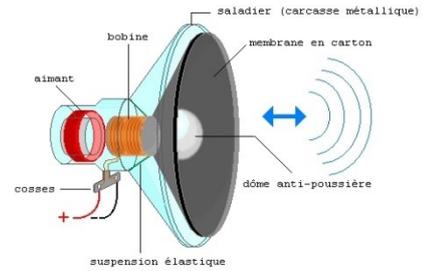
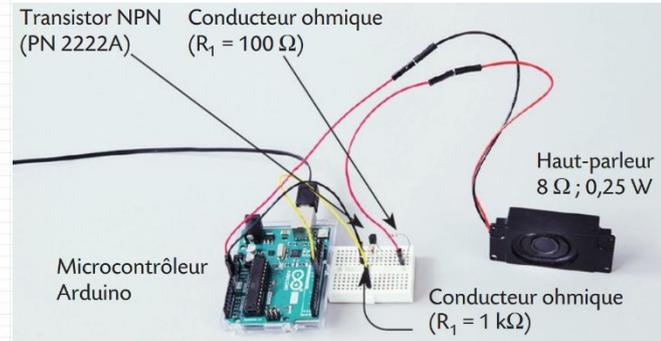
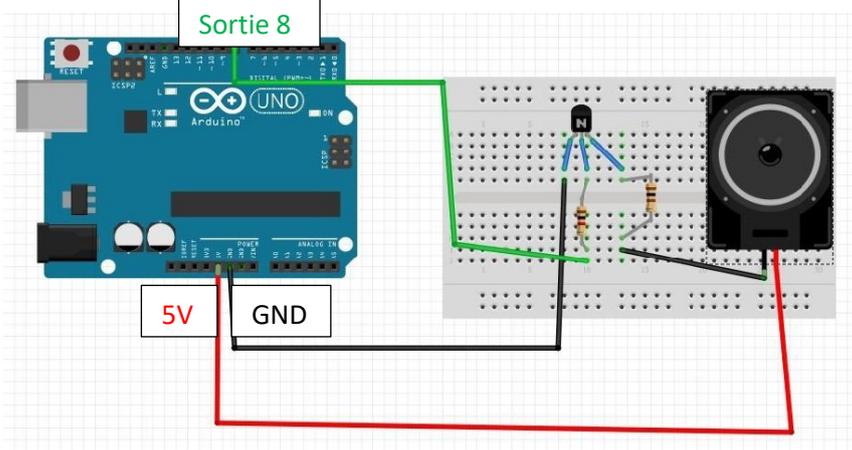


Un haut-parleur permet de transformer un signal électrique en signal acoustique. L'objectif de cette séance est d'émettre un signal électrique avec un microcontrôleur préalablement programmé afin de le transformer en signal acoustique qui peut être une mélodie. Nous pourrions aussi vérifier la fréquence des notes jouées.



### I- Réalisation du montage

→ Réaliser le montage ci-dessous, puis appeler le professeur pour vérification. [Réa]



### II- Jouer une note

→ Ouvrir le fichier « sketch\_1\_note ». Relier le Microcontrôleur Arduino à l'ordinateur et téléverser le programme. Écouter. [Réa]

```

sketch_1_note | Arduino 1.8.8
Fichier Édition Croquis Outils Aide

sketch_1_note pitches.h

#include "pitches.h"
int buzzerPin = 8;

void setup()
{
  pinMode(buzzerPin, OUTPUT);
  tone(buzzerPin, NOTE_A4, 1000);
}

void loop()
{

```

Téléverser le programme (attention de bien avoir sélectionné le bon type de carte et le port de celle-ci dans « outils »)

Bibliothèque de notes avec leur fréquence en Hz

Ici, l'instruction « tone » génère un signal de la fréquence de la « NOTE\_A4 » pendant 1000 ms = 1s sur la sortie 8.

```

sketch_1_note pitches.h

#define NOTE_G4 392
#define NOTE_G#4 415
#define NOTE_A4 440
#define NOTE_A#4 466
#define NOTE_B4 494

```

- Modifier le programme pour jouer une note de fréquence 494 Hz pendant 2s.
- Téléverser le programme dans le microcontrôleur, écouter.
- Indiquez si votre écoute confirme la modification.

[Ana]  
[Réa]  
[App]

### III- Jouer une mélodie et l'enregistrer

→ Ouvrir le fichier « sketch\_Melodie », le téléverser dans le microcontrôleur et écouter.

[Réa]

#### Explications sur le programme :

```
sketch_Melodie pitches.h

#include "pitches.h"
// notes de la mélodie:
int melody[] = {
  NOTE_A4, NOTE_A4, NOTE_A4, NOTE_F4, NOTE_C5, NOTE_A4, NOTE_F4, NOTE_C5, NOTE_A4
};
// durée d'une note: 4 = quart de note, 8 = huitième de note, ect.:
int noteDurations[] = {
  4, 4, 4, 6, 8, 4, 6, 8, 4
};

void setup() {
  //Jouer les notes de la mélodie:
  for (int thisNote = 0; thisNote < 9; thisNote++) {

    //pour calculer la durée de la note, prenez une seconde (1000 ms) divisée par le type de note.
    //e.g. quart de note = 1000/4, huitième de note = 1000/8, etc.
    int noteDuration = 1000 / noteDurations[thisNote];
    tone(8, melody[thisNote], noteDuration);

    //pour distinguer les notes, définissez un délai minimal entre elles.
    //Durée d'une note + 30% semble bien fonctionner:
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    // arrêter le son:
    noTone(8);
  }
}
```

→ Faire rejouer la mélodie (on peut téléverser à nouveau le programme) et l'enregistrer avec Audacity. [Réa]

<https://audacity.fr/>



→ Déterminer les fréquences de la 3<sup>ème</sup> et de la 4<sup>ème</sup> note de la mélodie enregistrée en expliquant la démarche.

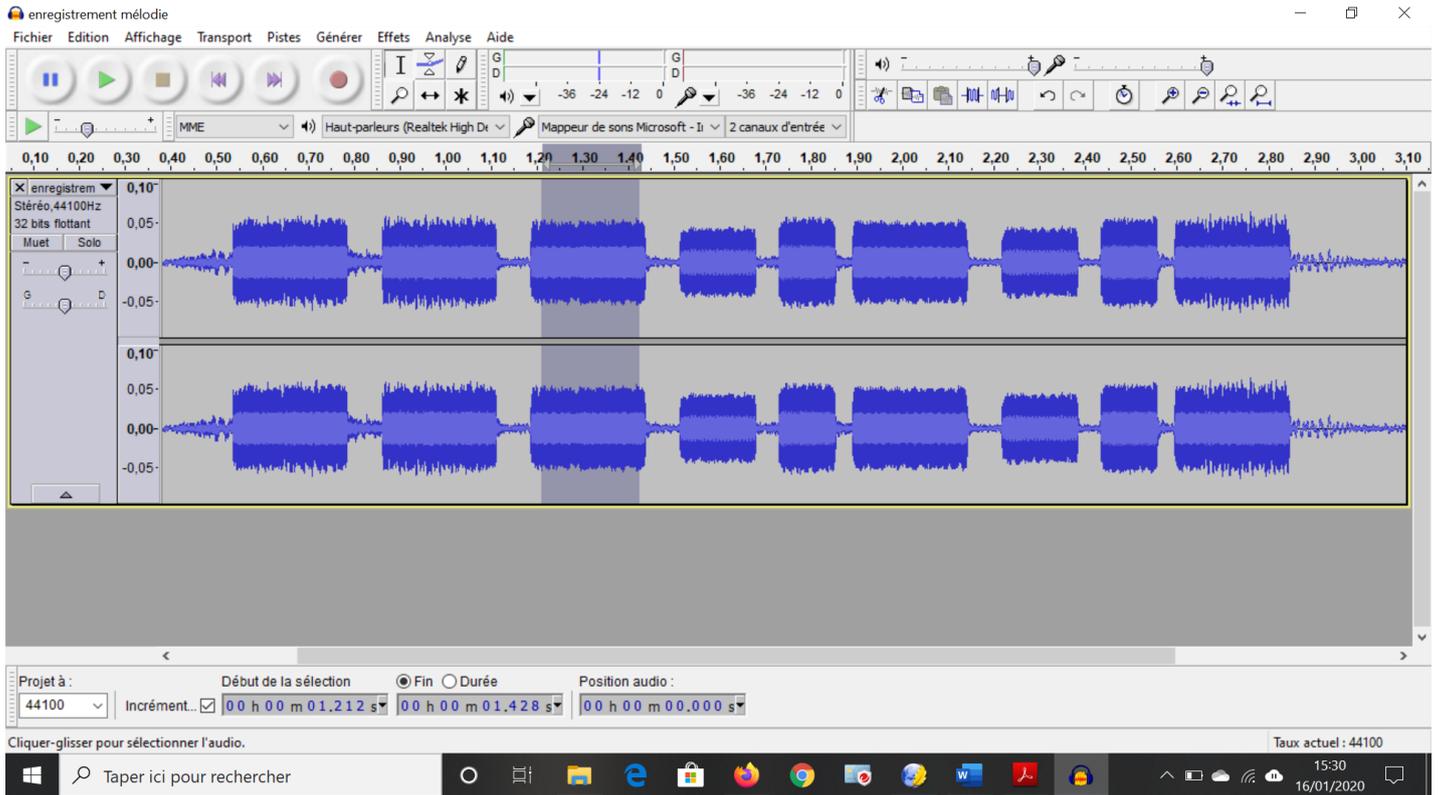
[Ana-Réa]

→ Les notes jouées correspondent-elles aux notes indiquées dans le programme ?

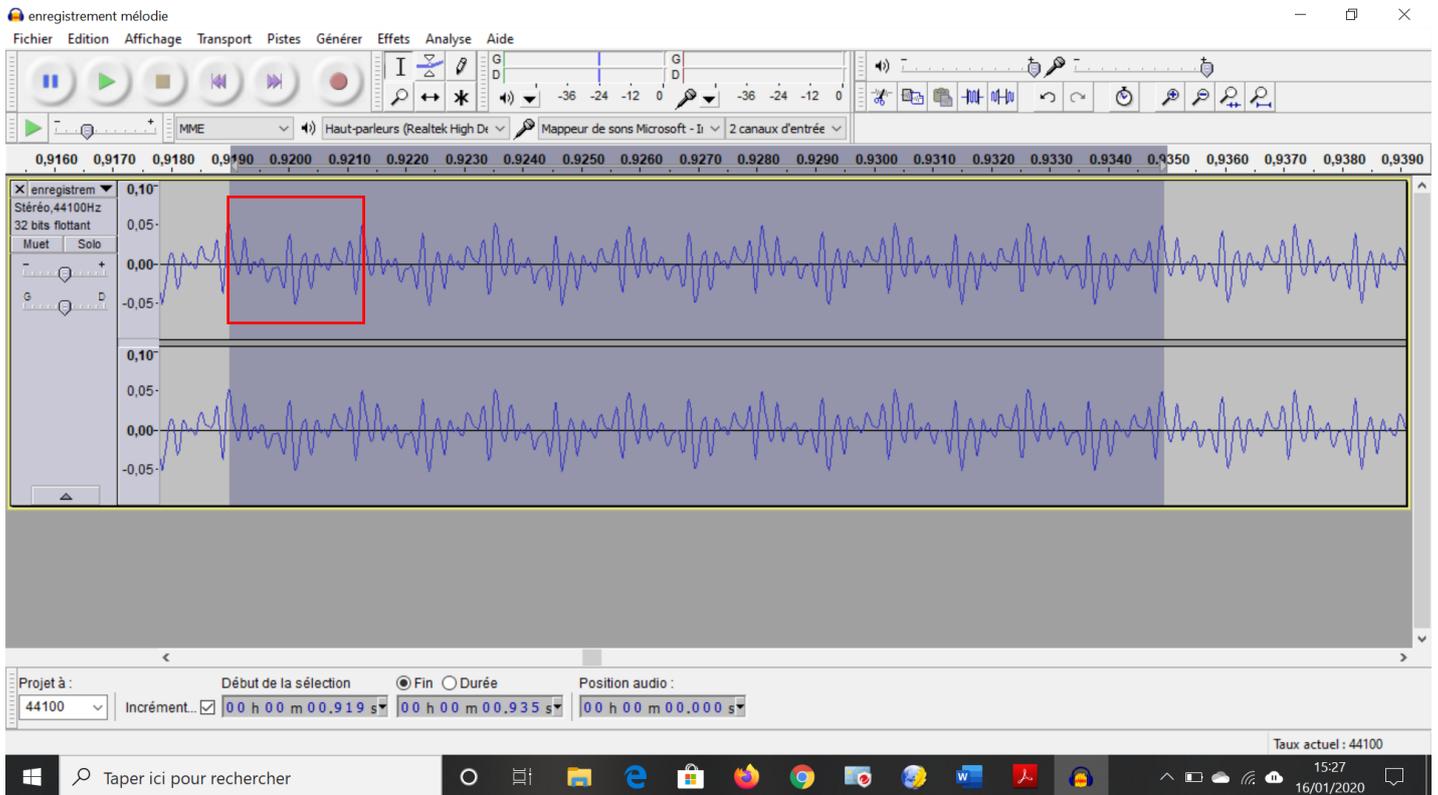
[App-Val]

## Éléments de corrections pour la partie III

Copie d'écran de l'enregistrement :

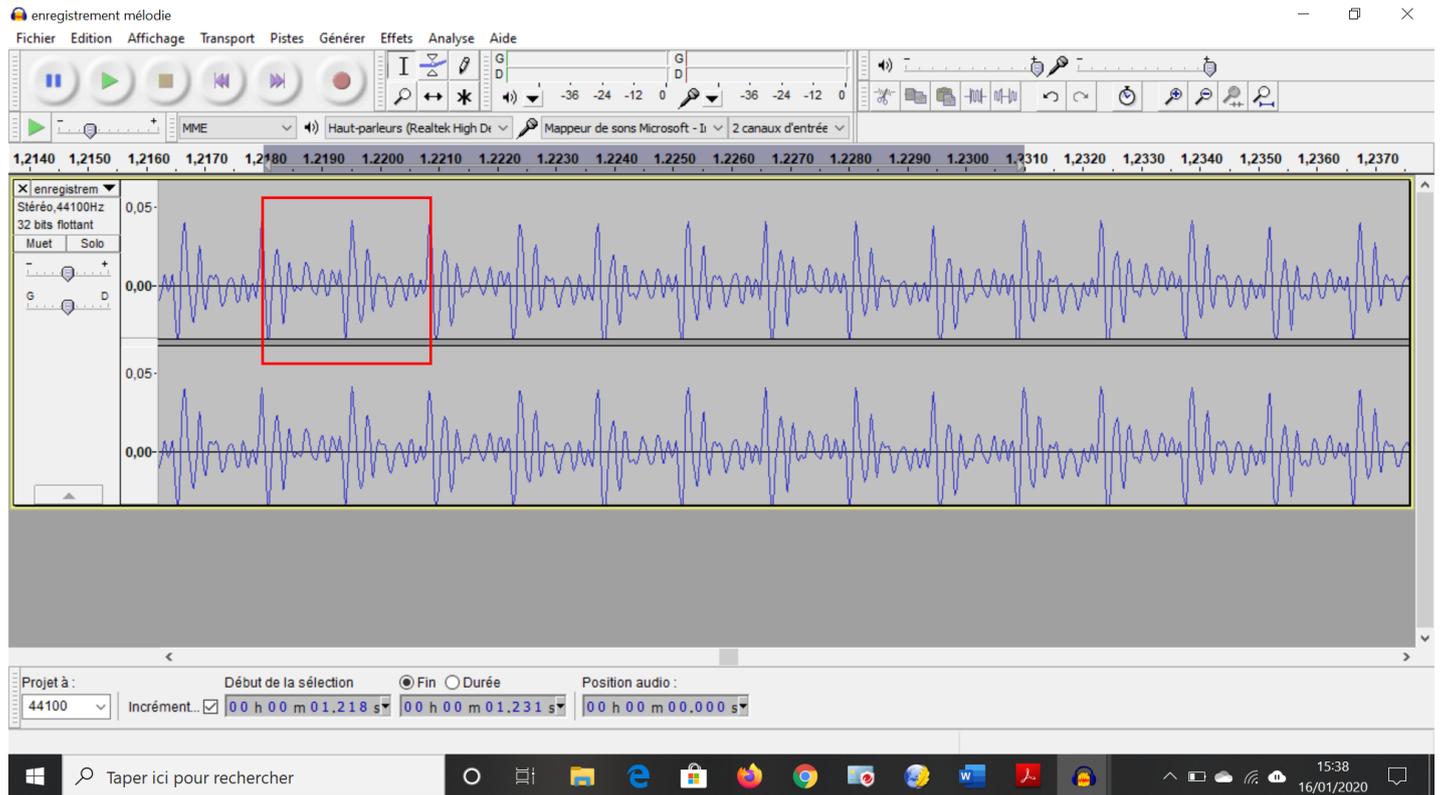


Copie d'écran du zoom sur la 3<sup>ème</sup> note :



Pour la 3<sup>ème</sup> note :  $T=(0,935-0,919)/7=2,29s$   $f=1/T=437$  HZ ce qui est proche des 440Hz de la « NOTE\_A4 »

Copie d'écran du zoom sur la 4<sup>ème</sup> note :



**Attention au motif élémentaire !**

Pour la 4<sup>ème</sup> note :  $T = (1,231 - 1,218) / 4,5 = 2,89 \text{ ms}$  soit  $f = 346 \text{ Hz}$  ce qui est proche des 349 Hz de la « NOTE\_F4 »

