

Das WAVE-Dateiformat

WAVE-Dateien dienen zur Speicherung von digitalisierten Audio-Signalen.

Analoge Tonereignisse werden mit einem **AD-Wandler** digitalisiert. Dies geschieht mit

- einer bestimmten **Abtastfrequenz (sampling frequency)**, Anzahl der Samples pro Sekunde. Sie bestimmt die maximal erfasste Frequenz – siehe Abtasttheorem),
- einer bestimmten **Auflösung (resolution)**, wieviel bit zur Speicherung eines Ereignisses zur Verfügung stehen. Sie bestimmt den maximalen Rauschabstand der Aufnahme)
- und einer bestimmten Anzahl von **Kanälen (channels)**.

Audio-CDs enthalten pro Sekunde 44100 16-bit Zahlenwerte, je einen für jeden Stereokanal.

Berechne, wieviel MB Speicherplatz für 1 Minute CD-Sound zur Verfügung stehen muss!

(Wenn Du dies mit den Daten für Speicherplatz und Spielzeit auf CD-ROMs vergleichst, wirst Du einen Unterschied feststellen. Eine CD mit 80 Minuten Spieldauer kann nur 700MB Daten fassen. Der verringerte Speicherplatz folgt daraus, dass Computerdaten Bit für Bit exakt sein müssen – es sind aufwendige Fehlerkorrekturinformationen nötig. Bei Musik ist das eine oder andere umgefallene Bit nicht weiter tragisch.)

In Tonstudios verwendet man vor dem Endmix meist 96kHz und 24 bit. (Warum? Speicherplatz pro Minute?)

Auf MS-PCs hat sich für verlustlose Audio-Speicherung (MP3s sind mit Verlusten komprimiert) das von Microsoft festgelegte **.WAV**-Format etabliert. Im Grunde ist es aber nur eine nicht ganz kompatible Adaption des **IFF** (Interchangable File Format), das die Firma Electronic Arts als offenen Standard für den Commodore Amiga eingeführt hatte. Es sollte den Austausch beliebiger Grafik-, Text- und Musikdaten erlauben. (Du erkennst die Grundstruktur auch in den .MID-Dateien wieder)

MS nennt dieses Format allerdings etwas verfremdet **'RIFF'** (resource independent file format).

IFFs bestehen aus sogenannten **chunks** (Blöcken). Jeder Chunk beginnt mit einer 32-bit **Kennung**, gefolgt von der 32-bit **Datenlänge** (die Anzahl der nun folgenden Bytes Information), sowie (falls vorhanden) den Daten (Padding mit 0 auf gerade Byteanzahl). Kennt ein Programm eine bestimmte Kennung nicht, kann dieser Block einfach überlesen werden und die Verarbeitung setzt beim nächsten Chunk fort.

Einziges wesentlicher Unterschied: Original IFFs wurden für Computer mit Motorola Prozessoren entwickelt und speichern Integers von hi nach lo, bei MS ist es wegen der Präferenzierung von Intel-Prozessoren genau umgekehrt.

Der hauptsächlich verwendete Wave-Typ wird von MS als **PCM** (pulse code modulation) bezeichnet. Das ist allerdings falsch. PCM stammt aus der Welt der Analogsynthesizer und bedeutet die Modulationsmöglichkeit der Pulsbreite von Rechteckswellen. Damit lassen sich mit wenig Aufwand interessante Veränderungen im Obertonreichtum der Wellen erzielen (die berühmten 'fetten Sounds' von Moog, Oberheim usw.). Wenn Du das WAV-Format durchliest, wirst Du Dich wohl ebenfalls fragen, wo hier denn etwas moduliert wird...

Abkürzungen:

B 8-bit Byte: kommt nur in den Daten vor.
 W 16-bit Word (lo->hi!)
 DW 32-bit DoubleWord (lo->hi)

DW	'RIFF'	Kennung
DW	nDataBytes	Länge aller folgenden Daten (9*4+Audiodaten)
DW	'WAVE'	Typ des Inhalts
DW	'fmt '	Kennung Parameterblock (Space bei [3])
DW	nDataBytes	Datenlänge = 16
W	formatTag	Kodierung: PCM = 1
W	nChannels	Anzahl Kanäle: Mono=1, Stereo=2,...
DW	nSamplesPerSec	Samplingrate
DW	nAvgBytesPerSec	nötige Bytes für 1 Sekunde Audiodaten (alle Kanäle)
W	nBlockAlign	nötige Bytes für einzelne Samplinginformation (alle Kanäle)
W	nBitsPerSample	Auflösung für 1 Sample eines Kanals (für alle gleich)
DW	'data'	Kennung zugehöriger Datenblock
DW	nDataBytes	Datenlänge
.....	...<sample><sample>...	die eigentlichen Audio-Daten

Wie Du siehst, sind hier eigentlich mehr Informationen als unbedingt nötig untergebracht. Tatsächlich sollen Einträge wie nAvgBytesPerSec auch nur dem Abspielprogramm helfen, rasch den benötigten Puffer-Speicherplatz zum Zwischenlagern der Datenstücke bestimmen zu können, die dann dem Betriebssystem zum DA-Wandeln übergeben werden. Da WAVE-Dateien (fast) beliebig groß sein dürfen (was geht sich in einem DW aus?), können sie ja nicht gesamt ins RAM geladen werden, sondern müssen in Teilen verarbeitet werden, ohne den Zuhörer die Schnittstellen wahrnehmen zu lassen (Buffer-Switching Techniken).

Sampleformat:

1-8 bit: unsigned integers, Stille = \$80
 9- bit: signed integers, Stille = 0, lo/hi bzw. lo/mid/hi zerlegt.

Bei 'ungeraden' Auflösungen sind die weniger(!) signifikanten bits des Datenbytes/wortes mit Nullen zu füllen!

Reihenfolge:

bei Stereo: erstes Sample ist links, erstes rechts, dann zweites links, zweites rechts,...

Erweiterung:

In eigenen Chunks könnten auch noch Informationen über 'Playlists' und 'Cuepoints' stehen.

Abtasttheorem:

Um eine Schwingung mit Frequenz f sinnvoll abzutasten, muss die Samplingrate mehr als das Doppelte dieser Frequenz ausmachen. (Bsp: Taste die Funktion $\sin(2\pi \cdot t)$ an den Stellen $0, \pi, 2\pi, 3\pi, \dots$ ab. Das entspricht genau der doppelten Frequenz und liefert statt der Schwingung nur konstant Null).

Faustregel: Sampling mit $f \rightarrow$ maximale Grenzfrequenz $f/2$

ADAT (Alesis, 1991)

Dies ist ein digitales Format zur Band(Cassetten-)aufzeichnung. (Um es nebenbei möglichst schwer zu machen, CDs verlustlos auf billiges Band zu kopieren, wählte man eine Abtastrate von 48kHz.) Obwohl damals massiv für die Heimanwendung beworben, heute nur noch in Tonstudios gebräuchlich (gute Mischpulte, Effektgeräte und Synthesizer liefern zusätzlich zum analogen Output digitale Signale im ADAT Format).