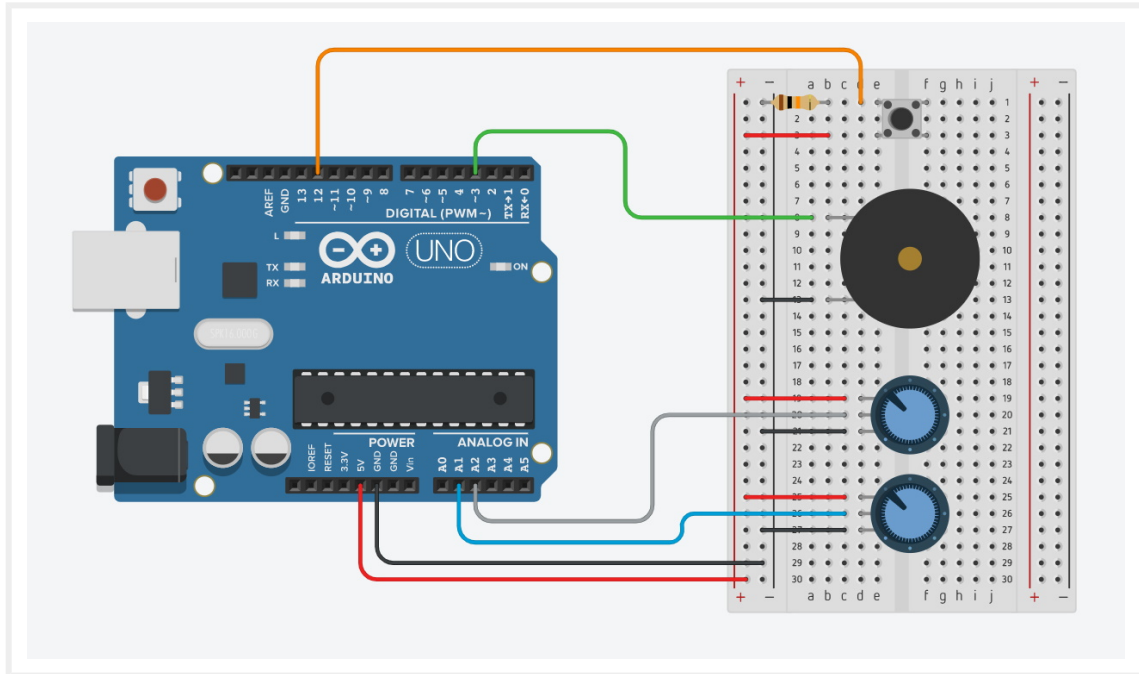


Ondes sonores – Réglage fréquence

(Régler la fréquence d'une onde sonore avec deux potentiomètres)



. Liste des composants :

- . 1 bouton poussoir
- . 1 résistance de 10 k Ω (résistance du circuit du bouton poussoir)
- . 1 haut-parleur (ou buzzer)
- . 2 potentiomètres de 10 k Ω
- . 1 plaque d'essai
- . Fils de connexion

. Objectif

Dans cette activité, l'appui sur le premier bouton-poussoir produit une onde sonore dont la fréquence est réglée à l'aide de 2 potentiomètres :

- le premier potentiomètre permet un réglage rapide de la fréquence entre 0 et 4080 Hz,
- le deuxième potentiomètre effectue un réglage fin de la fréquence sur une plage de 255 Hz,
- l'émission sonore est arrêtée en appuyant de nouveau sur le bouton poussoir.

Le potentiomètre de réglage rapide est connecté sur la broche **A1** de l'Arduino.

La tension de cette broche varie donc entre **0** et **5 V** en fonction de la position du curseur du potentiomètre.

La lecture de la valeur de la broche **A1** convertie par le convertisseur analogique numérique de l'Arduino donne donc un nombre entier entre **0** et **1023**.

Ce nombre est divisé par 4 de façon à obtenir un nombre entier compris entre **0** et **255** qui sera convertie en nombre binaire (sur 8 bits) :

0 en décimal = **00000000** en binaire

255 en décimal = **11111111** en binaire

Ce nombre binaire sur 8 bits est convertie en nombre binaire sur 12 bits en ajoutant 4 bits de poids faibles, **0000**, à sa fin. On obtient donc un nombre binaire (sur 12 bits) compris entre **000000000000** et **111111110000**, soit en décimal, un nombre entier entre **0** et **4080**.

Le potentiomètre de réglage fin est connecté sur la broche **A2** de l'Arduino. Selon le même principe que précédemment, la lecture de la broche **A2** donne une valeur comprise entre **0** et **1023**.

Ce nombre est également divisé par 4 et convertie en nombre binaire sur 12 bits. On obtient donc un nombre binaire compris entre **000000000000** et **000011111111** (entre 0 et 255 en décimal).

La conversion en décimal de l'addition des deux nombres binaires (issus de A1 et A2) nous donnent la valeur de la fréquence en Hz de l'onde sonore, soit entre **0** et **4335** Hz avec un pas de réglage de 1 Hz.

. Rappels

En informatique, outre la base 10, on utilise très fréquemment le système binaire (base 2) puisque la logique booléenne est à la base de l'électronique numérique. Deux symboles suffisent : 0 et 1. Cette unité élémentaire ne pouvant prendre que les valeurs 0 et 1 s'appelle un bit (de l'anglais binary digit). Une suite de huit bits s'appelle un octet.

Le tableau ci-dessous montre la représentation des nombres de 0 à 15 dans les bases 10 et 2 :

Décimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Binaire	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

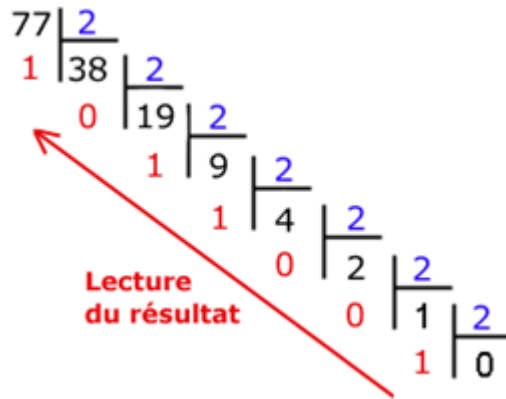
. Conversion du décimal en binaire :

La méthode de conversion la plus simple est celle de la division euclidienne par 2. Elle est facile à utiliser en programmation (il est facile d'en faire un algorithme). Voilà comment on fait :

. On a notre nombre en décimal, par exemple : **77**

- . On le divise par 2 et on note le reste de la division (c'est soit un **1** soit un **0**).
- . On refait la même chose avec le quotient précédent, et on met de nouveau le reste de côté.
- . On réitère la division, et ce jusqu'à ce que le quotient soit égale à 0.

Le nombre en binaire apparaît alors. le premier bit à placer est le dernier reste non nul. Ensuite, on remonte en plaçant les restes que l'on avait. On les place à droite du premier 1 :



77 s'écrit donc en base 2 : **1001101**

. Conversion du binaire en décimal :

Le nombre binaire **1001101** est composé de 7 bits et chaque bit correspond à une puissance de 2. Le premier (en partant de la droite) est le bit de la puissance 0, le deuxième celui de la puissance 1, le troisième celui de la puissance 3, etc...

Pour le convertir un nombre binaire en décimal, on procède de la manière suivante :

On multiplie par 2^0 la valeur du premier bit, par 2^1 la valeur du deuxième bit, par 2^2 la valeur du troisième bit, [...], par 2^{10} la valeur du onzième bit, etc... et on fait la somme des résultats :

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	0	1	1	0	1

Le nombre binaire **1001101** en base 10 est :

$$2^6 + 2^3 + 2^2 + 2^0 = 64 + 8 + 4 + 1 = 77$$

. Le programme

Ondes_sonores_Reglage_frequence

```
// Déclaration des constantes et variables

const int PinButton = 12;
const int PinTone = 3;
const int PinPot[] = {1,2};

int ValButton=0;
int OldValButton=0;
int State = 0;
int ValPot[] = {0,0};
int FreqWave=0;

// Définition de fonctions

String dec2bin(int d){
String b;
    if (d == 0){
        b = "0";
    }
    else{
        b = "";
        while (d != 0){
            b = "01"[d & 1] + b;
            d = d >> 1;
        }
        return b;
    }
}

int bin2dec(String b){
int i, len;
int result=0;
len = b.length();
for(i=0; i<len; i++)
{
    result = result*2+int(b[i]- '0');
}
return result;
}

int CalculFreq(int valpot1, int valpot2){
String valpot1bin, valpot2bin;
int Freq;
    valpot1bin = dec2bin(int(valpot1/4));
    valpot1bin= valpot1bin+"0000";
    valpot2bin = dec2bin(int(valpot2/4));
    valpot2bin = "0000"+dec2bin(int(valpot2/4));
    Freq = bin2dec(valpot1bin) + bin2dec(valpot2bin);
    return Freq ;
}
```

```
// Initialisation des entrées et sorties

void setup() {
  pinMode (PinButton, INPUT);
}

// Fonction principale en boucle

void loop() {
  ValButton = digitalRead(PinButton);
  delay(10);
  if ((ValButton == HIGH) and (OldValButton==LOW)) {
    State = 1 - State;
  }
  OldValButton = ValButton;
  if (State==1){
    ValPot[0]= analogRead(PinPot[0]);
    ValPot[1]= analogRead(PinPot[1]);
    FreqWave = CalculFreq(int(ValPot[0]), int(ValPot[1]));
    tone(PinTone, FreqWave);
  }
  else{
    noTone(PinTone);
  }
}
```

Déroulement du programme :

