



CE QUE NOUS ALLONS FAIRE:

A ce jour, nous avons contrôlé de la lumière, des mouvements et des électrons. C'est parti pour le son. Cependant le son est un phénomène analogique, comment faire avec notre Arduino ? Encore une fois, nous allons utiliser son incroyable rapidité qui lui permet d'imiter un fonctionnement analogique. Pour cela nous allons connecter un buzzer à une des pattes numériques de l'Arduino. Un buzzer émet un son à chaque fois qu'un courant le traverse. Si nous lui envoyons un signal à la bonne fréquence (par exemple 440 fois par seconde pour obtenir un LA). C'est parti pour expérimenter tout ça et faire jouer "Twinkle Twinkle Little Star" à votre Arduino.

LE CIRCUIT:

Composants :



CIRC-06
Feuille de Connexions
x1



Connecteur 2 patte
x4

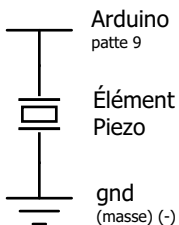


Élément Piezo
x1



Fil

Schéma



Internet

..Télécharger..

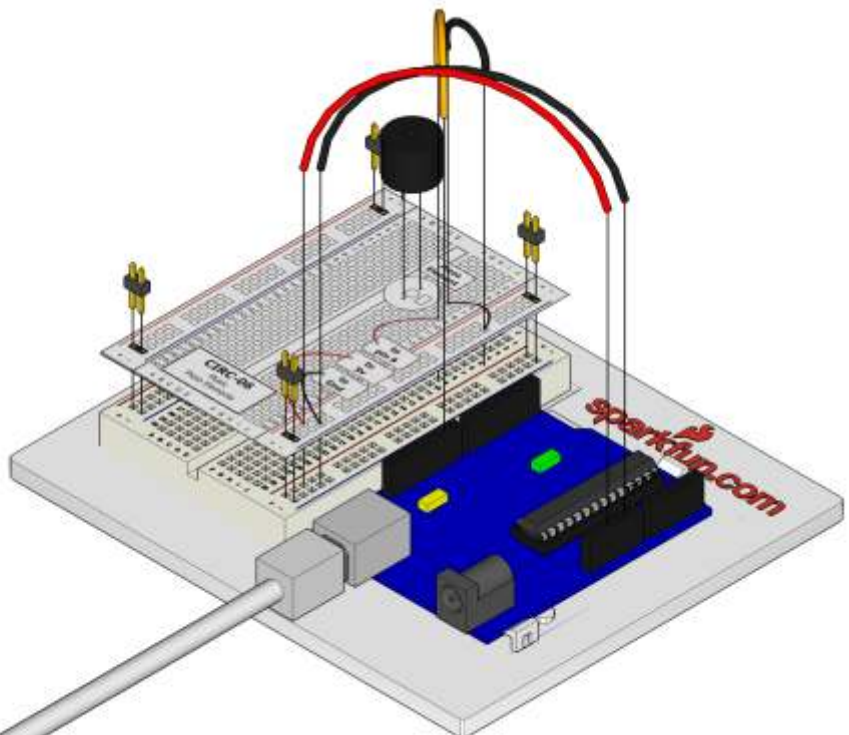
Feuille de Connexions

<http://ardx.org/BBL506S>

..Voir..

Vidéo de Montage

<http://ardx.org/VIDE06>



CODE (Il n'est pas nécessaire de taper quoi que ce soit)

Téléchargez le code (<http://ardx.org/CODE06>)

(et ensuite copiez le texte dans un nouveau sketch Arduino)

```

/*Mélodie
 * (cleft) 2005 D. Cuartielles pour K3
 *
 *Cet exemple utilise un buzzer piezo pour jouer des mélodies.
 *Il envoie
 *un signal carré de la fréquence adéquate au buzzer,
 *générant ainsi le ton correspondant.
 *
 *Le calcul du ton est effectué en suivant cette
 *opération mathématique :
 *
 *      tempsHaut = période / 2 = 1/(2 * fréquenceTon) <br> *
 *ou les différents tons sont décrits dans la table:
 *
 *      note      Fréquence période      tempsHaut
 *      c          261 Hz      3830      1915
 *      d          294 Hz      3400      1700
 *      e          329 Hz      3038      1519
 *      f          349 Hz      2864      1432
 *      g          392 Hz      2550      1275
 *      a          440 Hz      2272      1136
 *      b          493 Hz      2028      1014
 *      c          523 Hz      1912      956
 *
 *      http://www.arduino.cc/en/Tutorial/Melody
 */

int speakerPin = 9;
int length = 15; //Le nombre de notes<br>char notes[] =
"ccggaagffeeddc "; //un espace représente un silence<br>int
beats[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 4 };
int tempo = 300;

void playTone(int tone, int duration) {
  for (long i = 0; i < duration * 1000L; i += tone * 2) {
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(tone);
    digitalWrite(speakerPin, LOW);
    delayMicroseconds(tone);
  }
}

void playNote(char note, int duration) {
  char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'c' };
  int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };
  //joue le ton correspondant au nom de la note<br> for (int
  i = 0; i < 8; i++) {
    if (names[i] == note) {
      playTone(tones[i], duration);
    }
  }
}

void setup() {
  pinMode(speakerPin, OUTPUT);
}

void loop() {
  for (int i = 0; i < length; i++) {
    if (notes[i] == ' ') {
      delay(beats[i] * tempo); //silence<br> } else {
      playNote(names[i], beats[i] * tempo);
    }
    //pause entre les notes<br> delay(tempo / 2); }
  }
}

```

CELA FONCTIONNE PAS ? (3 choses à essayer)

Le son ne fonctionne pas

Vu la taille du buzzer, c'est facile de se tromper de trou pour les branchements, vérifiez-les une seconde fois

Je n'arrive pas à réfléchir avec tout ce bruit !

Il suffit de retirer le buzzer pendant que vous réfléchissez, ensuite chargez le programme et rebranchez le buzzer.

Marre de Twinkle Twinkle Little Star?

Le code est écrit de manière à ce que vous puissiez aisément ajouter vos propres musiques, allez faire un tour sur le code ci-dessous pour commencer.

AMÉLIORER LE MONTAGE

Jouer avec la vitesse:

la temps de chaque note est calculé à partir de variables, nous pouvons donc modifier la durée de chaque note ou son timing. Pour changer la vitesse de la mélodie vous n'avez besoin de changer qu'une ligne.
int tempo = 300; ---> int tempo = (Nouveau N°)
Changez le pour un nombre plus grand pour ralentir la mélodie ou pour un nombre plus petit pour l'accélérer.

Modifier les notes :

Si vous pensez que les notes ne sont pas aux bons tons, cela peut aussi être corrigé. Les notes ont été calculées sur la base de la formule commentée dans le bloc au début du programme. Pour modifier les notes individuellement ajustez les valeurs dans tones[] jusqu'à ce qu'elles soient bonnes. (chaque note correspond à son nom dans names[] (ex : c = 1915)

```
char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'c' };
```

```
int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };
```

Composer vos propres mélodies :

Le programme est configuré pour jouer 'Twinkle Twinkle Little Star' mais la façon dont le programme est fait vous permet de le modifier simplement. Chaque musique est définie en un int et deux tableaux, l'int length définit le nombre de notes, le premier tableau notes[] définit chaque note et le second beats[] définit le temps pendant lequel chaque note est jouée. Quelques exemples :

Twinkle Twinkle Little Star

```
int length = 15;
char notes[] = {"ccggaagffeeddc "};
int beats[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 4 };
```

Joyeux Anniversaire (première ligne)

```
int length = 13;
char notes[] = {"ccdcfcgccg "};
int beats[] = {1,1,1,1,1,2,1,1,1,1,1,2,4};
```

PLUS, PLUS, PLUS :

Plus de détails, où acheter des composants, où poser plus de questions :

<http://ardx.org/CIRC06>