

Thérémine

Contexte du sujet.

Vous êtes fan de musique électro, et vous vous demandez si pourriez fabriquer un thérémine simplifié.

Suivez ce lien et écoutez attentivement.

https://www.youtube.com/watch?v=Xh_oWPtC-Kk















Appropriation : qu'est-ce qu'un thérémine ?

- 1- Quels sont les deux capteurs en entrée du thérémine ?
Il y a deux antennes. L'antenne verticale et l'antenne en forme de boucle.
- 2- A quelle grandeur physique sont-ils sensibles ?
Ces deux antennes sont sensibles au champ électromagnétique, lui-même dépendant de la distance de la main à l'antenne.
- 3- Quel est l'actionneur commandé en sortie du thérémine ?
Le thérémine commande un haut-parleur.
- 4- Quelles grandeurs physiques sont-elles respectivement commandées en sortie ?
La distance de la main droite à l'antenne verticale commande la hauteur de la note jouée, soit sa fréquence. De même la distance de la main gauche à l'antenne en boucle commande le volume, soit l'intensité sonore.

Analyse : quels composants choisir ?

Vous décidez de modéliser un thérémine simplifié avec un microcontrôleur Arduino Uno. Parmi les dipôles ci-dessous, lesquels choisissez-vous ?

 <p>Potentiomètre Type de résistance dont la résistance change lorsqu'un bouton est actionné.</p>	 <p>Résistance Restreint le débit d'électricité dans un circuit, réduisant ainsi la tension et le courant.</p>	 <p>LED RVB Type de voyant LED qui associe le rouge le vert et le bleu pour produire une couleur.</p>	 <p>Capteur infrarouge passif Capteur de mouvement infrarouge passif permettant de détecter un...</p>
 <p>Condensateur Stocke et libère de l'énergie électrique dans un circuit.</p>	 <p>LED Diode lumineuse qui s'allume lorsque l'électricité la traverse dans la bonne direction.</p>	 <p>Diode Permet au courant électrique de circuler dans une seule direction.</p>	 <p>Piezo Type de ronfleur qui émet du bruit à différentes fréquences.</p>
 <p>Interrupteur à glissière Un commutateur à deux positions: ouvert ou fermé.</p>	 <p>Bouton poussoir Une pression sur cet interrupteur permet de fermer un circuit.</p>	 <p>Photorésistance Capteur dont la résistance change en fonction de la quantité de lumière qu'il détecte.</p>	 <p>Capteur de température [TMP36] Capteur produisant des tensions différentes en fonction de la température...</p>

1- Pour modéliser les deux capteurs ?

Pour modéliser l'antenne en boucle, un potentiomètre peut par exemple être utilisé. Il contrôlera facilement le niveau sonore.

Quant à l'antenne verticale (et au mouvement de la main), elle peut être modélisée par une photorésistance qui sera plus ou moins à l'ombre d'une main. L'intensité lumineuse tombant sur la photorésistance, et donc la tension à ses bornes commandera la fréquence.

2- Pour modéliser l'actionneur ?

Le haut-parleur est bien sûr remplacé par le piezo.

Analyse : comment accorder un thérémine véritable et votre thérémine simplifié ?

1- Rappelez le protocole pour accorder le thérémine, c'est-à-dire comment régler la gamme de fréquence à laquelle il joue.

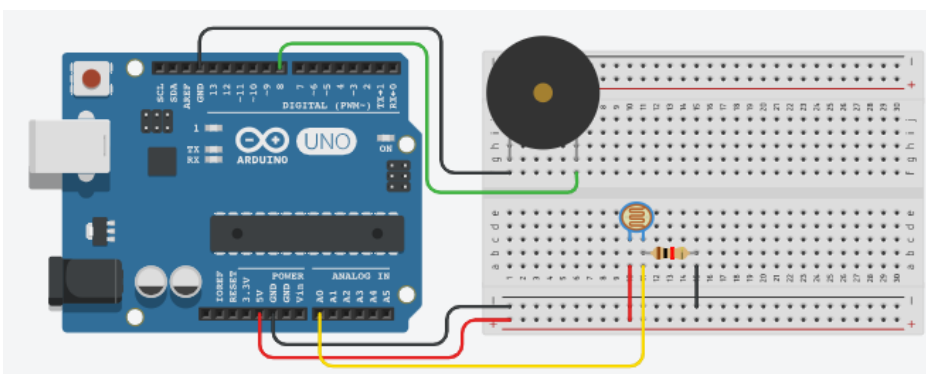
La note la plus aigüe jouée est réglée en touchant l'antenne verticale : la distance main antenne est minimale, (et la capacité maximale). La note la plus grave en plaçant la main derrière son corps : la distance main antenne est alors maximale.

Dans un premier temps, vous chercherez seulement à modéliser les variations des fréquences du thérémine par les variations de luminosité à l'aide d'une photorésistance. Cette dernière est utilisée pour contrôler le piezo.

2- Sachant que la gamme des fréquences d'un violon va de 200 à 2650 Hz, rédigez le protocole permettant d'accorder votre thérémine simplifié comme un violon.

On place la main sur la photorésistance, la luminosité est alors minimale, on règle la fréquence à une valeur extrême de la gamme, par exemple 200 Hz. Puis on place la photorésistance face à la lumière, et on règle sur l'autre valeur de fréquence, 2650 Hz. La luminosité mesurée par la photorésistance est proportionnelle à la fréquence jouée par le buzzer. Plus la luminosité reçue par la photorésistance est importante, plus la fréquence jouée par le buzzer sera élevée.

Réalisation.



1- Faites le montage suivant permettant de contrôler la fréquence.

2- Compléter le texte avec les noms des pins, (broches), et sachant que OUTPUT est la sortie et INPUT l'entrée.

Par ailleurs, vous aurez à faire un changement d'échelle avec la fonction map connaissant le fonctionnement des entrées analogiques.

a- Les entrées analogiques.

L'Arduino UNO est pourvu de 6 entrées analogiques. Ce sont les broches étiquetées A suivie d'un nombre. Les tensions, toujours entre 0 et 5V, présentes sur ces broches, peuvent être numérisées via un convertisseur analogique-numérique (CAN) ou ADC (Analog Digital Converter). La fonction analogRead(...) remplit ce rôle. Le convertisseur des Arduino effectue une conversion sur 10 bits, c'est à dire qu'il convertit la tension en un nombre entier ayant une valeur de 0 à 1023. 0 correspond au 0V de la tension et 1023 au 5V. La résolution, c'est à dire la différence entre deux valeurs successives de la tension correspondant à une différence de 1 sur l'entier résultat, est donc d'environ 5mV.

b- La fonction map.

Réétalonne un nombre d'une fourchette de valeur vers une autre fourchette. Ainsi, une valeur basse source sera étalonnée en une valeur basse de destination, une valeur haute source sera étalonnée en une valeur haute de destination, une valeur entre les deux valeurs source sera étalonnée en une valeur entre les deux valeurs destinations, en respectant la proportionnalité. Cette fonction est très utile pour effectuer des changements d'échelle automatiques.

Syntaxe.

```
map (valeur, limite_basse_source, limite_haute_source, limite_basse  
_destination, limite_haute_destination)
```

```
/*  
  Entrée et sortie analogiques + communications série  
  Ce programme va émettre un son via le buzzer branché sur la broche .....  
  La fréquence joué par le buzzer sera proportionnelle à la luminosité  
  captée par la photorésistance branchée sur la broche .....  
*/  
  
// Initialisation des constantes :  
const int analogInPin = .....; // Numéro de la broche à laquelle est connectée la  
photorésistance  
const int analogOutPin = .....; // Numéro de la broche à laquelle est connectée le  
buzzer  
  
int sensorValue = 0; // Valeur lue sur la photorésistance  
int outputValue = 0; // Valeur envoyée au buzzer  
  
void setup() {  
  // Initialise la communication avec l'ordinateur  
  Serial.begin(9600);  
  
  // Indique que la broche analogOutPin est une sortie :  
  pinMode(analogOutPin, .....);  
  // Indique que la broche analogInPin est une entrée :  
  pinMode(analogInPin, .....);  
}
```

```

}

void loop() {
  // lit la valeur de la photorésistance et stocke le résultat dans sensorValue :
  sensorValue = analogRead(analogInPin);
  // change sensorValue vers une intervalle de 200 à 2650 et stocke le résultat dans
  outputValue :
  outputValue = map(sensorValue, ..... , ..... , ..... , .....);
  // envoie de cette nouvelle valeur (en Hz) au buzzer
  tone(8, outputValue ,20);
  delay(10);

  // envoie tout ça vers l'ordinateur
  Serial.print("sensor = " );
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);
}

```

Correction

```

/*
  Entrée et sortie analogiques + communications série
  Ce programme va émettre un son via le buzzer branché sur la broche 8.
  La fréquence joué par le buzzer sera proportionnelle à la luminosité
  captée par la photorésistance branchée sur la broche A0.
*/
// Initialisation des constantes :
const int analogInPin = A0;    // Numéro de la broche à laquelle est connectée la
photorésistance
const int analogOutPin = 8;    // Numéro de la broche à laquelle est connectée le buzzer

int sensorValue = 0;          // Valeur lue sur la photorésistance
int outputValue = 0;          // Valeur envoyée au buzzer

void setup() {
  // Initialise la communication avec l'ordinateur
  Serial.begin(9600);

  // Indique que la broche analogOutPin est une sortie :
  pinMode(analogOutPin, OUTPUT);
  // Indique que la broche analogInPin est une entrée :
  pinMode(analogInPin, INPUT);
}

void loop() {
  // lit la valeur de la photorésistance et stocke le résultat dans sensorValue :
  sensorValue = analogRead(analogInPin);
  // change sensorValue vers une intervalle de 200 à 2650 et stocke le résultat dans
  outputValue :
  outputValue = map(sensorValue, 0, 1023, 200, 2650);
  // envoie de cette nouvelle valeur (en Hz) au buzzer
  tone(8, outputValue ,20);
  delay(10);

  // envoie tout ça vers l'ordinateur
  Serial.print("sensor = " );
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);
}

```

3- Comment garder son thérémine accordé ? Le code ci-dessous permet pendant 5 secondes de mesurer les valeurs maximale et minimale de l'intensité lumineuse ambiante. La LED sur le pin 13 est alors allumée pendant ce laps de temps. Ainsi le thérémine simplifié reste-t-il quelque soit les conditions de lumière accordé. Complétez ce nouveau code.

```
int sensorValue;
int sensorLow = .....;
int sensorHigh = .....;

const int ledPin = .....;

void setup()
{
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, HIGH);

  while (millis() < 5000) {
    sensorValue = analogRead(A0);
    if (sensorValue > sensorHigh){
      sensorHigh = sensorValue;
    }
    if (sensorValue < sensorLow){
      sensorLow = sensorValue;
    }
  }
  digitalWrite(ledPin, LOW);
}

void loop()
{
  sensorValue = analogRead(A0);
  int pitch = map(sensorValue, sensorLow, sensorHigh, 200, 2650);
  tone(8, pitch, 20);
  delay(10);
  Serial.println(analogRead(A0));
}
```

Correction

```
int sensorValue;
int sensorLow = 1023;
int sensorHigh = 0;

const int ledPin = 13;

void setup()
{
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, HIGH);

  while (millis() < 5000) {
    sensorValue = analogRead(A0);
```

```
    if (sensorValue > sensorHigh){
        sensorHigh = sensorValue;
    }
    if (sensorValue < sensorLow){
        sensorLow = sensorValue;
    }
}
digitalWrite(ledPin, LOW);
}

void loop()
{
    sensorValue = analogRead(A0);
    int pitch = map(sensorValue, sensorLow, sensorHigh, 200, 2650);
    tone(8, pitch, 20);
    delay(10);
    Serial.println(analogRead(A0));
}
```

4- Téléversez ce code et ... jouez !