



Objectif : Utiliser un dispositif comportant un microcontrôleur pour produire un signal sonore.

Un microcontrôleur permet de générer des signaux électriques que l'on peut transformer en signaux sonores à l'aide d'un haut-parleur. Mais les sons générés par un microcontrôleur ressemblent-ils à ceux joués par un instrument ? Serez-vous capable de jouer les premières notes du jeu Mario avec le microcontrôleur Arduino ?

Document 1 : Le haut-parleur

Un haut-parleur doit convertir un signal électrique en signal mécanique (changements de pression : succession de compressions et dilatations de l'air). La membrane du haut-parleur vibre à la même fréquence que celle du signal électrique. La vibration crée un signal sonore ayant lui aussi cette fréquence.



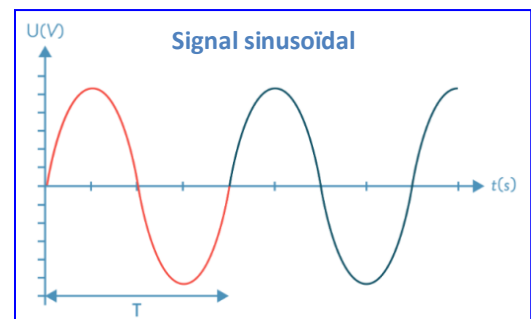
Document 2 : Signaux périodiques

Un phénomène périodique est un phénomène qui se reproduit identique à lui-même à intervalles de temps égaux.

La période T d'un phénomène périodique est la plus petite durée au bout de laquelle le phénomène se reproduit identique à lui-même. Elle s'exprime en secondes (s).

La fréquence f d'un phénomène périodique correspond au nombre de périodes par unité de temps, c'est-à-dire le nombre de fois où le phénomène se reproduit par seconde.

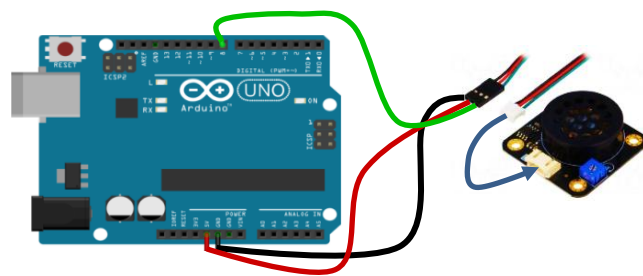
La fréquence est l'inverse de la période
La fréquence s'exprime en hertz (Hz)



$$f = \frac{1}{T} \text{ avec } \begin{cases} f: \text{fréquence du phénomène (Hz)} \\ T: \text{période du phénomène (s)} \end{cases}$$

Document 3 : Schéma du montage

- Fil vert sur borne de sortie 8
- Fil rouge sur la borne 5V
- Fil noir sur une borne GND (masse)



Document 4 : Programme Arduino

Ces lignes de code permettent de générer un signal Sonore de 800 Hz.

- Loop : boucle de répétition
- L'état bas correspond à une tension nulle
- L'état haut correspond à une tension maximale
- Le code delayMicroseconds permet de donner la durée de chaque état.

```

note
//Générer un son de 800Hz

void setup() {
  // on prépare la borne 8 en mode sortie
  pinMode(8, OUTPUT);
}

void loop() {
  //boucle qui se répète
  digitalWrite(8,0) ;// état bas (tension nulle)
  delayMicroseconds(625); // état bas pendant 625 microseconds
  digitalWrite(8,1) ;// état haut (tension maximale)
  delayMicroseconds(625); // état haut pendant 625 microseconds
    
```

Document 5 : Fréquences et notes

Note	Sol3	La3	Si3	Do4	Ré4	Mi4	Fa4	Sol4
f(Hz)	392	440	494	523	587	660	698	784

Document 6 : Arduino et la fonction tone()

La fonction tone() permet de générer un signal carré, périodique, avec comme paramètres le port, la fréquence et la durée en milliseconde.

Entre deux sons successifs différents, on peut utiliser delay() avec en paramètre la durée avant l'émission d'un nouveau son. Le delay doit être supérieur au temps du premier son.

```
char hp = 8 ; /* on précise que le haut-parleur est connecté
à la borne 8 de l'arduino (borne de sortie)*/



void setup() {

  pinMode(8, OUTPUT); /* on initialise la borne choisie comme borne
de sortie*/
  tone (hp, 800, 1000) ; // son de 800Hz pendant 1000 ms
  delay(2000); // pause
  tone (hp, 800, 2000); //son de 800Hz pendant 2000 ms
}

void loop() {
}
```

- **Copier les dossiers** qui se trouvent dans le serveur « physique-chimie »- « seconde 2019 » - « AE arduino son » et les placer dans votre dossier personnel « Mes documents ».
- Effectuer le montage. **Attention : surtout ne pas brancher le port usb !**

APPEL n°2

	Appeler le professeur pour vérifier votre montage ou en cas de difficulté.	
---	--	---

- Après vérification, relier votre microcontrôleur Arduino à l'ordinateur.
- Ouvrir le logiciel « Arduino » situé dans le dossier « Logiciels informatiques » et suivre la fiche du logiciel.
- A partir du logiciel Arduino, ouvrir le fichier « **note1** »
- Téléverser le code vers le microcontrôleur et écouter le son. Pour arrêter le son, il faut débrancher le câble de l'ordinateur. **Attention : si un message d'erreur sur le bandeau inférieur apparaît, débrancher puis rebrancher le câble usb et appuyer à nouveau sur « téléverser » (voir fiche élèves « Utilisation microcontrôleur » - icône 2 !**

Etude de ce premier son et de ses lignes de code.

1. En étudiant les documents 2 et 4, expliquez comment ce code peut générer un son de fréquence égale à 800 Hz.
2. Ce son est-il sinusoïdal ?
3. On a enregistré le son émis à l'aide du logiciel « Audacity », visualisez le signal sonore, **nommé « note Arduino »** à l'aide du logiciel Audacity présent dans votre dossier personnel « Mes documents ». Vous avez une fiche avec l'explication du logiciel « Audacity ». Vérifiez que la fréquence est bien de l'ordre de 800 Hz.
4. Modifier le programme du microcontrôleur et générer un son de 400 Hz. Avant de téléverser rebrancher le câble à l'ordinateur. **On peut modifier le code delayMicroseconds présent dans la parenthèse.** Ne pas enregistrer les modifications sur Arduino.
5. Quel est l'inconvénient de ce programme ?



Deuxième étude

- A partir du logiciel Arduino, ouvrir le fichier « **son2** ». Rebrancher le câble.
 - Téléverser le code vers le microcontrôleur et écouter le son.
1. Modifier le codage pour émettre un La3 pendant 2 s puis un Sol4 pendant 1 s après un delay de 3000.
 2. Puis téléverser votre son ainsi générer **et écouter le son.**
 3. Maintenant coder et faire jouer la musique de Mario = mi4 - mi4 - mi4 - do4 - mi4 - sol4 - sol4

On peut copier des blocs de lignes : delay (...) le temps indiqué entre parenthèse doit être supérieur à celui d'émission des sons tone (hp, ...) (en utilisant les raccourcis CTRL + C -pour copier puis CTRL+V-pour coller)

Petits conseils : la durée est la même pour toutes les notes, mais pour trouver le bon rythme il faudra varier le delay entre chaque note. <https://www.youtube.com/watch?v=205a9RTGE7M>

APPEL n°3

	Appeler le professeur pour vérifier votre codage et la musique de Mario ou en cas de difficulté.	
---	--	---