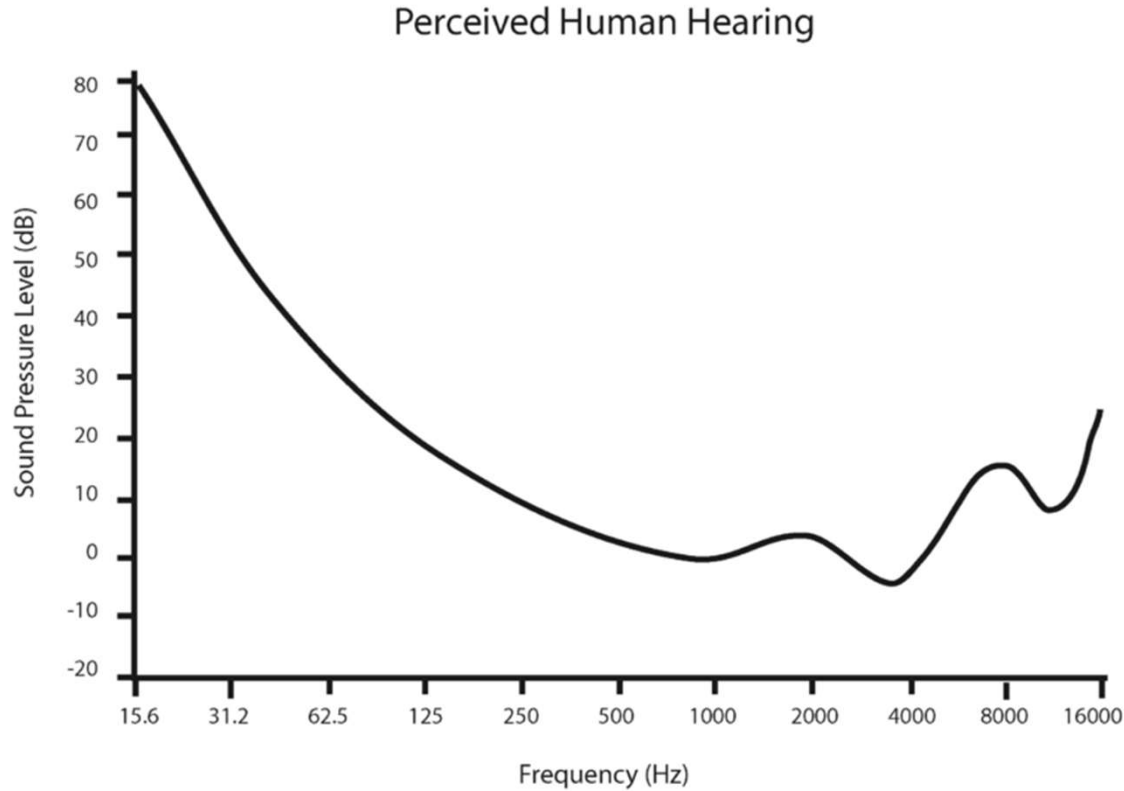


- mittlere Datenrate wird konstant gehalten
  - Qualität variiert bildinhaltsabhängig
- Qualität beeinflussbar durch
  - Güte der Bewegungskompensation
    - sorgfältige Bewegungskompensation >>> hohe Kompression
  - Feinheit der Quantisierung, Unterdrückung von Koeffizienten
    - feine Quantisierung >>> weniger Artefakte, geringere Kompression
  - Verhältnis I / P / B - Bilder
    - mehr I-Bilder >>> bessere Editierbarkeit, geringere Kompression

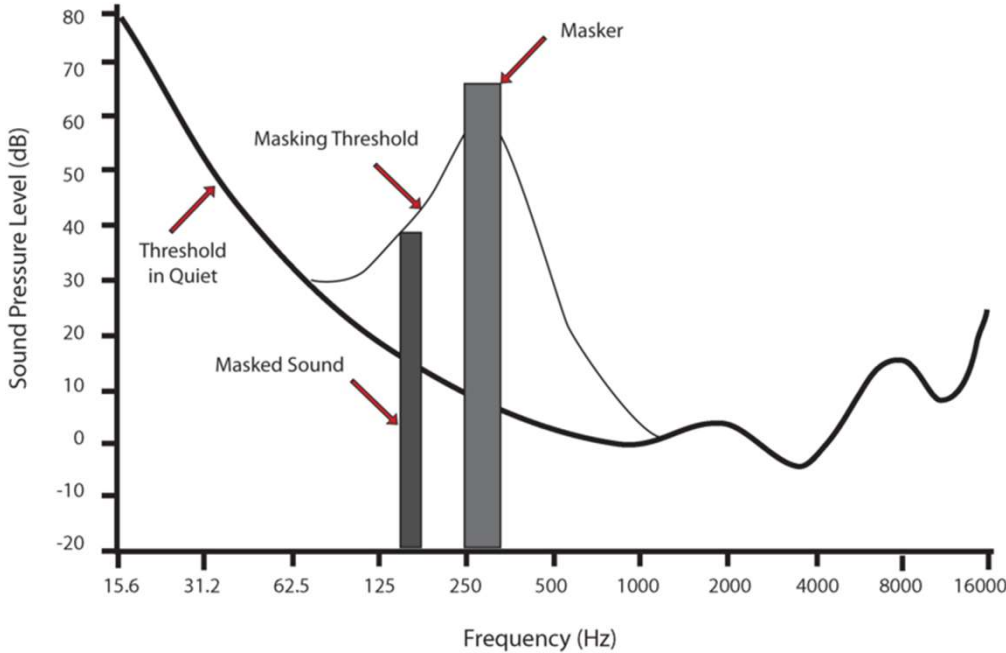
- Modellierung des Hörempfindens



File:Perceived Human Hearing.png  
From Wikipedia, the free encyclopedia

Abhängig von der Frequenz ist unterschiedlich viel Energie notwendig, um einen hörbaren Ton zu erzeugen

- Modellierung der Maskierung

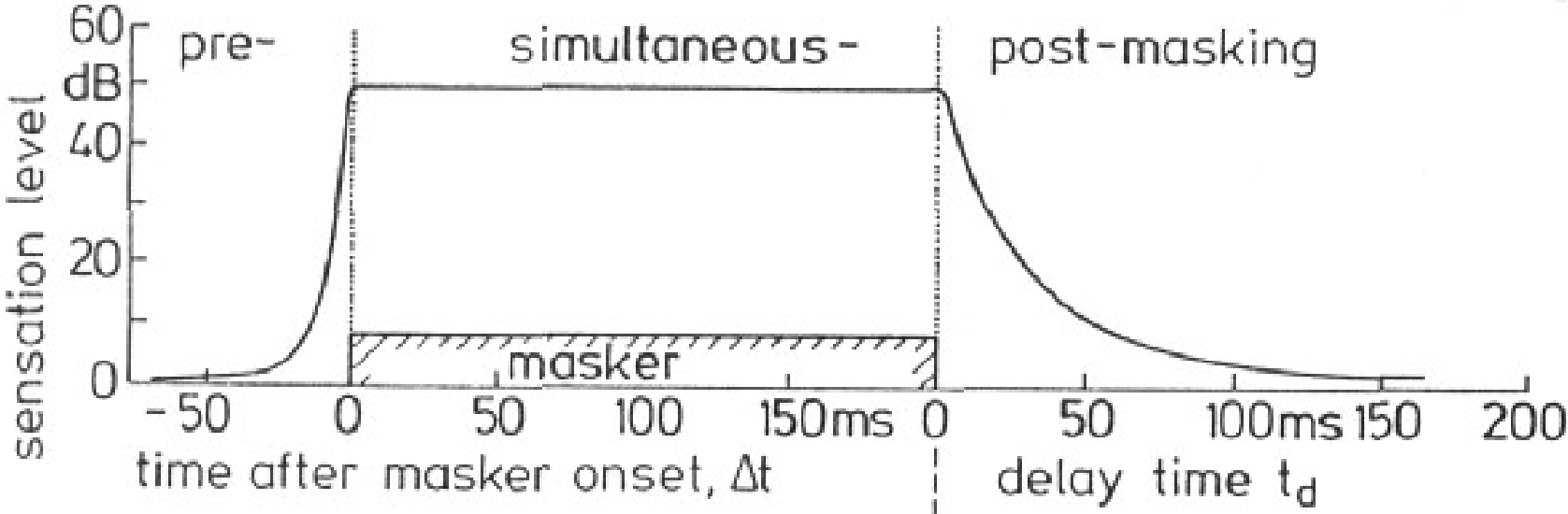


Laute Frequenzen übertönen leise benachbarte.  
Sie wirken als “Masker” und verändern die Wahrnehmbarkeitsschwelle

File:File:Audio Mask Graph.png  
From Wikipedia, the free encyclopedia

- Modellierung des Hörempfindens, Maskierung

Illustration: Zwicker, Fastel: Psychoacoustics

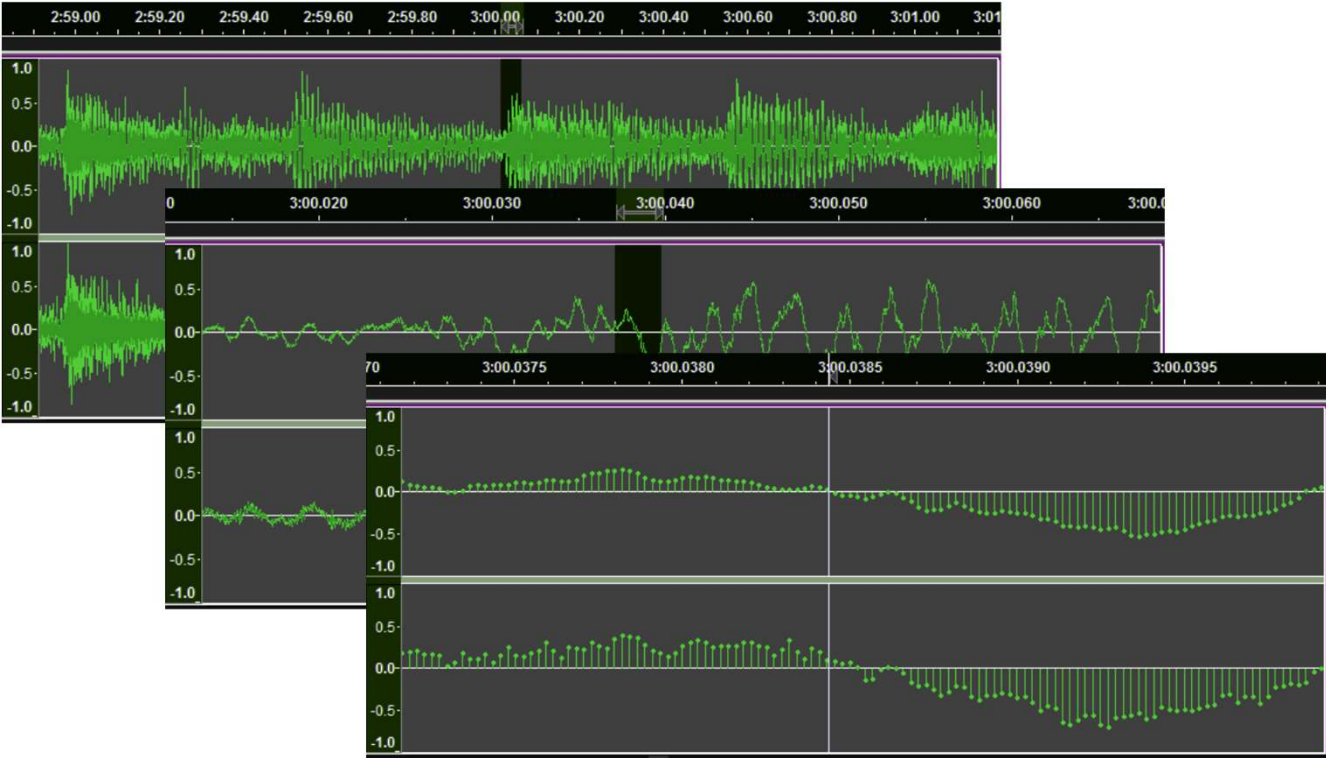


Auch auf der Zeitachse treten entsprechende Maskierungen auf

- Charakteristiken
  - Abtastrate (Hz) → **maximale Frequenz**
  - Auflösung (Bit) → **Dynamikumfang (leise ... laut)**
  - Zahl der Kanäle → **räumlicher Klangeindruck (Mono ... Stereo ... Surround 5.1 ... 7.1)**
  - Spieldauer (s)
  - Dateigröße (Byte)
  - Datenrate (kBit/s)

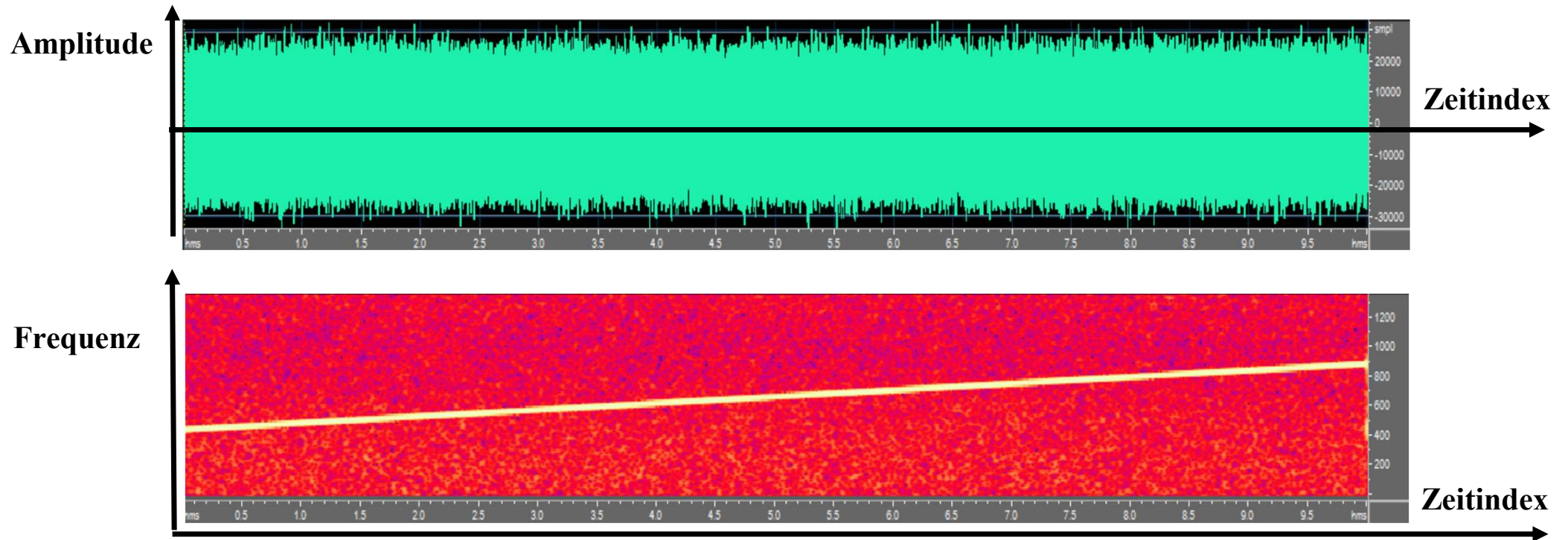
# Digitale Audiodaten

- Beispiel: dekodierte Daten
  - 2 Sekunden
  - Abtastrate 44,1 kHz
  - Auflösung 16 Bit
  - stereo
- = 176.400 Integerwerte  $\approx$  1/3 MB



# Spektrogramm

- Darstellung im „Zeitbereich“ versus „Frequenzbereich“

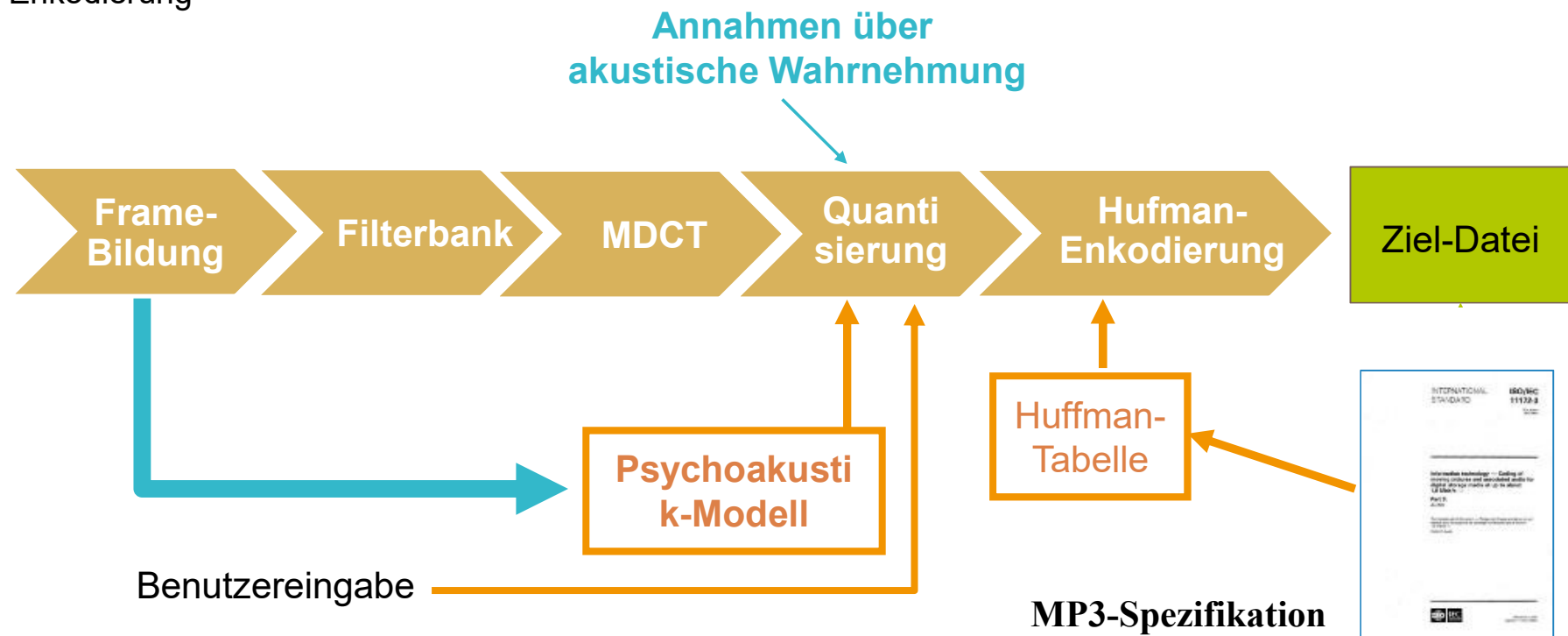


- weitere Unterscheidungskriterien:
  - Codecs optimiert für Musik oder Sprache?
  - Speicherformate: streamingbasiert und/oder dateibasiert?

| Medientyp | verlustbehaftete Codecs                     | verlustfreie Codecs     | System / „Container“       |
|-----------|---|-------------------------|----------------------------|
| Audio     | MP3, AAC, WMA, MP2, Vorbis, AMR, GSM, G.72x | xPCM, FLAC, MPEG-4, WMA | RIFF / WAV, MP4, AIFF, OGG |



- MP3-Enkodierung



- kleinste abspielbare Einheit: „MP3-Frame“
- jedes Frame = 2 x 576 Samples  $\triangleq$  ca. 1/40 Sekunde (long blocks)
- Aufbau eines Frames:
  - header (4 Byte)
  - error\_check (CRC-16 = 2 Byte)
  - audio\_data
    - sideinfo (17 Byte bei Mono, 32 Byte bei Stereo)
    - main\_data (variable Länge)
  - ancillary\_data (variable Länge)



## Header

| Field              | Length | Unit | Value / Description    | Notes                                   |                    |
|--------------------|--------|------|------------------------|---|--------------------|
| syncword           | 12     | bits | „1111 1111 1111“ (fix) | entspricht 0xFFFB                       |                    |
| ID                 | 1      | bit  | meistens „1“           | entspricht „MPEG-1“                     |                    |
| layer              | 2      | bits | immer „01“             | entspricht „Layer III“                  |                    |
| protection_bit     | 1      | bit  | meistens „1“           | bedeutet: „kein CRC dabei“              |                    |
| bitrate_index      | 4      | bits | 0-14                   | entspricht 32, 40, 48, ... , 320 kBit/s |                    |
| sampling_frequency | 2      | bits | 0-2                    | entspricht 32 / 44.1 / 48 kHz           | darf variabel sein |
| padding_bit        | 1      | bit  |                        |   |                    |
| private_bit        | 1      | bit  |                        |   |                    |
| mode               | 2      | bits | 0-3                    | entspricht „stereo“, „mono“ usw.        |                    |
| mode_extension     | 2      | bits |                        | müssen konstant sein                    |                    |
| copyright          | 1      | bit  |                        |   |                    |
| original/home      | 1      | bit  |                        |   |                    |
| emphasis           | 2      | bits |                        |   |                    |

# MP3 Frames

- Startcode: häufig „0xFFFB“
- Sideinfo: 32 Byte
- Main\_data:  
ca. 100-1500 Byte

Coralreef.FullHD.WMV3\_mp4 Startup SL0032-07 Afro Blue.mp3 x

```

Edit As: Hex Run Script Run Template: MP3.bt
0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
1:1C50h: FF FB D2 44 76 8C C6 87 68 CF 9B 99 4B 70 C1 AD yûôïvEE+hï>™KpÁ-
1:1C60h: A 03 6F 29 4E 18 DD A3 40 6E 65 89 C3 36 B4 67 ..o)N.Ýf@ne%Ã6'c
1:1C70h: D CC A5 78 48 93 61 00 10 83 6F 1A C0 A1 DD 5B .ìÿxH"a..fo.À;Ý[
1:1C80h: 07 13 88 AA 68 5B 59 4C 90 51 54 6D 01 BA 43 4C ..^*h[YL.QTm.°CL
1:1C90h: 68 48 15 EA B1 2D 4B A9 9B F9 C8 69 FF 6B 3B A3 hH.ê±-K©>ùÈiÿk;£
1:1CA0h: DD 2D DE 00 02 05 16 42 ED DA 5A 45 78 01 36 0E ...3/ D&ÚEY 6
  
```

Template Results - MP3.bt

| Name                          | Value | Start  | Size | Color   | Comment |
|-------------------------------|-------|--------|------|---------|---------|
| > struct MPEG_FRAME mf[22]    |       | 1190Ch | 836  | Fg: Bg: |         |
| ▼ struct MPEG_FRAME mf[23]    |       | 11C50h | 836  | Fg: Bg: |         |
| ▼ struct MPEG_HEADER mpeg_hdr |       | 11C50h | 4    | Fg: Bg: |         |
| uint32 frame_sync : 12        | FFFh  | 11C50h | 4    | Fg: Bg: |         |
| uint32 mpeg_id : 1            | 1h    | 11C50h | 4    | Fg: Bg: |         |
| uint32 layer_id : 2           | 1h    | 11C50h | 4    | Fg: Bg: |         |
| uint32 protection_bit : 1     | 1h    | 11C50h | 4    | Fg: Bg: |         |
| uint32 bitrate_index : 4      | Dh    | 11C50h | 4    | Fg: Bg: |         |
| uint32 frequency_index : 2    | 0h    | 11C50h | 4    | Fg: Bg: |         |
| uint32 padding_bit : 1        | 1h    | 11C50h | 4    | Fg: Bg: |         |
| uint32 private_bit : 1        | 0h    | 11C50h | 4    | Fg: Bg: |         |
| uint32 channel_mode : 2       | 1h    | 11C50h | 4    | Fg: Bg: |         |
| uint32 mode_extension : 2     | 0h    | 11C50h | 4    | Fg: Bg: |         |
| uint32 copyright : 1          | 0h    | 11C50h | 4    | Fg: Bg: |         |
| uint32 original : 1           | 1h    | 11C50h | 4    | Fg: Bg: |         |
| uint32 emphasis : 2           | 0h    | 11C50h | 4    | Fg: Bg: |         |
| > ubyte mpeg_frame_data[832]  |       | 11C54h | 832  | Fg: Bg: |         |
| > struct MPEG_FRAME mf[24]    |       | 11F94h | 836  | Fg: Bg: |         |

- Besonderheit: Abstand zum nächsten Frame steht nicht explizit im Header!
  - muss berechnet werden
  - $L \approx 144 \cdot \text{Bitrate} / \text{Abtastfrequenz}$
- Beispiel: angenommen MP3 mit 256 kBit/s, 44.1 kHz
  - → Länge eines Frames = 836 Byte
- mögliche Werte:  $L = 96 \dots 1.440$  Byte

© ISO/IEC

- Typische MP3-Datei „mymusic.mp3“

