

Thérémine avec la lumière

Principe : Utiliser la quantité de lumière pour changer la tonalité du thérémine à l'aide d'une photo résistance.

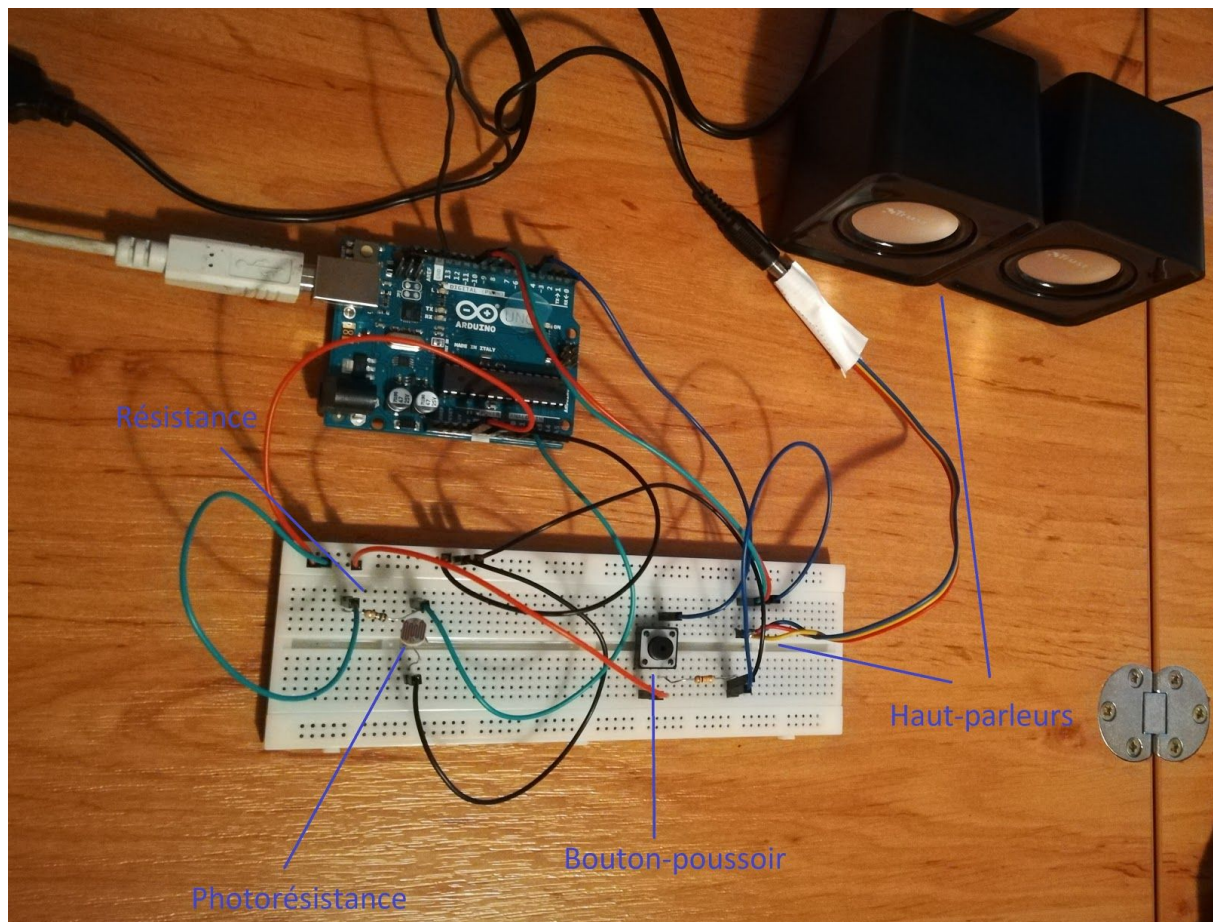
Composants :

- 1 arduino
- 1 breadboard
- 1 haut-parleur
- 1 photo-résistance
- 1 résistance
- 1 bouton-poussoir
- des fils électriques

Le bouton poussoir servira de bouton on-off.

Conception :

- 1) Réaliser le circuit suivant :



2) On téléverse le code suivant vers l'arduino

// Theremin optique //Branchement

```
#define PHONES 9 // Haut parleur connecté à la sortie 9
#define PHOTOCELL 0 //Photorésistance connecté à l'entrée A0
const int buttonPin = 4; const int ledPin = 13;
int on = 0;
int lastState = LOW; //variable definitions
long val = 0; //stockage de la valeur de Photoresistance
long lastVal = 0;
long maxread = 0; //valeur max de la phase de calibrage
long minread = 1000; //valeur min de la phase de calibrage
double f = 0; // fréquence du son
double normf = 0; // fréquence normalisé
double logfre = 0; // logarithme de la fréquence normalisé
int ilogf = 0; // logarithme arrondi int i = 0; //
double factor = 0; // échelle du facteur pour la calibration
double shift = 0; // espace de calibration
long maxfreq = 440; // Maximum de la fréquence désirée après la calibration
long minfreq = 55; // minimum de la fréquence désirée après la calibration

//Valeur pour changer l'écart entre 2 notes
//double gap = 1.148698355; //(pentatonic)
double gap = 1.059463094; //(chromatic)
void setup() {
    pinMode(PHONES, OUTPUT); // mise en sortie des pattes digitales
    pinMode(ledPin, OUTPUT);
    digitalWrite(ledPin, HIGH); // boucle de calibration pour déterminer un les niveaux
    Max et min de la photorésistance
    for (i = 0; i < 500; i++) { // calibration loop runs for 5 seconds
        val = analogRead(PHOTOCELL); // lecture de la photorésistance
        tone(PHONES, val); // joue la note brute pour guide la calibration
        if (val > maxread) { // mise en place de la valeur la plus grande
            maxread = val; }
        if (val < minread) { // mise en place de la valeur la plus petite
            minread = val; }
    }
    delay(10);
}
// on utilise la calibration pour calculer l'échelle et les changements de paramètre
factor = (double)(maxfreq - minfreq) / (double)(maxread - minread);
shift = factor * minread - minfreq;
digitalWrite(ledPin, LOW);
}
```

```

void loop() {
    int buttonState = digitalRead(buttonPin);
    if (buttonState == HIGH && lastState == LOW) {
        //à l'appui du bouton, on allume ou on éteint le son
        if (on == 1) {
            on = 0;
        }
        else {
            on = 1;
        }
    }
    delay(50);
    lastState = buttonState;
    val = analogRead(PHOTOCELL); // lecture de la photoresistance Serial.println(val);
    int averageVal = (lastVal + val)/2; //calcul de la note moyenne avec la note
    précédente
    f = factor * averageVal - shift; // linéarise la fréquence // entre une valeur Max et min
    normf = f / (double) minfreq; // divise une fonction exponentielle par la valeur min
    logfre = log(normf) / log(gap); // prends le log de base gap et le résultat
    ilogf = round(logfre); // est le nombre de notes au dessus de la plus basse, une fois
    arrondie
    f = minfreq * pow(gap,ilogf); //
    if (on == 1) {
        tone(PHONES, f); // produit la tonalité du signal si le son doit être on
    } else {
        noTone(PHONES); // on ne joue pas de son
    }
    lastVal = averageVal; // sauvegarde de la note actuelle
}

```

Utilisation :

- 1) Brancher l'arduino et l'allumer.
- 2) Un son se jouera pendant que le calibrage de la lumière se fera. Essayer de faire varier la lumière pendant cette phase de calibrage.
- 3) Allumez le son en appuyant sur le bouton poussoir
- 4) Vous pouvez éteindre le son en appuyant sur ce bouton
- 5) À vous de jouer !