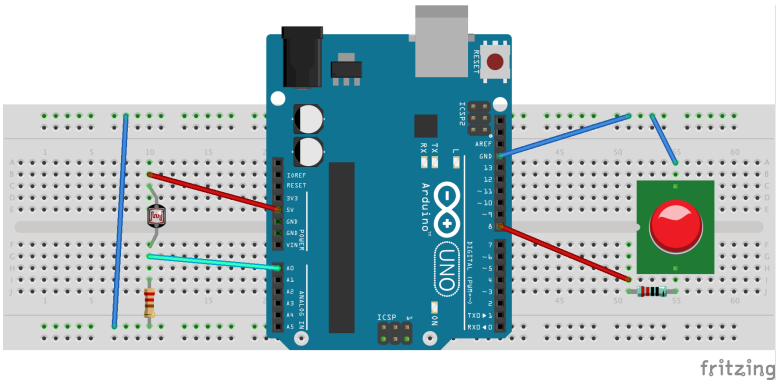
Arduino

Transmettre aux enfants le goût de la programmation, de la robotique et de l'ingénierie

Atelier Arduino Musique



DEVOXX 4KIDS

Si cet atelier vous a plu, vous pouvez refaire cet atelier chez vous avec un budget de moins de 100 €.

Il requiert :

- un Arduino Uno ou un clone (environ 15 €)
- une planche d'essai (breadboard)
- quelques LEDs
- un buzzer
- quelques résistances 220 Ω
- des fils
- pour les ateliers plus avancés une photo-résistance et une résistance 330 Ω

Il existe un starter kit officiel Arduino mais vous pouvez trouver ces composants dans presque tous les starter kits (nous avons des starter kits Farnell et Funduino).

Les supports sont disponibles sur le dépôt de documents de Devoxx4kids France

http://devoxx4kidsfr.github.io/materials/ateliers/arduino/index.html

Vous trouverez des informations plus générales sur nos activités et les différents ateliers sut le site Web de Devoxx4Kids France

http://www.devoxx4kids.org/france/ateliers/arduino/

DEVOXX™ 4KIDS

Arduino FM 01

Jouer une note

facile

Comment sont faits les synthétiseurs et les guitares électriques ?
Nous allons voir comment l'Arduino peut jouer des sons.

Nous aurons besoin de

L'Arduino Uno

1 Buzzer

1 résistance 220 Ω

Le programme JouerUneNote

Ouvre le programme JouerUneNote et sauve le sous un autre nom.

On va principalement utiliser la fonction `tone`. Si tu as Internet tu peux aller voir la description de cette fonction.

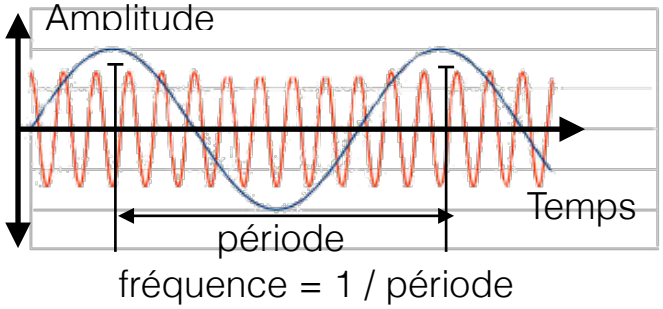
http://www.arduino.cc/en/Reference/Tone

Elle attend

- le pin sur lequel elle doit envoyer le courant
- la fréquence du son que l'on doit jouer
- la durée du son

Nous entendons les notes parce que nous percevons la vibration de l'air provoquée par l'instrument dans nos oreilles.

La note produit une vibration qui a une **fréquence** bien précise. La fréquence rouge est plus élevée. Le son paraîtra plus aigu, plus haut. La fréquence bleu est basse. Le sont paraîtra plus grave, plus bas.



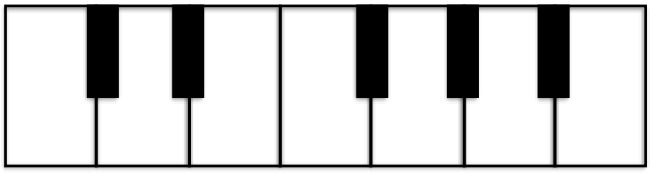
fréquence = 1 / période

L'**amplitude** (la différence entre le haut et le bas de la courbe) correspond au niveau sonote. Nous entendons le son plus fort quand l'amplitude est grande.

Les notes ont des fréquences fondamentales. La fréquence du La3 est 440 Hz (Hertz).

Observe que si on divise ou multiplie la fréquence par 2, on change d'octave.

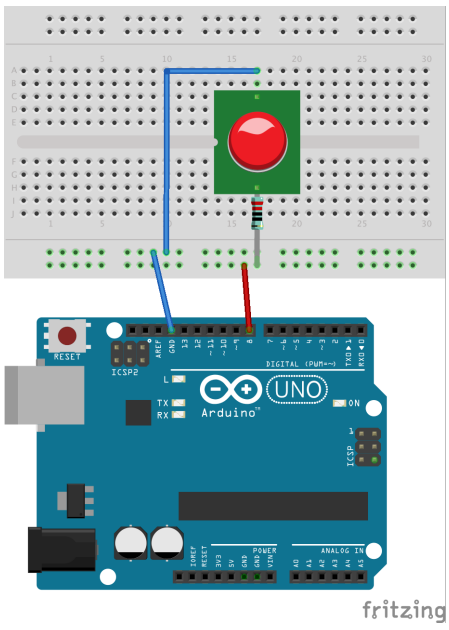
La fréquence des autres notes se calcule à partir du La. Le tableau contient le résultat pour 3 octaves.



Do2	Ré2	Mi2	Fa2	Sol2	La2	Si2
132	148,5	165	176	198	220	247,5
Do3	Ré3	Mi3	Fa3	Sol3	La3	Si3
264	297	330	352	396	440	495
Do4	Ré4	Mi4	Fa4	Sol4	La4	Si4
528	594	660	704	792	880	990

Le programme te permet de tester les notes en changeant `hauteur`. Tu peux aussi changer la durée en modifiant le 3ième paramètre de `tone`.

Le programme utilise le pin 8. Le rond rouge est le buzzer (qui est noir en fait). Ses pattes doivent être sur des rangées différentes.



DEVOXX 4KIDS

Arduino FM 02

Jouer un morceau

facile

Pour jouer plusieurs notes on pourrait écrire des tas de ligne tone en changeant la valeur.
Nous allons voir comment écrire le morceau plus simplement en utilisant un tableau et une boucle

Nous aurons besoin de

L'Arduino Uno

1 Buzzer

1 résistance 220 Ω

Le programme JouerUneNote

Tu peux reprendre le programme JouerUneNote.

Le Super Mario Mushroom Power Up est composé de 22 notes : 523, 392, 523, 659, 784, 1047, 784, 415, 523, 622, 831, 622, 831, 1046, 1244, 1661, 1244, 466, 587, 698, 932, 1195, 1397, 1865, 1397.

On pourrait les écrire l'une après l'autre :

```
void setup () {
  int DUREE = 34;
  tone(PIN_BUZZER, 523, DUREE);
  delay(DUREE * 1.4);
  noTone(PIN_BUZZER);
  tone(PIN_BUZZER, 392, DUREE);
  delay(DUREE * 1.4);
  noTone(PIN_BUZZER);
```

Hmmm …

On peut faire un peu mieux avec une fonction JouerUneNote

```
void jouerUneNote(int hauteur, long duree) {
  tone(PIN_BUZZER, hauteur, duree);
  delay(duree * 1.4);
  noTone(PIN_BUZZER);
}

void setup () {
  jouerUneNote(523);
  jouerUneNote(392);
  jouerUneNote(523);
  …
```

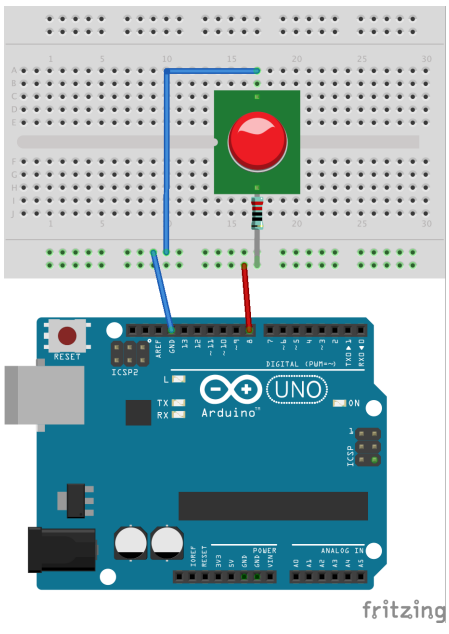
Mais il y a encore mieux. On peut écrire toutes les notes à la suite dans un **tableau** et lire le tableau dans une **boucle for.**

Le tableau notes défini une liste de nombres. On pourra accéder à la première note par notes[0], puis notes[1].

```
int notes[]= { 523, 392, 523, 659, 784, 1047, 784, 415,
523, 622, 831, 622, 831, 1046, 1244, 1661, 1244, 466, 587,
698, 932, 1195, 1397, 1865, 1397 };
int NB_NOTES = 22;
int DUREE = 34;
```

La boucle for va démarrer avec i qui vaut 0. Elle continue tant que i est < NB_NOTES et elle augmente de 1 à chaque passage. i sera utilisé dans le bloc qui est entre { }.

```
void setup() {
  for (int i=0; i<NB_NOTES; i++) {
    jouerUneNote(notes[i], DUREE);
  }
}
```



Le programme utilise le même circuit que JouerUneNote.

DEVOXX 4KIDS

Arduino FM 03

Le Thérémine

facile

Un thérémine est un instrument de musique électronique où le son est contrôlé par la position des mains.
On va construire un thérémine en utilisant le capteur de luminosité. Notre main va masquer la lumière et contrôler la hauteur de la note.

Nous aurons besoin de

L'Arduino Uno

1 Buzzer

1 résistance 220 Ω

La fiche photorésistance

Le programme Theremine

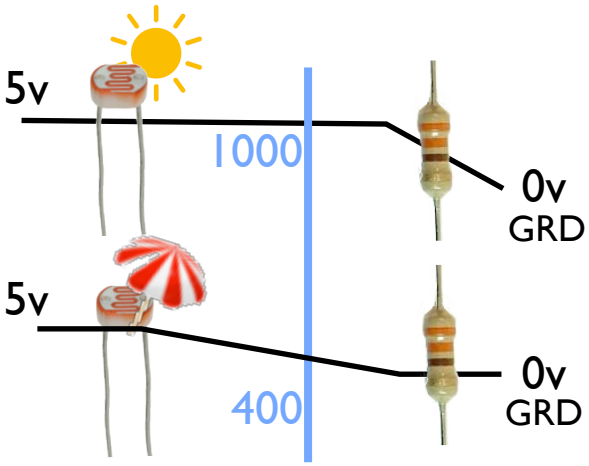
1 photo-résistance

1 résistance 330 Ω

Jusque là pour allumer et éteindre la lampe, nous avons passé tout le courant (HIGH) ou pas de courant du tout (LOW).
Il existe des résistances qui laissent plus ou moins passer de courant. Par exemple, la photo-résistance varie avec la lumière qu'elle reçoit et va nous permettre de lire une valeur variable.

Tu trouveras plus d'explications sur la fiche photorésistance.

La mesure relevée en sortie de la photorésistante va permettre de calculer la hauteur de la note.



AnalogRead sur le pin A0

Le programme calcule la hauteur de la note et l'envoie sur le pin 8

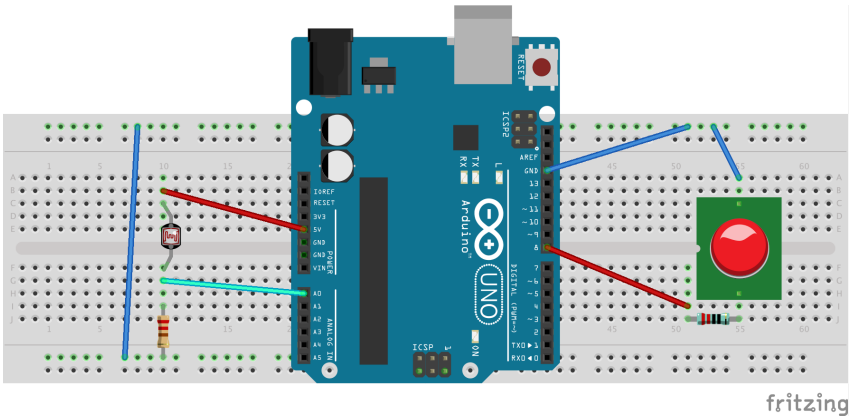
Tu auras besoin de la fonction map. Elle permet de convertir les valeurs qu'envoie la photorésistance entre 400 et 1000, en des hauteurs de note entre 132 (Do2) et 990 (Si4).

```
int frequenceNote = map(lumiere, 400, 1000, 132, 990);
```

Il te faudra ensuite `tone` pour jouer la note sur le buzzer.

Essaie de changer les valeurs dans map pour voir comment se comporte ton instrument. Tu peux utiliser `Serial.Begin` et `Serial.println` pour voir ce qui se passe.

Tu peux repartir du circuit du buzzer. Il faut ajouter la partie du circuit qui lit les données.



DEVOXX 4KIDS

Arduino AM 04

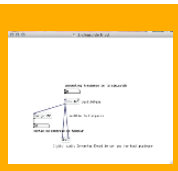
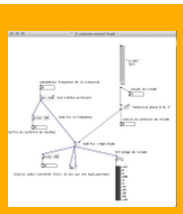
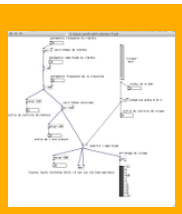
Le synthétiseur

Avancé

Nous allons jouer avec un synthétiseur et comprendre comment la musique est produite par l'ordinateur.

Nous aurons besoin de

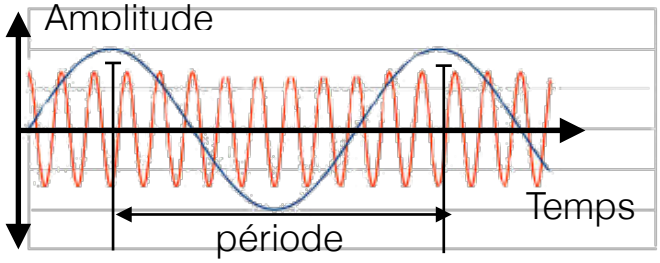
Le logiciel Pure Data

Les patch Pure Data

Installe Pure Data s'il n'est pas déjà disponible.

Notre oreille détermine la hauteur du son selon la fréquence de la vibration de l'air, et le volume selon l'amplitude de la vibration.



fréquence = 1 / période

Nous percevons un son plus fort quand l'amplitude de la vibration est plus importante.

Pure Data fourni des opérations qui permettent de générer des sons, modifier leur fréquence et leur volume.

On pourra ensuite connecter Pure Data à l'Arduino.

Mais d'abord il faut comprendre comment Pure Data fonctionne.

DEVOXX 4KIDS

Pure Data :

Pure Data est un outil de programmation visuelle qui permet de manipuler les sons et les vidéo.

Dans Pure Data, un programme s'appelle un patch. Pour écrire un programme, il faut ajouter des boîtes et les connecter en elles pour décrire le processus.

Les boîtes contiennent diverses opérations, des paramètres, des outils de contrôle permettant à l'utilisateur de modifier les valeurs.

Ouvre les patch et joue avec le synthétiseur.

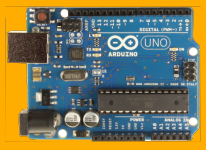
- Prend le temps de comprendre comment le programme est fait.
- Assure toi que tu es en Edit Mode si tu veux modifier le patch
- Le DSP (Digital Signal Processeur) n'est pas activé par défaut. Il faut le démarrer dans la fenêtre principale de Pure Data ou par le menu.

Arduino AM 05

SynthéArduino
Avancé

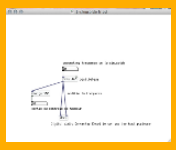
Nous allons connecter le synthétiseur à l'Arduino pour le commander via l'Arduino.

Nous aurons besoin de



L'Arduino Uno

Le logiciel Pure Data



Le patch arduino_example

Le programme PureData_example

Programme de démo

Ce programme envoie des valeurs croissantes puis décroissantes sur le port série (le Serial que nous avons utilisé pour les print).
C'est le composant comport du patch qui lit les données sur le port série.

- Charge le programme Arduino PureData_Example et téléverse le sur l'Arduino
- Démarre Pure Data (pd-extended sur le Mac)
- Charge le programme Pure Data arduino_example.pd
- Suit les instructions en commentaire dans le programme Arduino. Il faut
  - lister les devices pour choisir un des ports de l'Arduino,
  - sélectionner le bon port dans le paramètre de open
  - lancer le patch Pure Data en cliquant sur open
  - cliquer sur DSP

Contrôle par l'Arduino

Reprend le circuit et le code du Thérémine.

Modifie le programme pour que les données lues sur le connecteur analogique relié à la photo-résistance soit écrites sur le port série comme dans la démo.

Tu peux maintenant contrôler la fréquence du synthétiseur avec ta main via la photo-résistance.

Programmes plus avancé

En reprenant les exemples de contrôle de volume et du vibrato et la partie comport qui permet d'intégrer les données du port série au patch tu peux contrôler le volume ou le vibrato avec la photo-résistance ou un potentiomètre (voir AM06 pour la solution)