

SOX

はじめの一步

- 録音 (rec コマンド)

```
$ rec hoge.wav
```

- 再生 (play コマンド)

```
$ play hoge.wav
```

- そんなだけ！
- sox ツールと言いつつコマンド名は play/rec. 本物の (?) sox コマンドは後で登場します

sox についてこれ以上何を知る？

- sox は色々な符号化形式をサポートしている
 - その中の最もストレートな形式 (raw 形式, ヘッダレス) については, 中身を理解して自分で読めるように (Linear PCM)
- raw 形式でもいくつかのパラメータがある
 - 各標本の bit 数, 符号化
 - チャンネル数 (モノラル, ステレオ, ...)
 - 標本化周波数
- 入力, 出力先の変更方法 (標準入出力)

色々な符号化形式があるということ

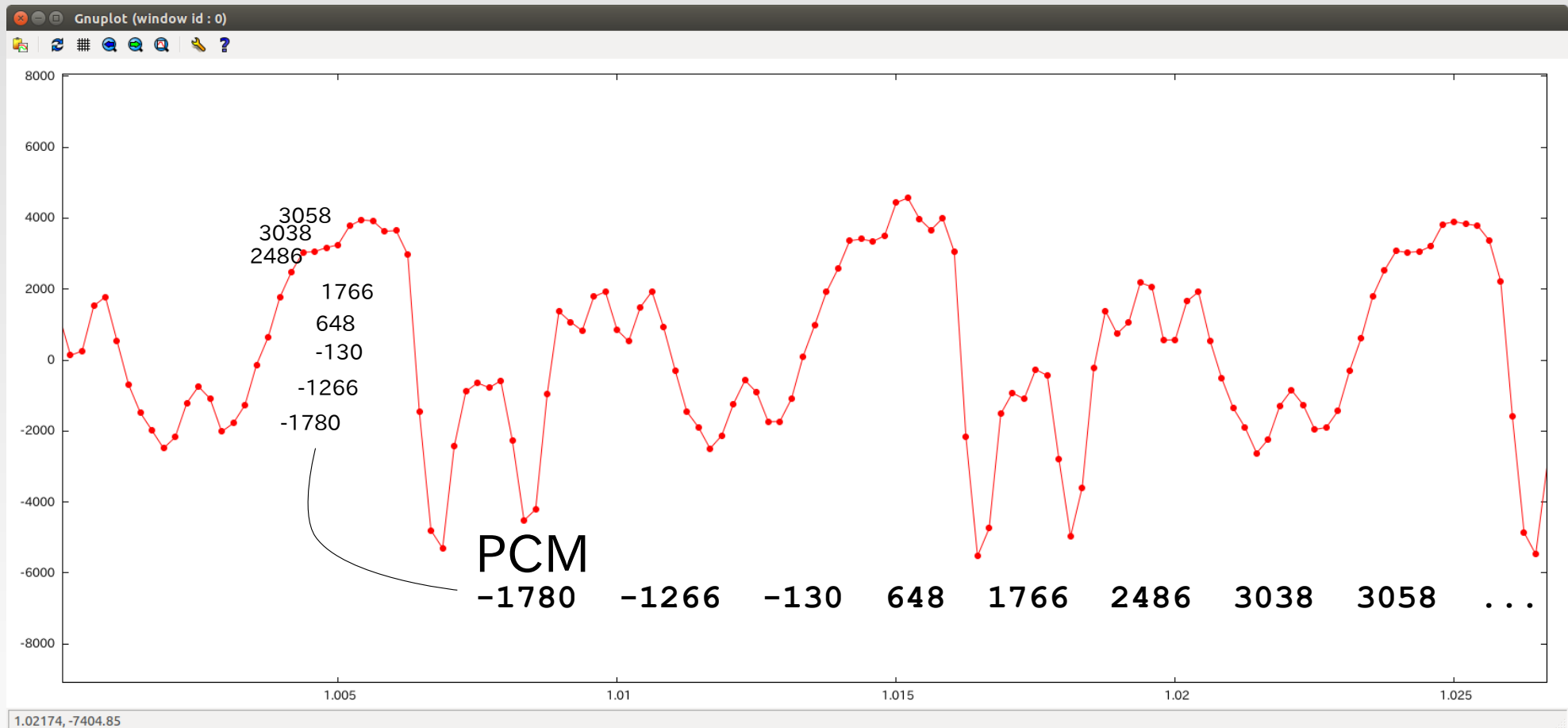
- raw, wav, mp3, ogg, aac, ...
 - 「波形」である音をどういうビット列で表すかの方法
- sox は , ファイルの拡張子 + デフォルトルールで形式を選びます

```
$ rec hoge.wav    # .wav だから .wav 形式ね  
$ rec hoge.raw    # .raw だから .raw 形式  
$ rec hoge.mp3    # .mp3 形式
```

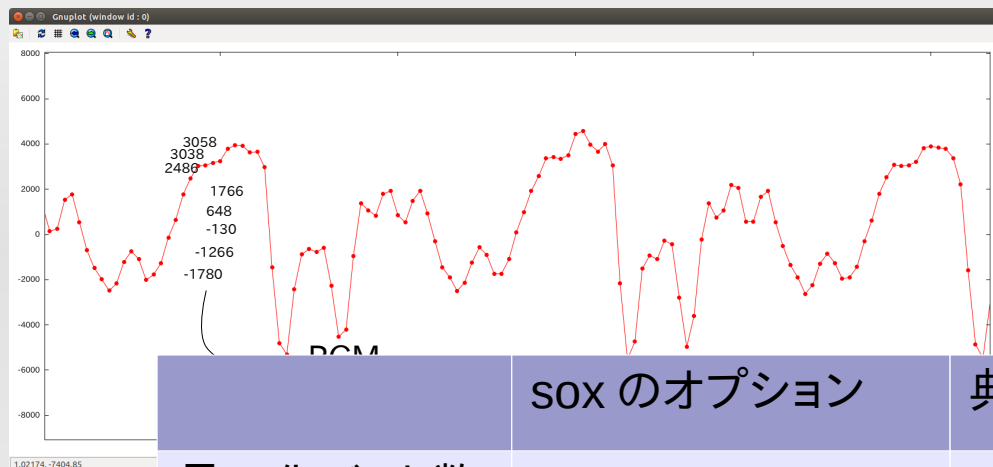
- 同じくらいの秒数でも形式によってサイズ (圧縮率) が全然違うのを見てみよ

最弱・最単細胞形式として raw 形式なるものがあるということ

- raw 形式 : 波形データの値を並べただけ (PCM)



raw 形式でも複数の符号化形式調節可能なパラメータがあるということ



	sox のオプション	典型値	rec の規定値
量子化ビット数	-b	8, 16	16
チャネル数	-c	1, 2	2
標本符号化	-e	s, un	s
標本化周波数	-r	8000, 44100, 48000	48000

- 注 : **s** = signed integer, **un** = unsigned integer
- raw 形式は「データ」そのもののしか含まないためそれらのパラメータは別途与えないといけない

要するに今日一番大事なのは **raw** 形式を自分の **C** プログラムで読み書きできるようになることです!

- raw 形式は余計なものがないので簡単
- 例 1: 8 bit unsigned integer モノラルなら
 - `unsigned char data[N];` // char は 8 bit
 - `read(fd, data, N);`
 - 符号のある波形にするには , 128 を引く
- 例 2: 16 bit signed integer, モノラルなら
 - `short data[N];` //short は 16 bit
 - `read(fd, data, sizeof(short) * N);`

半分余談

- .wav
 - メタデータヘッダ + raw 形式
 - メタデータには、データ部分の量子化ビット数、チャンネル数、標本化周波数などが記述されている
 - つまりファイルを渡されたただけで正しく再生できる
 - メタデータ形式も簡単なのでちょっと調べれば自分で読み書きできます
 - raw 形式にはヘッダすらないので自分で -b, -c, -e, -r を指定しないといけない
 - いわゆる音楽 CD の形式 (16 bit ステレオ, 44100Hz)

半分余談

- mp3, aac, ogg とか
 - 圧縮形式
 - 波形を周波数ごとに分解
 - 人間の耳に聞こえにくい周波数は粗い精度 (少ない bit 数) で符号化したり, 大きく鳴っている周波数のそばは聞こえにくいことを利用したりするらしいです

入出力先の変更

- 電話を作るのに，
 - rec で一旦ファイルに保存して，またそれを改めて読みだすというのは冴えない ...
 - 受け取った音を一旦ファイルに保存して play にそれを読ませるのも同様に冴えない
- rec が読みだした端から受け取りたい；受け取った端から play に渡したい
 - rec には出力を標準入力，play には入力を標準入力にする機能がある
 - パイプでつなげば良い！

- rec に標準出力に出力させる

```
$ rec -t raw -
```

- 注意：ファイル名から形式を決めてくれなくなるので，-t で指定する（標本化周波数などは規定値）
- 端末に読めない文字が表示される
- play に標準入力から読ませる

```
$ rec -t raw -r 44100 -e s -b 16 -c 1 - < hoge.raw
```

- こちらはあらゆるパラメータの指定が必須（間違ったパラメータは危険だから？）

つまりゆくゆくはこんなことをするでしょう

- 電話

```
$ rec -t raw - | ./g1g2g3_phone お相手 | play -t raw ... -
```

- rec がマイク代わり, play がスピーカ代わり
- 自分が作る電話の本体 (g1g2g3_phone) は,
 - こちら側の音を標準入力から受け取り,
 - 相手から受け取った音を標準出力に出す

sox コマンドって？

- 入力も出力も自在にかえられるのが sox

```
$ rec hoge.wav          # マイク → hoge.wav  
$ play hoge.wav         # hoge.wav → スピーカ  
$ sox hoge.wav hoge.hoge.wav # hoge.wav → hoge.hoge.wav
```

- rec/play とも sox の縮退版ともいえる
- rec : 入力がマイクに限定
- play : 出力がスピーカに限定
- sox の一般形

```
$ sox [ 入力形式オプション ] 入力 [ 出力形式オプション ] 出力
```

- 入出力形式オプションは rec, play, sox 全て共通

- rec, play で様々な形式での録音 (再生) が可能なように, sox では様々な形式「間」の変換が可能

```
$ sox hoge.wav hoge.mp3 # .wav を .mp3 に変換
```

- 地味 CD リッピングソフトができちゃった
- 標本化周波数の変更とかも朝飯前

```
$ sox -t raw -b 16 -c 1 -e s -r 44100 in.raw ( 次行へ続く )  
-t raw -b 16 -c 1 -e s -r 8000 out.raw
```

- コマンドライン長すぎワロタ? → 色々覚えて (詳しくは教科書)