

# Videos mit variabler Bildrate

□

## Inhaltsverzeichnis

- [1 Einblenden des Zeitstempels mit FFmpeg](#)
  - [1.1 Zeitstempel in Millisekunden](#)
  - [1.2 Zeitstempel in Sekunden](#)
- [2 AVIdemux als Alternative](#)

## Einblenden des Zeitstempels mit FFmpeg

### Zeitstempel in Millisekunden

Die Stapelverarbeitungsdatei, die den Zeitstempel mittel [FFmpeg](#) ins Video schreibt, besteht aus einer einzigen Zeile:

```
"%ProgramFiles%\FFmpeg\bin\ffmpeg" -y -i %1 -vf  
"drawtext=fontsize=h/8:fontcolor=white:x=0.1*tw:y=h*0.5-  
th:text='%%{eif\:1K*t-878\:d\:04} ms' " "%~dpn1_tc%~x1"
```

- **"%ProgramFiles%\FFmpeg\bin\ffmpeg"** spezifiziert den genauen Pfad zu FFmpeg, damit genau diese Version gestartet wird (und nicht irgendeine andere, die sich in einem der vorgegebenen Pfade finden lässt)
- **-y** überschreibt das Ausgabevideo ohne Rückfrage, falls es schon vorhanden ist
- **-i %1** Eingabevideo wird als Parameter an das Skript übergeben
- **-vf "drawtext ...** erledigt die eigentliche Arbeit
- **fontsize=h/8:fontcolor=white** weiße Schrift, ein Achtel der Bildhöhe  $h$  groß
- **x=0.1\*tw:y=h\*0.5-th** x-Position ein Zehntel der Textlaufweite  $tw$  vom linken Bildrand entfernt, y-Position auf halber Bildhöhe  $h$  - Texthöhe  $th$
- **text='%%{eif\:1K\*t-878\:d\:04} ms'** Das Tausendfache  $1K$  des Zeitstempels  $t$  (in s) als Ganzzahl  $eif$  mit vier Stellen  $\:d\:$  und dem Anhängsel  $ms$  als Text formatieren. Die 878 ist ein willkürlicher Offset, der die Zeitanzeige zu Beginn des interessierenden Intervalls nullt.
- **"%~dpn1\_tc%~x1"** Name der Ausgabedatei = Eingabedatei mit Anhängsel  $_tc$  für »timecode«

Bei Videos mit variabler Bildrate (vfr) kann es erforderlich sein, vor dem Videofilter (-vf) die Anweisung

```
-vsync 2
```

einzufügen, mit der das Video explizit zu vfr erklärt wird.

Mit der letzten Angabe wird das Containerformat, z. B. mp4 oder mov, für die Ausgabedatei übernommen. Audio- und Videocodec werden, da nicht weiter spezifiziert, automatisch gewählt. Für das Video bedeutet dies bei diesen Containertypen AVC (H.264).

Hier eine Datei (Matroska Video-Container), um das Skript zu testen: [80-80-40.mkv](#)

## Zeitstempel in Sekunden

Der Codebestandteil

```
text='...'
```

entscheidet darüber, was wie angezeigt wird. Die einfachste Art, die Zeit in Sekunden auszugeben, ist

```
text='%{%pts}'
```

dann wird der *presentation time stamp* ausgegeben - mit sechs Nachkommastellen und ohne Möglichkeit, dies zu beeinflussen.

Es gibt Anfang 2021 keine Möglichkeit, eine Gleitkommazahl formatiert auszugeben. Man kann sich die Ausgabe nur aus Ganzzahlen zusammenbasteln, indem man die Zeit in den ganzzahligen Anteil und die Nachkommastellen zerlegt:

```
text='\%{eif\:t\:d}" , "%{eif\:(t-trunc(t))*1000\:d\:3} s'
```

Die Formatierung durch *eif* schneidet Nachkommastellen per se ab, sodass die Operation vor dem (Dezimal)komma die ganzen Sekunden ergibt. Die Operation

```
t-trunc(t)
```

liefert die Nachkommastellen, die mit 1000 multipliziert werden, um Millisekunden zu erhalten.

## AVIdemux als Alternative

Das freie Programm AVIdemux ist eines der wenigen Programme, die vfr-Video halbwegs korrekt behandeln. AVIdemux verfügt über einige rudimentäre Filter, u.a. über einen, der die Bildnummer und den Zeitstempel (*presentation time stamp*) ins Bild einbrennt:

[http://wiki/Datei:80\\_80\\_40.mp4](http://wiki/Datei:80_80_40.mp4)