

## **Aufbau einer Microsoft Wav-Datei (PCM)**

### **RIFF Chunk (Länge 12 bytes)**

Byte Nummer

- 0 – 3 „RIFF“ geschrieben (ASCII)
- 4 – 7 Gesamtlänge der Datei (uInt32)
- 8 – 11 „WAVE“ geschrieben (ASCII)

### **FORMAT Chunk (Länge 24 bytes)**

Byte Nummer

- 0 – 3 Name, „fmt\_“ geschrieben (ASCII)
- 4 – 7 Länge des FORMAT CHUNKS (uInt32, 0x10)
- 8 – 9 WAV-FORMAT-TAG (Kompression) (uInt16)  
0x01 – unkomprimiertes Audio, andere Werte – komprimiertes Audio bzw. Kompressionsverfahren
- 10 – 11 Anzahl der Kanäle (uInt16) (0x01 – mono, 0x02 – stereo)
- 12 – 15 Sample rate (uInt32, in Hz)
- 16 – 19 NAvgBytesPerSec - Bytes pro Sekunde (uInt32)
- 20 – 21 Bytes pro Sample (1= 8 bit mono, 2= 8 bit stereo, 3 = 16 bit mono, 4 = 16 bit stereo) (uInt16)
- 22 - 23 Auflösung, Bits pro Sample (uInt16)

### **RGAD Chunk (optional, neuere Erweiterung)**

Byte Nummer

- 0 – 3 „rgad“ geschrieben (ASCII)
- 4 – 7 Länge des RGAT CHUNKS (uInt32)
- 8 – 11 fPeakamplitude - Übersteuerungspunkt des Audiodaten (float32)
- 12 – 13 NRadioRgAdjust - Pegelkorrektur für Radioqualität Wiedergabe (uInt16)
- 14 - 15 NAudiophileRgAdjust – Pegelkorrektur für höchstqualitative Audiowiedergabe (uInt16)

### **DATA Chunk (Länge abhängig von der Länge des Audiomaterials)**

Byte Nummer

- 0 – 3 „data“ geschrieben (ASCII)
- 4 – 7 Länge der folgenden Audiodaten (uInt32)
- 8 – Ende Audiodaten (Samples als signed integer little endian\*) (uInt16)

Beispiel einer wav-Datei in HEX (rgad chunk nicht vorhanden):

Ding.wav:

```

246E:0100  52 49 46 46 46 2D 00 00-57 41 56 45 66 6D 74 20  RIFFF...WAVEfmt
246E:0110  10 00 00 00 01 00 01 00-22 56 00 00 22 56 00 00  ..... "V.. "V..
246E:0120  01 00 08 00 64 61 74 61-22 2D 00 00 80 80 80 80  ....data"-.....
246E:0130  80 80 80 80 80 80 80 80-80 80 80 80 80 80 80 80  .....
246E:0140  80 80 80 80 80 80 80 80-80 80 80 80 80 80 80 80  .....

```

1. rot - "RIFF" ASCII
2. blau - 0x2D46 Dezimal 11590 Länge der Datei abzüglich den 8 Bytes für „RIFF“ und der Länge
3. rot - WAVEfmt ASCII
4. orange - fmt ASCII
5. grün - 0x00000010 Länge des Format Chunks, immer konstant
6. violett - 0x0001 Dezimal 1 wav-format tag
7. hellgrün - 0x0001 mono Datei
8. blau - 0x00005622 Dezimal 22050 Sample Rate
9. rosa - 0x00005622 Dezimal 22050 Bytes per Second (hier identisch, da mono 8Bit, für 16 Bit stereo 4x die Sample rate)
10. rot - "Data" ASCII die Audiodaten folgen
11. grün - 0x00002D22 Dezimal 11554 Länge der Audiodaten
12. schwarz Audiodaten

\*

### "Little Endian" (Computer mit Intel-Prozessoren)

Das niederwertigste Byte eines Wertes belegt die niederwertigste Speicheradresse. Die höherwertigen Bytes belegen die höherwertigen Speicheradresse (the little end comes first).

Ein 4 Byte LongInt Wert wird wie folgt gespeichert:

LongInt - Wert:      Byte3 Byte2 Byte1 Byte0

Speicheradresse	Base Address+0	Byte0
	Base Address+1	Byte1
	Base Address+2	Byte2
	Base Address+3	Byte3

### "Big Endian" (Computer mit Motorola Prozessoren)

Das höchstwertige Byte wird auf der niederwertigsten Speicheradresse abgelegt, die niederwertigen Bytes entsprechend auf den höherwertigen Speicheradressen (the big end comes first). Derselbe LongInt –Wert würde wie folgt gespeichert.

Speicheradresse	: Base Address+0	Byte3
	Base Address+1	Byte2
	Base Address+2	Byte1
	Base Address+3	Byte0