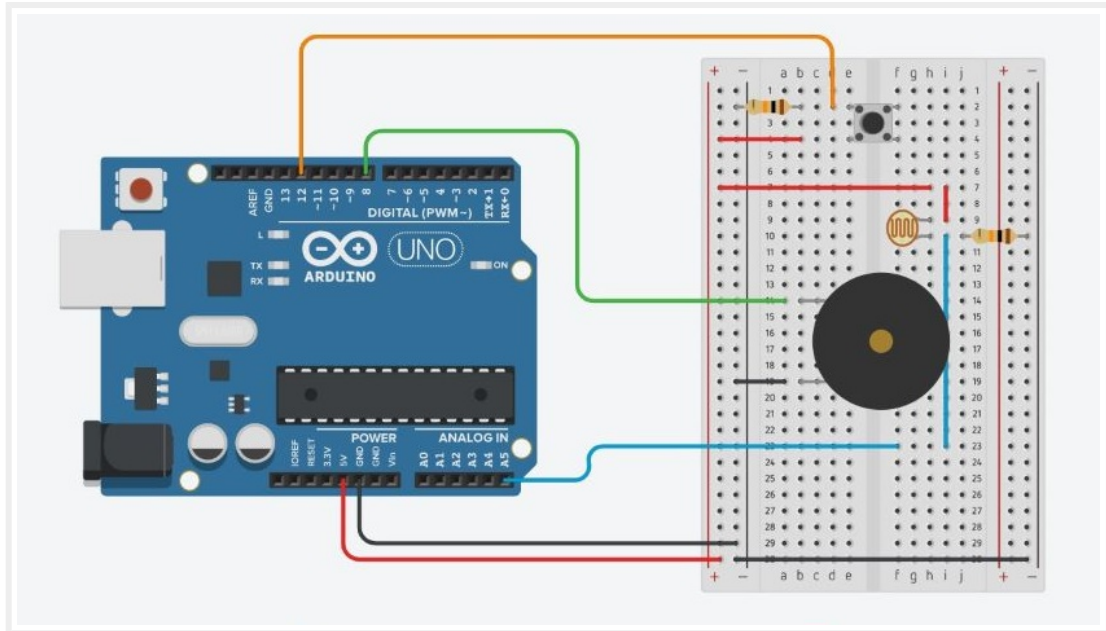


Theremin

(Régler la fréquence d'une onde sonore avec une photorésistance)



. Liste des composants :

- . 1 photorésistance
- . 1 résistance de 10 k Ω (résistance du circuit de la photorésistance)
- . 1 bouton poussoir
- . 1 résistance de 10 k Ω (résistance du circuit du bouton poussoir)
- . 1 buzzer
- . 1 plaque d'essai
- . Fils de connexion

. Objectif

L'objectif de cette activité est de réaliser un pseudo-thérémine.

Le thérémine est un des plus anciens instruments de musique électronique, inventé en 1920 par le Russe Lev Sergueïevitch Termen (connu sous le nom de « Léon Theremin »).

Composé d'un boîtier électronique équipé de deux antennes, l'instrument a la particularité de produire de la musique sans être touché par l'instrumentiste. Dans sa version la plus répandue, la main droite commande la hauteur de la note (la fréquence du son), en faisant varier sa distance à l'antenne verticale. L'antenne horizontale, en forme de boucle, est utilisée pour faire varier le volume selon sa distance à la main gauche.



Dans cette activité, on ne simulera que la commande de la hauteur de la note, à l'aide de la photorésistance du montage d'étude.

La sortie de la photorésistance est connectée à l'entrée analogique **A5** de l'Arduino. La valeur de cette broche (nombre entier entre 0 et 1023), lue par la fonction “**analogRead()**“, est alors proportionnelle à l'intensité lumineuse reçue par la photorésistance.

En approchant la main de la photorésistance, la tension mesurée au niveau de la broche A5 diminue et inversement en reculant la main la tension mesurée augmente.

Ainsi en fonction de la position de la main par rapport à la photorésistance, on va pouvoir jouer la gamme de notes de l'octave 3 sans les notes diésées (#) :

Notes	Do	Ré	Mi	Fa	Sol	La	Si
Fréquence (Hz)	262	294	330	350	392	440	494

En premier, il faut étalonner la photorésistance de façon à connaître les valeurs minimale et maximale de la broche A5 en fonction des conditions d'éclairement de la photorésistance.

Puis on attribue pour chaque note de musique une plage de valeurs de tension réparties entre les valeurs minimale et maximale :



Chaque plage de tension représente : $\frac{\text{Max}-\text{Min}}{7}$

On crée alors une liste de notes à jouer et une liste de fréquences associée aux notes :

```
String Notes[]={"Do","Re","Mi","Fa","Sol","La","Si"};
```

```
const int FreqNotes[] = {262,294,330,350,392,440,494};
```

Alors :

Le "Do" est la note "0" : **Notes[0]** (fréquence : **FreqNotes[0]**),

Le "Ré" est la note "1" : **Notes[1]** (fréquence : **FreqNotes[1]**),

...

Et la plage de tension pour chaque note "i" :

$$\text{Min} + i \times \left(\frac{\text{Max}-\text{Min}}{7}\right) < \text{Valeur A5} < \text{Min} + (i+1) \times \left(\frac{\text{Max}-\text{Min}}{7}\right)$$

Il suffira ensuite de lire la valeur de la broche A5, de tester dans quelle plage de mesure, elle se trouve et de jouer la note associée.

[. Le programme](#)

Voici le programme de l'activité:

Theremin

```
// Déclaration des constantes et variables
```

```
const int PinPhotoR = 5;
```

```
const int PinTone = 8;
```

```
const int PinButton = 12;
```

```
int State = 0;
```

```
int ValButton = 0;
```

```
int OldValButton = 0;
```

```
int ValCapteur = 0;
```

```
int MinPhotoR = 0;
```

```
int MaxPhotoR = 0;
```

```
String Notes[]={"Do","Re","Mi","Fa","Sol","La","Si"};
```

```
const int FreqNotes[] = {262,294,330,350,392,440,494};
```

```
// Définition de fonctions
```

```
void Etalonnage_PhotoR() {  
  char c=' '  
  Serial.println("Etalonnage de la photoresistance:");  
  Serial.println("");  
  while(c!='C')  
  {  
    int Val=0;  
    Serial.println("Veuillez cacher la photoresistance et saisir 'C' pour continuer.");  
    while(!Val)  
    {  
      delay(200);  
      Val=Serial.available();  
      ValCapteur = analogRead(PinPhotoR);  
    }  
    c = Serial.read();  
  }  
  MinPhotoR=ValCapteur;  
  Serial.print("Valeur minimale de la broche de la photoresistance: ");  
  Serial.println(MinPhotoR);  
  Serial.println("");  
  c=' '  
  while(c!='C')  
  {  
    int Val=0;  
    Serial.print("Veuillez exposer la photoresistance a la lumiere ");  
    Serial.println("et saisir 'C' pour continuer.");  
    while(!Val)  
    {  
      delay(200);  
      Val=Serial.available();  
      ValCapteur = analogRead(PinPhotoR);  
    }  
    c = Serial.read();  
  }  
  MaxPhotoR=ValCapteur;  
  Serial.print("Valeur maximale de la broche de la photoresistance: ");  
  Serial.println(MaxPhotoR);  
  Serial.println("");  
}
```

```
// Initialisation des entrées et sorties
```

```
void setup() {  
  Serial.begin(9600);  
  pinMode (PinButton, INPUT);  
  Etalonnage_PhotoR();  
  boolean Etalonnage=true;  
  while (Etalonnage == true)  
  {  
    char c=' '  
    while(c!='O' and c!='N')  
    {  
      Serial.println("Voulez vous recommencer l'etalonnage (Saisir 'O' ou 'N')?");  
      Serial.println("");  
      int Val=0;  
      while(!Val)  
      {  
        delay(200);  
        Val=Serial.available();  
      }  
      c = Serial.read();  
    }  
  
    if (c=='O') {  
      Etalonnage PhotoR();  
    }  
    else{  
      Etalonnage=false;  
      Serial.println("Appuyez sur le bouton poussoir pour jouer des notes de musique...");  
    }  
  }  
}
```

```
// 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) {  
    ValCapteur = analogRead(PinPhotoR);  
    for(int i = 0 ; i < 7 ; i++) {  
      if (ValCapteur> MinPhotoR+ i*(MaxPhotoR-MinPhotoR)/7 and  
          ValCapteur< MinPhotoR+ (i+1)*(MaxPhotoR-MinPhotoR)/7) {  
        Serial.println(Notes[i]);  
        tone (PinTone, FreqNotes[i]);  
        break;  
      }  
    }  
  }  
  else {  
    noTone (PinTone);  
  }  
}
```

Déroulement du programme :

– Déclaration des constantes et variables :

```
. const int PinButton = 12 (constante nombre entier correspondant au n° de la broche sur laquelle le
bouton poussoir est connecté)

. const int PinTone = 8 (constante nombre entier correspondant au n° de la broche sur laquelle le
buzzer est connecté)

. const int PinPhotoR = 5 (constante nombre entier correspondant au n° de la broche sur laquelle la
photorésistance est connectée)

. int State = 0 (variable nombre entier correspondant à l'action à effectuer)

. int ValButton = 0 (variable nombre entier pour stocker la valeur de la broche du bouton poussoir)

. int OldValButton = 0 (variable nombre entier pour stocker la valeur précédente de la broche du
bouton poussoir)

. int ValCapteur = 0 (variable nombre entier pour stocker la valeur de la broche de la photorésistance)

. int MinPhotoR = 0 (variable nombre entier pour stocker la valeur minimale de la broche de la
photorésistance)

. int MaxPhotoR = 0 (variable nombre entier pour stocker la valeur maximale de la broche de la
photorésistance)

. String Notes[]={« Do », «Re », «Mi », «Fa », «Sol », «La », «Si »}
const int FreqNotes[] = {262,294,330,350,392,440,494}
(Listes des notes et leurs fréquences)
```

– Définition de fonctions :

void Etalonnage_PhotoR() (fonction permettant l'étalonnage de la photorésistance):

- Utilisation du moniteur série pour demander à l'utilisateur de cacher la photorésistance,
- lecture de la valeur de la broche A5,
- l'utilisateur doit envoyer le caractère 'C' à l'aide du moniteur série pour continuer l'étalonnage,
- Affichage de la valeur minimale mesurée de la broche A5,
- L'utilisateur doit exposer la photorésistance à la lumière ambiante,
- lecture de la valeur de la broche A5,
- l'utilisateur doit envoyer le caractère 'C' pour quitter l'étalonnage,
- Affichage de la valeur maximale mesurée de la broche A5.

– Initialisation des entrées et sorties :

- . La liaison série est initialisée à 9600 bauds,
- . La broche du bouton poussoir est initialisée comme une entrée digitale,
- . Appel de la fonction d'étalonnage,
- . Possibilité de recommencer l'étalonnage.

– Fonction principale en boucle :

