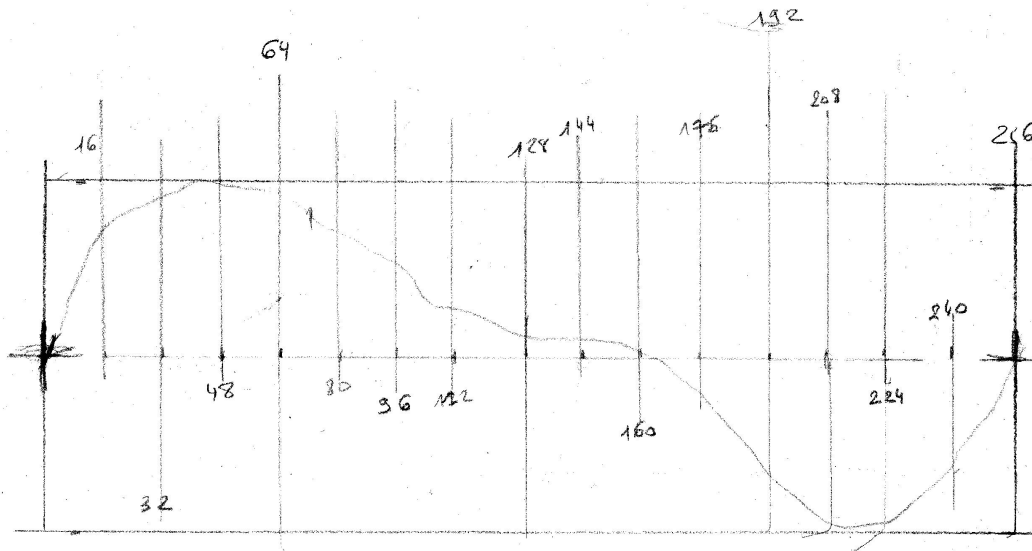


Générer du son, avec peu de moyens

1) on enregistre le son qui nous plaît, instrument seul et jouant une seule note. Par exemple j'ai enregistré un bout de « oud »

2) avec un éditeur, on sélectionne une période, on l'affiche et on la reproduit sur un calque par exemple (pour ceux qui savent comment récupérer les données dans un fichier wav, on peut certainement simplifier cette partie) ; ça donne quelque chose comme ça

3) à partir de ces données, en



mesurant, on reproduit la série de valeur dans une table en abscisse 256 points (en ordonnée, on verra apres avec quel facteur d 'échelle) ; ça donne quelque chose comme ça :

```
unsigned int onde_oud [256] =
{ 2048 , 2150 , 2252 , 2354 , 2455 , 2555 , 2654 , 2752 , 2850 ,
2948 , 3046 , 3148 , 3250 , 3340 , 3430 , 3510 , 3535 , 3560 , 3585 ,
3610 , 3635 , 3660 , 3680 , 3700 , 3720 , 3740 , 3760 , 3780 , 3800 ,
3825 , 3860 , 3890 , 3940 , 3995 , 4050 , 4085 , 4090 , 4095 , 4090 ,
4095 , 4092 , 4088 , 4084 , 4080 , 4076 , 4072 , 4068 , 4065 , 4060 ,
4054 , 4048 , 4051 , 4044 , 4037 , 4028 , 4020 , 4008 , 3998 , 3983 ,
3970 , 3955 , 3942 , 3930 , 3919 , 3900 , 3872 , 3840 , 3802 , 3776 , 3746 ,
3716 , 3686 , 3646 , 3608 , 3570 , 3541 , 3512 , 3485 , 3462 , 3444 ,
3762 , 3421 , 3410 , 3400 , 3280 , 3260 , 3238 , 3313 , 3280 , 3246 ,
3204 , 3182 , 3166 , 3150 , 3134 , 3117 , 3070 , 3022 , 2976 , 2925 ,
2900 , 2876 , 2854 , 2834 , 2812 , 2786 , 107 , 2638 , 109 , 110 , 111 ,
2572 , 113 , 114 , 115 , 2528 , 117 , 118 , 119 , 2485 , 121 , 122 ,
123 , 2398 , 125 , 126 , 127 , 2267 , 2268 , 2266 , 2264 , 2266 , 2257 ,
2264 , 2262 , 2260 , 2253 , 2247 , 2240 , 2235 , 2229 , 2220 , 2211 ,
2203 , 2200 , 2196 , 2192 , 2189 , 2187 , 2184 , 2181 , 2178 , 2175 ,
2172 , 2168 , 2164 , 2152 , 2138 , 2124 , 2101 , 2070 , 2034 , 2010 ,
1988 , 1956 , 1944 , 1908 , 1875 , 1840 , 1808 , 1778 , 1744 , 1707 ,
1670 , 1630 , 1592 , 1547 , 1500 , 1458 , 1400 , 1320 , 1244 , 1180 ,
1133 , 1074 , 1008 , 952 , 872 , 189 , 190 , 191 , 698 , 643 , 586 ,
530 , 479 , 432 , 406 , 380 , 356 , 316 , 276 , 236 , 196 , 164 , 138 ,
118 , 88 , 79 , 72 , 68 , 65 , 61 , 57 , 50 , 43 , 45 , 50 , 61 , 74 , 88 ,
104 , 118 , 130 , 146 , 160 , 176 , 206 , 236 , 268 , 306 , 348 , 402 ,
474 , 540 , 600 , 645 , 690 , 738 , 784 , 834 , 892 , 945 , 1000 , 1077 ,
1144 , 1228 , 1308 , 1388 , 1472 , 1560 , 1656 , 1746 , 1850 , 1928 ,
2000 };
```

je n'ai mis que des entiers positifs, le 0 est donc avec une valeur non nulle, on verra plus tard comment changer ça (nécessaire si on veut appliquer un coefficient multiplicateur comme enveloppe ADSR ou volume par exemple)

4) il faut récupérer les fréquences des sons que l'on veut reproduire. Par exemple (extrait de wikipedia)

C5	523,251
B4	493,883
A♯4 ou B♭4	466,164
A4	440
G♯4 ou A♭4	415,305
G4	391,995
F♯4 ou G♭4	369,994
F4	349,228
E4	329,628
D♯4 ou E♭4	311,127
D4	293,665
C♯4 ou D♭4	277,183
C4 (en)	261,626
B3	246,942
A♯3 ou B♭3	233,082
A3	220
G♯3 ou A♭3	207,652
G3	195,998
F♯3 ou G♭3	184,997
F3	174,614
E3	164,814
D♯3 ou E♭3	155,563
D3	146,832
C♯3 ou D♭3	138,591
C3	130,813

5) on regarde comment balayer la table de forme d'onde, pour obtenir la période désirée :

on connaît la fréquence d'échantillonnage, qui restera constante (dans les exemples qui suivent, j'ai utilisé 16kHz – mais pour la flute par exemple j'utiliserai sans doute 32 kHz pour sortir de la gamme audible).

Si on prend n points (n n'est entier que pour le temps du raisonnement), espacés régulièrement, on va générer une période de n/f_e , soit une fréquence de f_e/n

la fréquence à reproduire f_r est donc $f_r = f_e/n$, soit ce qui nous intéresse $n = f_e/f_r$
on va balayer la table avec des incréments tels que $n \cdot \text{inc} = 256$ (taille de la table, modifiable si souhaité)

soit encore $\text{inc} = 256/n$ ou $\text{inc} = 256 \cdot f_r / f_e$

n et inc ne doivent pas être entiers, sinon les erreurs liées à l'arrondi seront trop grandes (plus d'un ton dans certains cas), et comme il est souhaitables que les opérations soient faites sur des entiers, on génère une table avec $256 \cdot \text{incrément}$, et on divisera par 256 à la fin

la table pour $f_e = 16\text{kHz}$, avec multiplication par 256 est la suivante pour les deux octaves de C3 à C5 :

```
unsigned int increment[25] = { 536 , 568 , 601 , 637 , 675 , 715 , 758 ,  
803 , 851 , 901 , 955 , 1011 , 1072 , 1135 , 1203 , 1275 , 1350 , 1430 ,  
1515 , 1606 , 1701 , 1802 , 1909 , 2023 , 2143 } ;
```

6) le signal à sortir est donc donné par :

```
index (n)= index (n-1)+inc ;  
sample(n)= table_onde[index(n)>>8] ;
```

7) le facteur d'échelle, et le 0 à 0 ou pas (nombres signés ou non) sont à choisir en fonction du moyen de sortie du son. En général, il faudra mettre le 0 à 0 dans la table (pour les multiplications), et rajouter un décalage en sortie (pour fast PWM, ou DAC).

Le facteur d'échelle dépend de la technique de sortie

- avec un DAC 10 bits, l'échelle sera de 1024 (donc table de -512 à +511)

- avec un PWM qui nous laisse entre 700 et 800 cycles d'horloge (pour un échantillonnage à 16 kHz, soient 1000 périodes d'horloge moins le temps de gérer les interruptions, de faire les calculs etc..) le facteur devra être inférieur à cette valeur

8) manipulations de l'index :

en modifiant l'incrément, on modifie la fréquence, donc, on peut faire à peu de frais, du pitchbend, du vibrato etc..

voilà de quoi s'amuser

Jacques