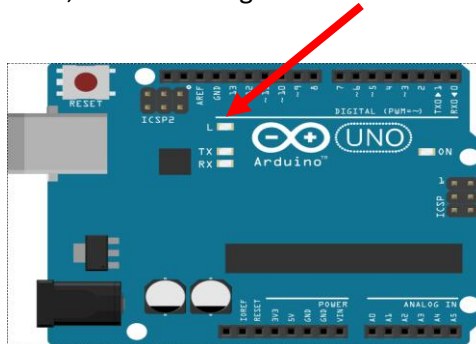





1. Prise en main

1.1 Tester la carte.

- Pour vérifier que la carte fonctionne, on va faire clignoter la LED de test sur la carte Arduino.

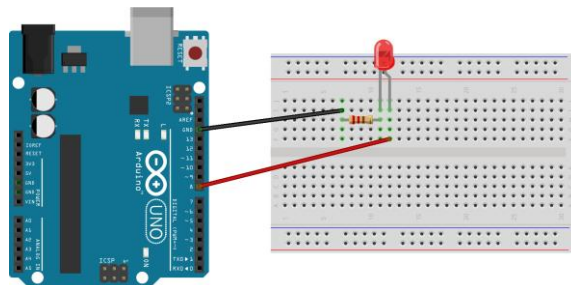


- Ouvrir le logiciel 
- Relier la carte à l'ordinateur en utilisant le câble USB
- Dans le menu Outils, vérifier :
 - le type de carte (qui correspond à la carte que vous utilisez : Arduino/Genuino Uno)
 - le port (qui apparaît en bas de la fenêtre)
- Dans le menu Fichier, choisir Exemples, Basics et le programme « Blink ».
- Téléverser le programme sur la carte Arduino.



1.2 Allumer une LED.

- Réaliser le montage ci-contre.
- Relier la carte à l'ordinateur.
- Dans le menu Outils, vérifier :
 - le type de carte
 - le port
- Recopier le programme ci-dessous.



```
int LED = 8;           // la LED est branchée sur la broche 8

void setup()
{
  pinMode(LED, OUTPUT); // la broche 8 est une sortie
}

void loop()
{
  digitalWrite(LED, HIGH); // la LED est allumée
  delay(1000);             // délai d'attente de 1 s
  digitalWrite(LED, LOW);  // la LED est éteinte
  delay(1000);            // délai d'attente de 1 s
}
```

- Téléverser le programme sur la carte Arduino.

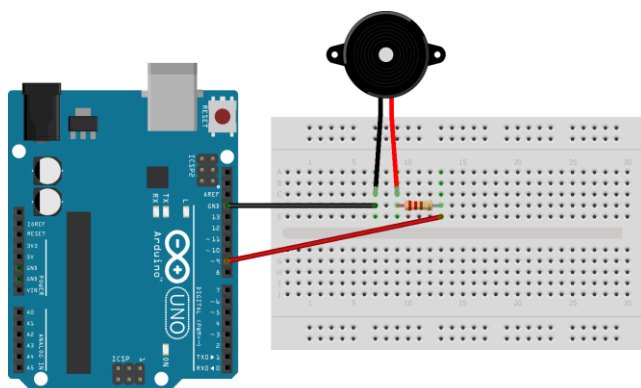
2. Activité en 2^{nde} – Thème « Ondes et signaux »

1. Émission et perception d'un son	
Notions et contenus	Capacités exigibles <i>Activités expérimentales support de la formation</i>
Signal sonore périodique, fréquence et période. Relation entre période et fréquence.	<p>Mesurer la période d'un signal sonore périodique.</p> <p>Utiliser un dispositif comportant un microcontrôleur pour produire un signal sonore.</p> <p>Capacités mathématiques : identifier une fonction périodique et déterminer sa période.</p>

Comment produire un signal sonore avec la carte Arduino ?

Document 1 : Dispositif expérimental.

On utilise un buzzer branché en série avec un conducteur ohmique de résistance $R = 1 \text{ k}\Omega$ pour obtenir une intensité sonore moyenne et afin d'avoir une intensité du courant inférieure à 40 mA.



Montage expérimental

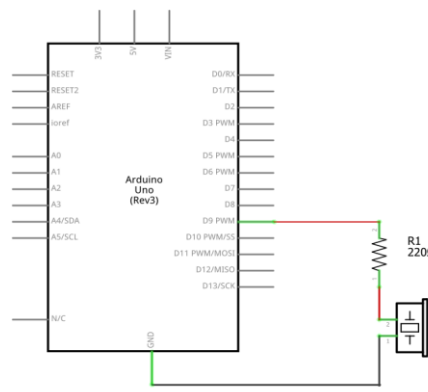


Schéma du montage

Document 2 : Programme pour émettre un La₃.

```

AE13_La3 | Arduino 1.8.7
Fichier Édition Croquis Outils Aide
AE13_La3
/*
 * Emettre un son avec une carte Arduino
 */

int PIN_BUZZER = 9;

void setup()
{
  pinMode(PIN_BUZZER, OUTPUT);
  tone(PIN_BUZZER, 440);
  delay(2000);
}

void loop()
{
  noTone(PIN_BUZZER);
}

```

Document 3 : La fonction « tone ».

tone()

[Advanced I/O]

Description

Generates a square wave of the specified frequency (and 50% duty cycle) on a pin. A duration can be specified, otherwise the wave continues until a call to noTone(). The pin can be connected to a piezo buzzer or other speaker to play tones.

Only one tone can be generated at a time. If a tone is already playing on a different pin, the call to tone() will have no effect. If the tone is playing on the same pin, the call will set its frequency.

Use of the tone() function will interfere with PWM output on pins 3 and 11 (on boards other than the Mega).

It is not possible to generate tones lower than 31Hz. For technical details, see Brett Hagman's notes.

Syntax

tone(pin, frequency)

Document 4 : Notes de musique et octaves.

Les touches blanches d'un clavier sont associées à 7 notes qui se répètent dans des octaves différentes. Lorsqu'on passe d'une octave à celle supérieure, la fréquence de la note double.



TRAVAIL A REALISER

1. Réaliser le montage expérimental du **document 1**.

APPEL N°1
Appeler le professeur pour vérification ou en cas de difficulté

2. Ouvrir le logiciel  et écrire le programme du **document 2**.

3. Vérifier le programme



et dans le menu Outils, vérifier :

- le type de carte
- le port

4. A l'aide de la documentation sur Arduino (onglet Aide puis Référence) et du tutoriel fourni, indiquer la fonction de ce programme.

5. Vérifier le en téléversant le programme sur la carte Arduino.

6. A l'aide de l'oscilloscope, déterminer la période et la fréquence du signal généré par la carte.

7. Vos mesures sont-elles en accord avec les données du programme ? Justifier.

8. En utilisant la touche FFT sur l'oscilloscope, visualiser le spectre du signal représentant l'amplitude en fonction des fréquences.

9. Que constatez-vous pour les fréquences et les amplitudes ?

10. Modifier le programme de manière à alterner « indéfiniment » l'émission d'un Do₃ de fréquence $f = 262$ Hz durant 2 secondes avec une plage de silence de 2 secondes.

11. Modifier le programme de manière à alterner « indéfiniment » l'émission d'un La₄ durant 2 secondes avec une plage de silence de 5 secondes.

12. Associer chacune de ces fréquences à un son aigu ou à un son grave.

<u>Compétences auto-évaluées :</u>		A	B	C	D
S'approprier	Rechercher et organiser l'information en lien avec la problématique				
Réaliser	- Mettre en œuvre un protocole expérimental en respectant les règles de sécurité - Tester et modifier un programme informatique				
Valider	Exploiter et interpréter des observations, des mesures				

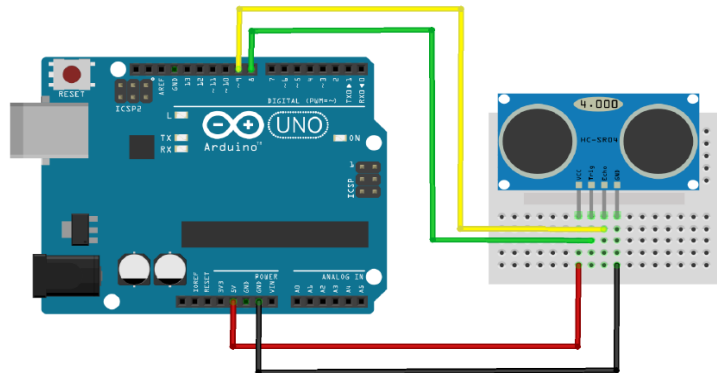
3. Activité en 1^{ère} Spécialité – Thème « Ondes et signaux »

1. Ondes mécaniques	
Notions et contenus	Capacités exigibles <i>Activités expérimentales support de la formation</i>
	<i>Déterminer, par exemple à l'aide d'un microcontrôleur ou d'un smartphone, une distance ou la célérité d'une onde. Illustrer l'influence du milieu sur la célérité d'une onde.</i>

Quelle est la distance de la table au plafond ?

3.1 Réaliser le montage ci-dessous.

3.2 Ecrire le programme ci-dessous en le complétant.




```
int trigg = 8; // la patte trigg est branchée sur la broche 8
int echo = 9; // la patte écho est branchée sur la broche 9
float temps; // on définit la variable temps comme un réel
float distance; // on déclare la variable distance comme un réel
int compt = 0; // on déclare un entier compt

void setup()
{
  pinMode(trigg, OUTPUT); // la broche 8 est une sortie
  digitalWrite(trigg, LOW); // la broche 8 ne délivre pas de tension
  pinMode(echo, INPUT); // la broche 9 est une entrée
  Serial.begin(9600); // la liaison série est activée
}

void loop()
{
  digitalWrite(trigg, HIGH); // on active la broche 8 pour émettre un signal
  delayMicroseconds(10); // la carte attend 10 microsecondes
  digitalWrite(trigg, LOW); // la broche 8 ne délivre plus de tension
  temps= pulseIn(echo, HIGH); // mesure du temps entre l'émission et la réception du signal ultrason
  distance = .....
  compt += 1;
  if (compt>100)
  {
    exit(0);
  }
  Serial.println(distance); //on affiche la distance dans le moniteur série (avec retour à la ligne)
  delay(500); // toutes les demi-secondes
}
```

3.3 Téléverser le programme sur la carte Arduino.

3.4. Ouvrir le moniteur série  afin d'afficher les valeurs de la distance. Copier/Coller les mesures dans un fichier.txt et l'ouvrir avec Regressi. Tracer l'histogramme.