsr #ffcfcf 'TO TRACK:#' ausgeben
.. c254 85 fc sta #fc
.. c256 20 cf ff jsr #ffcfcf
.. c259 85 fd sta #fd
.. c25b a9 00 lda #00
.. c25d 85 d0 sta #d0
.. c25f a5 fa lda #fa
.. c261 a6 fb ldx #fb
.. c263 20 04 c4 jsr #c404 Umrechnung in HEX-Byte
.. c266 8d dc c1 sta #c1dc Anfangstrack setzen
.. c269 a5 fc ldx #fc
.. c26b a6 fd ldx #fd
.. c26d 20 04 c4 jsr #c404 Umrechnung in HEX-Byte
.. c270 8d dd c1 sta #c1dd Endetrack setzen
.. c273 ee dd c1 inc #c1dd plus 1 als Vergleichswert
.. c276 ea nop
.. c277 ea nop
.. c278 ea nop
.. c279 ea nop
.. c27a ea nop
.. c27b ea nop
.. c27c ea nop
.. c27d ea nop
.. c27e ea nop
.. c27f ea nop
.. c280 ea nop
.. c281 ea nop
.. c282 ea nop
.. c283 ea nop
.. c284 4c 93 c2 jmp #c293 weiter

.. c287 bd 4d c3 lda #c34d,x Ausgabe der Texte
.. c28a f0 06 beq #c292
.. c28c 20 d2 ff jsr #ffd2
.. c28f e8 inx
.. c290 d0 f5 bne #c287
.. c292 60 rts

.. c293 a9 0d lda #0d
.. c295 20 d2 ff jsr #ffd2
.. c29b a9 0d lda #0d
.. c29a 20 d2 ff jsr #ffd2
.. c29d a9 00 lda #00
.. c29f a2 c0 ldx #c0
.. c2a1 85 a7 sta #a7
.. c2a3 86 a8 stx #a8
.. c2a5 a9 00 lda #00
.. c2a7 a2 05 ldx #05
.. c2a9 85 a9 sta #a9
.. c2ab 86 aa stx #aa
.. c2ad a9 08 lda #08
.. c2af 20 b1 ff jsr #ffb1 LISTEN für Gerät Nummer 8
.. c2b2 a9 6f lda #6f
.. c2b4 20 93 ff jsr #fff93 15; Kommandokanal
.. c2b7 a9 4d lda #4d
.. c2b9 20 a8 ff jsr #ffa8 Programm zur Floppy senden
.. c2bc a7 2d lda #2d
.. c2be 20 a8 ff jsr #ffa8
.. c2c1 a9 57 lda #57
.. c2c3 20 a8 ff jsr #ffa8
.. c2c6 a0 00 ldy #00
.. c2c8 a5 a9 lda #a9
.. c2ca 20 a8 ff jsr #ffa8
.. c2cc a5 aa lda #aa
.. c2ce 20 a8 ff jsr #ffa8
.. c2d0 a9 1e lda #1e
.. c2d2 a9 1e lda #1e
.. c2d4 20 a8 ff jsr #ffa8
.. c2d7 b1 a7 lda (#a7),y
.. c2d9 20 a8 ff jsr #ffa8
.. c2dc c8 iny
.. c2dd c0 1e cpy #1e
.. c2df 90 f6 bcc #c2d7
.. c2e1 20 ae ff jsr #ffae
.. c2e4 18 clc
.. c2e5 a5 a7 lda #a7
.. c2e7 69 1e adc #1e

```

Listing 1.
Eine neue Formattieroutine.
Eine Diskette wird nicht
nur schneller formatiert,
sondern Sie können auch
angeben, welche Spuren
formatiert werden sollen.
Was das für Vorteile hat,
erfahren Sie im Bericht.

```

.. c2e9 85 a7 sta #a7
.. c2eb 90 03 bcc #c2ef0
.. c2ed e6 a8 inc #a8
.. c2ef 18 clc
.. c2f0 a5 a9 lda #a9
.. c2f2 a6 aa ldx #aa
.. c2f4 69 1e adc #1e
.. c2f6 85 a9 sta #a9
.. c2f8 90 02 bcc #c2fc
.. c2fa e6 aa inc #aa
.. c2fc e0 07 cpx #07
.. c2fe 90 ad bcc #c2ad
.. c300 c9 0d cmp #0d
.. c302 90 a9 bcc #c2ad
.. c304 a9 08 lda #08
.. c306 20 b1 ff jsr #ffb1 LISTEN für Gerät 8
.. c309 a9 6f lda #6f 15; Kommandokanal
.. c30b 20 93 ff jsr #fff93
.. c30e a9 4d lda #4d
.. c310 20 a8 ff jsr #ffa8
.. c313 a9 2d lda #2d
.. c315 20 a8 ff jsr #ffa8
.. c318 a9 45 lda #45
.. c31a 20 a8 ff jsr #ffa8
.. c31d a9 60 lda #60
.. c31f 20 a8 ff jsr #ffa8
.. c322 a9 06 lda #06
.. c324 20 a8 ff jsr #ffa8
.. c327 20 ae ff jsr #ffae
.. c32a a9 00 lda #00
.. c32c 85 90 sta #90
.. c32e a9 08 lda #08 Fehlermeldung holen
.. c330 20 b4 ff jsr #ffb4
.. c333 a9 6f lda #6f
.. c335 20 96 ff jsr #fff96
.. c338 20 a5 ff jsr #ffa5
.. c33b 20 d2 ff jsr #ffd2 und anzeigen
.. c33e 24 90 bit #90
.. c340 50 f6 bvc #c338
.. c342 20 ab ff jsr #ffab
.. c345 4c dc c3 jmp #c3dc Endebehandlung

.. c348 00 00 00 00 00 00 93 20 20
.. c350 20 20 20 20 20 2A 2A 2A
.. c358 20 44 49 53 4B 2D 46 4F
.. c360 52 4D 41 54 2D 53 59 53
.. c368 54 45 4D 20 2A 2A 2A 0D
.. c370 0D 0D 20 28 43 29 20 31
.. c378 39 38 35 20 42 59 20 4B
.. c380 4F 53 53 20 20 20 0D 0D
.. c388 0D 44 49 53 4B 4E 41 4D
.. c390 45 3A 20 00 0D 0D 44 49
.. c398 53 4B 2D 49 44 3A 20 00
.. c3A0 0D 0D 46 52 4F 4D 20 54
.. c3A8 52 41 43 4B 3A 2A 00 0D
.. c3B0 0D 54 4F 20 54 52 41 43
.. c3B8 4B 3A 24 00 0D 0D 41 4E
.. c3C0 4F 54 48 45 52 20 46 4F
.. c3C8 52 4D 41 54 20 28 59 2F
.. c3D0 4E 29 20 3F 20 0D 0D 00
.. c3D8 00 00 00 00 20 29 04 A2

.. c3dc 20 29 c4 jsr #c429 SAVE-Vektor stellen
.. c3df a2 6f ldx #6f
.. c3e1 20 87 c2 jsr #c287 'ANOTHER FORMAT (Y/N) ?' ausgeben
.. c3e4 20 e4 ff jsr #ffe4
.. c3e7 f0 fb beq #c3e4
.. c3e9 c9 59 cmp #59
.. c3eb d0 03 bne #c3f0
.. c3ed 4c 00 c2 jmp #c200
.. c3f0 60 rts
.. c3f1 00 brk
.. c3f2 a5 b7 lda #b7
.. c3f4 f0 03 beq #c3f9
.. c3f6 4c ed f5 jmp #f5ed
.. c3f9 20 00 c2 jsr #c200
.. c3fc a9 01 lda #01
.. c3fe a2 00 ldx #00
.. c400 a0 00 ldy #00
.. c402 10 clc
.. c403 60 rts

.. c404 85 02 sta #02
.. c406 86 03 stx #03
.. c408 a5 02 lda #02
.. c40a c9 41 cmp #41
.. c40c 90 03 bcc #c411
.. c40e 18 clc
.. c40f 69 09 adc #09
.. c411 29 0f and #0f
.. c413 0a asl
.. c414 0a asl
.. c415 0a asl
.. c416 0a asl
.. c417 85 02 sta #02
.. c419 a5 03 lda #03
.. c41b c9 41 cmp #41
.. c41d 90 03 bcc #c422
.. c41f 18 clc
.. c420 69 09 adc #09
.. c422 29 0f and #0f
.. c424 05 02 ora #02
.. c426 85 02 sta #02
.. c428 60 rts

.. c429 a9 f2 lda #f2
.. c42b 8d 32 03 stx #0332
.. c42e a9 c3 lda #c3
.. c430 9d 33 03 stx #0333
.. c433 60 rts

```

Listing 1 (Schluß).
Zwischen den Adressen
C348 und C3DA
liegt eine ASCII-Tabelle

SAVE-Vektor herstellen
auf Adresse #c3f2 setzen

