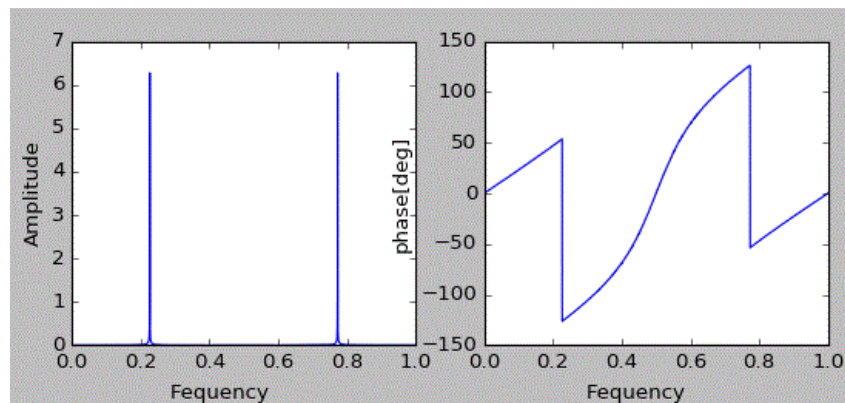


ステレオのWAVファイルのサンプリング周波数をFFTを使って2倍に変換するものをで作ってみた。  
FFT(高速フーリエ変換)によるサンプリング周波数変換の方法はWikipediaでも紹介されて珍しいものではない。

デジタルフィルタを使った方法では、元のサンプル点でもその値は変化してしまう可能性があるが、  
FFT法では理論的に変換された値も元のサンプル点では同じになるはずである。(補間される、元のサンプル点の間にあるポイントは 新たに計算される。)

零データを挿入して逆変換で実数に戻すところで 少しつまづいたので メモしておく。

(虚部がない) 実数をフーリエ変換すると、下図の例のように、サンプリング周波数の $1/2(=0.5)$ を中心に振幅は対称に、位相は反対称になる。



逆に、逆フーリエ変換をして(虚部が零の)実数に戻すため、この振幅対称と位相反対称を満たすように、零点を内挿する必要がある。(多分、必要十分条件ではないだろう。)

Nサンプルをフーリエ変換した値は、Nサンプルの虚数になるが、はじめの値は直流に相当する値で、残りのN-1個が交流に相当する。

サンプリング周波数を2倍にするためには、折り返し中間点の間に、零をN個挿入すればできそうだが、実際にはうまくいかない。

そこで、零をN-1挿入して、零挿入した前後のポイントの値を折り返し点の値の半分の値にすることにした。

下記は 入力と2倍にアップサンプリングした値を順番に表示した例である。出力は入力のため、入力側はひとつとびになっている。

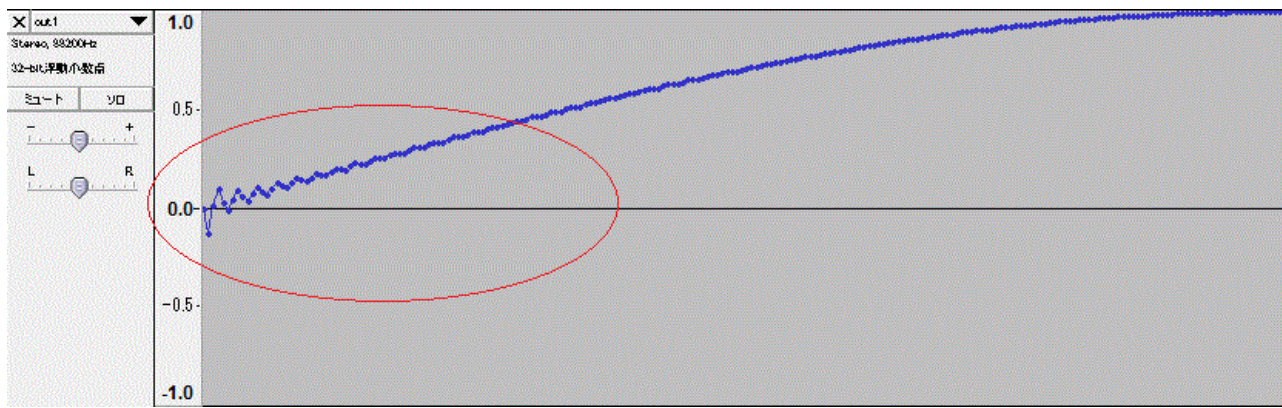
実部は元のサンプル点と同じで、虚部が(ほとんど)零の値になった。(10の-12乗は16ビット整数に対して無視できる。計算精度は有限なので 計算誤差が発生し完全な零にはならない。)

入力 サンプリングタイムを2倍に変換した出力(複素数)  
2バイト整数

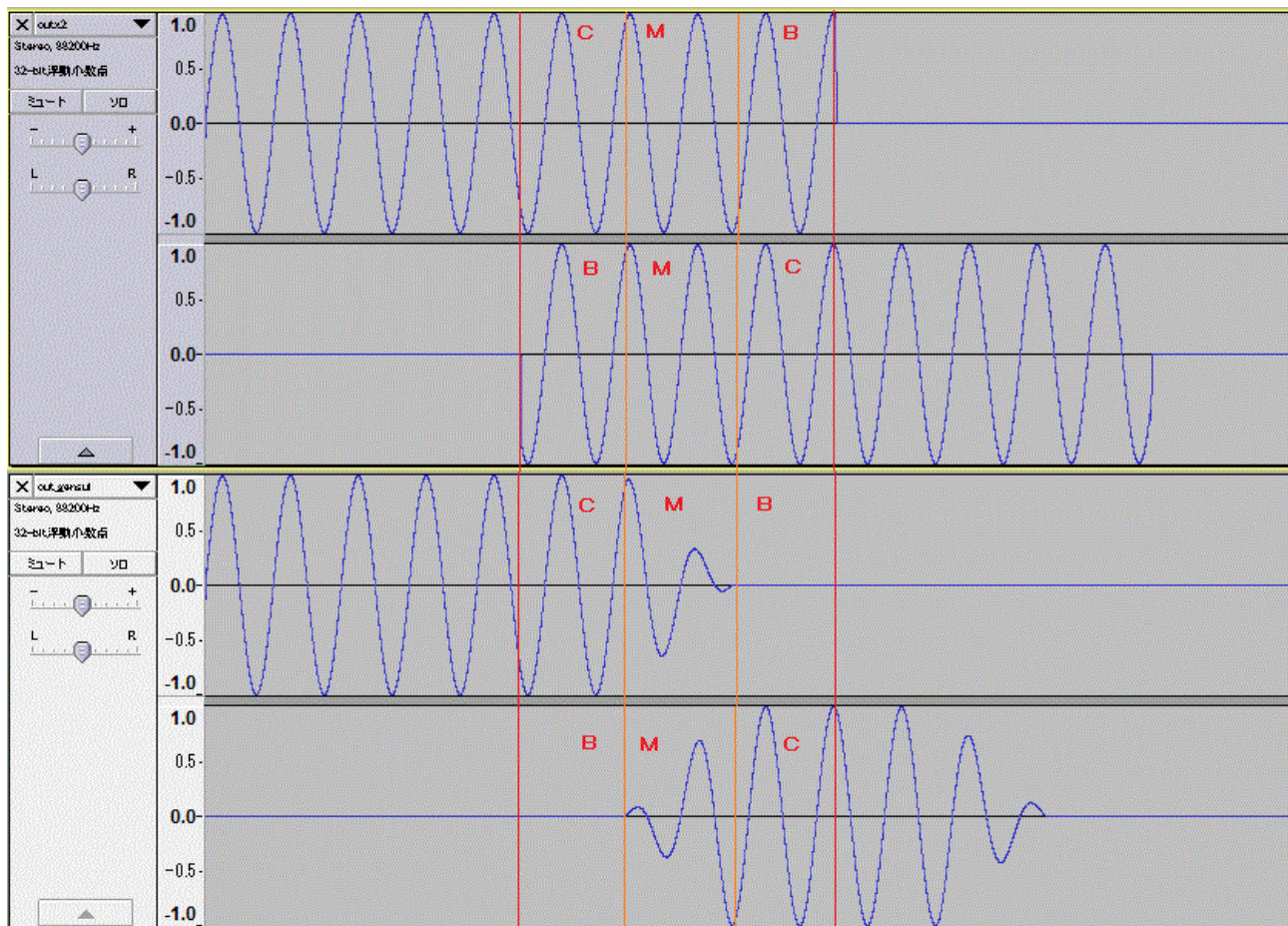
```
0 0j
(5277.45188367-3.54072327013e-12j)
4653 (4653-3.13526982154e-13j)
(5312.77115817-3.00204305859e-13j)
9211 (9211-6.0154442212e-12j)
(12548.8376043-1.17287058895e-12j)
13583 (13583-6.77680134231e-13j)
(14818.7340917-3.93773902374e-12j)
17679 (17679-9.45021838561e-13j)
(20283.7155019-1.39666056498e-12j)
21418 (21418-4.84501327946e-12j)
(22555.5802309-3.10151904159e-12j)
24722 (24722-7.79750656198e-12j)
(26682.2144647-2.76264798875e-12j)
27525 (27525-4.32276436868e-12j)
(28289.3165054-3.93951538058e-12j)
29771 (29771-5.49181742916e-12j)
(31053.9259416-4.21006209679e-12j)
31413 (31413-5.98682963246e-12j)
(31650.6893263-5.95211397868e-12j)
32418 (32418-5.63153364891e-12j)
(32989.9823047-5.54271442797e-12j)
32767 (32767-2.73450418966e-12j)
(32403.4875307-6.84516533814e-12j)
32451 (32451-7.74491581979e-12j)
(32312.0829796-8.90532092512e-12j)
31478 (31478-5.98010529984e-12j)
(30503.0783406-8.23785484272e-12j)
29867 (29867-5.61969307688e-12j)
(29063.1334135-5.70784762847e-12j)
27651 (27651-3.62998520131e-12j)
(26112.7098023-5.70166136526e-12j)
24874 (24874-8.66862137627e-13j)
(23496.7983355-1.53699275529e-12j)
21594 (21594+9.85878045867e-14j)
(19592.8010061-1.74260605945e-12j)
17876 (17876+1.26051339299e-12j)
(16058.2048491+2.422475654e-12j)
13795 (13795+7.45803419022e-12j)
(11471.0808552+8.37108160567e-13j)
```



FFTを計算する区間は有限のため、下図のSIN波の例のように、特に端で歪が生じて、完全なSIN波にはならない。



そこで、歪を抑えることを目的に、下図のように、サンプル数の半分ずれた値を合成して、信号をつないでいくことにした。



計算結果そのもの(上段図)に、重み係数をかけて(下段図)、2つを足し合わせて 1本につないでいく。