

# MEETUP MAKER GIRLS

---

## Arduino et Capteurs

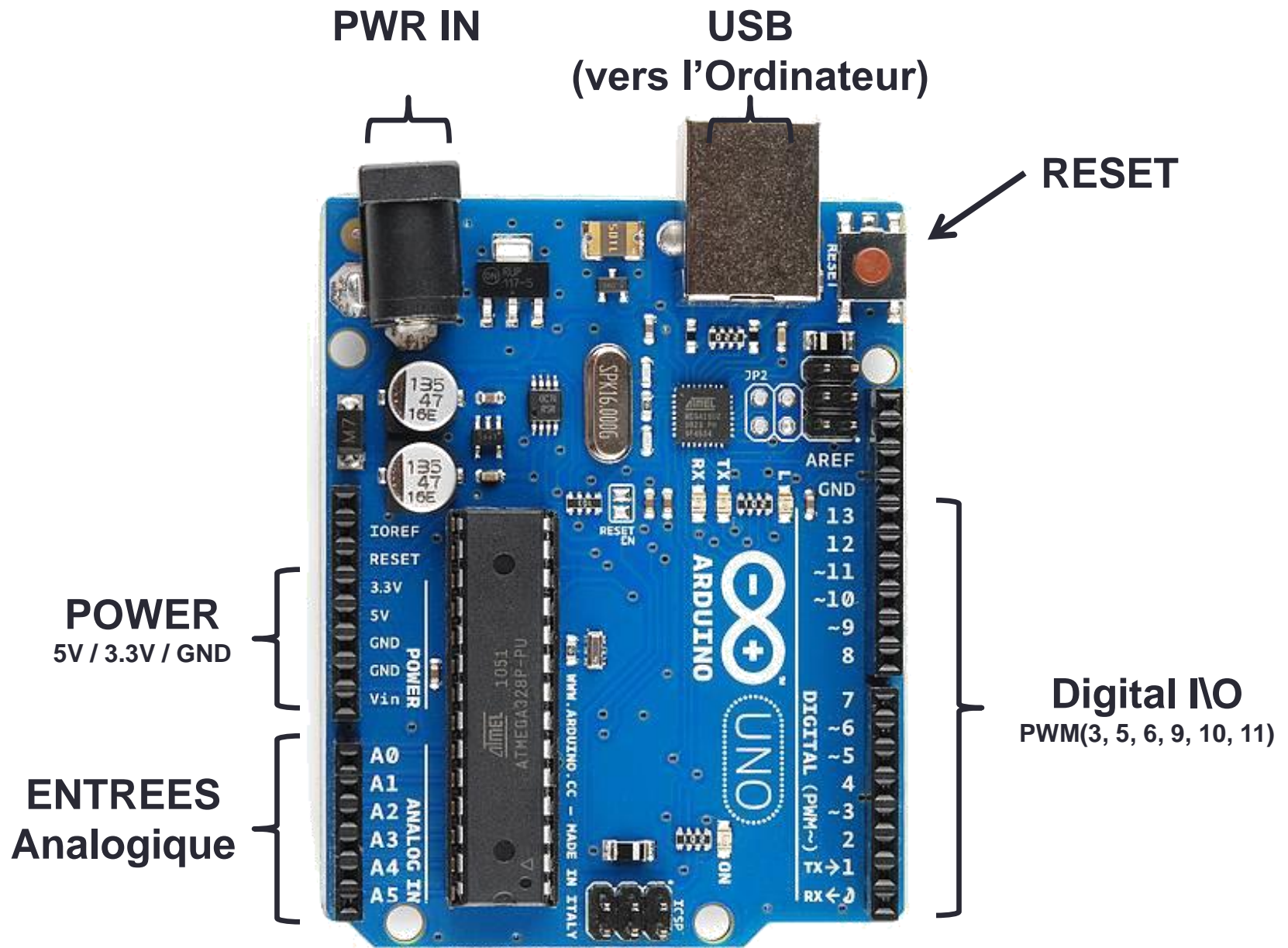
(Tutorial basé sur celui de Limor Fried qui est disponible à <http://www.ladyada.net/learn/arduino> et celui de Sparkfun disponible à <https://learn.sparkfun.com/tutorials/what-is-an-arduino>, et celui sur <http://forum.snootlab.com>, ainsi que du matériel pris du site Arduino (<http://www.arduino.cc/>), tous les trois sous une licence [CC BY-NC-SA 3.0](#), ce qui est aussi la licence de ce tutorial)

# Sommaire

- Petit rappel
- Le buzzer
- La photorésistance

# Sommaire

- Petit rappel
- Le buzzer
- La photorésistance



# La Charge Electrique

- La **tension**
  - La différence de charge entre deux points
- Le **courant**
  - La vitesse à laquelle la charge s'écoule
- La **résistance**
  - La tendance d'un matériau à résister à l'écoulement de la charge (courant)
- Un **circuit** est une boucle fermée qui permet la charge de se déplacer d'un endroit à l'autre.
- Les **composants** dans le circuit nous permettent de contrôler cette charge et l'utiliser pour faire du travail

# Symboles d'un Schéma du Circuit :

## Rappel



Une résistance

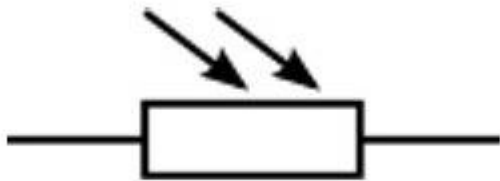


Une DEL

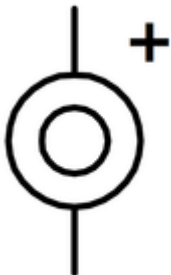


L'alimentation électrique  
et la terre

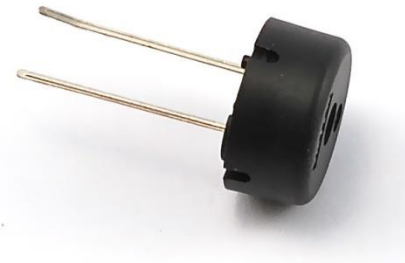
# Symboles d'un Schéma du Circuit



Une photorésistance



Un buzzer



# Sommaire

- Petit rappel
- **Le buzzer**
- La photorésistance



# La piézoélectricité

- De Wikipedia: « La **piézoélectricité** est la propriété que possèdent certains corps de se polariser électriquement sous l'action d'une contrainte mécanique et réciproquement de se déformer lorsqu'on leur applique un champ électrique. »
- Dans le cas d'un buzzer, il se déforme avec l'application d'une tension, et crée du son

# Comment ca marche

Le son consiste des vibrations dans l'air.

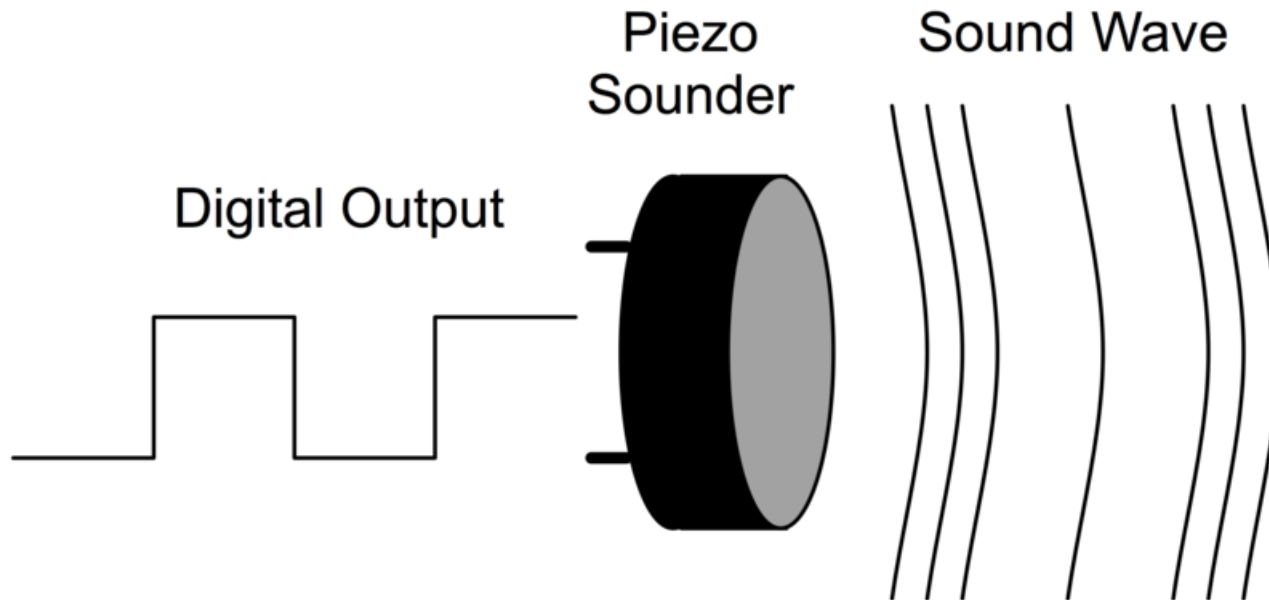
La vitesse de ces vibrations (cycles par seconde, ou Hertz) donne le ton.

Plus la vitesse des vibrations (la fréquence) augmente, plus le ton augmente.

Par exemple, do central a une fréquence de 261 Hz. Si vous basculez une sortie numérique 261 fois par seconde, la sortie va etre do central.

Pour entendre la sortie, il faut la connecter à quelquechose qui va convertir le signal électrique aux ondes sonores.

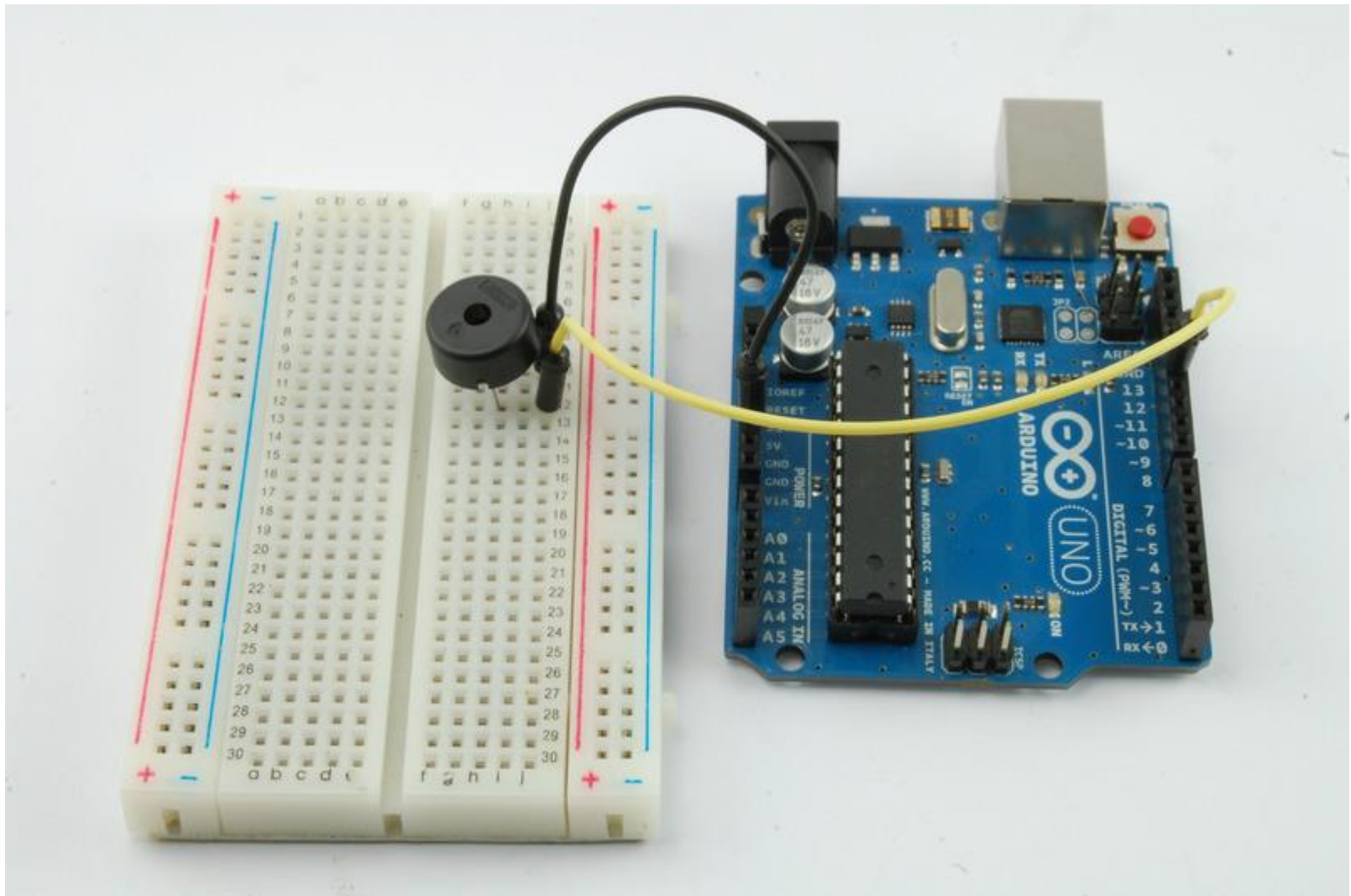
La piézo ici a un crystal qui agrandit et se contracte avec le passage du signal. C'est cela qui va créer du son que l'on peut entendre



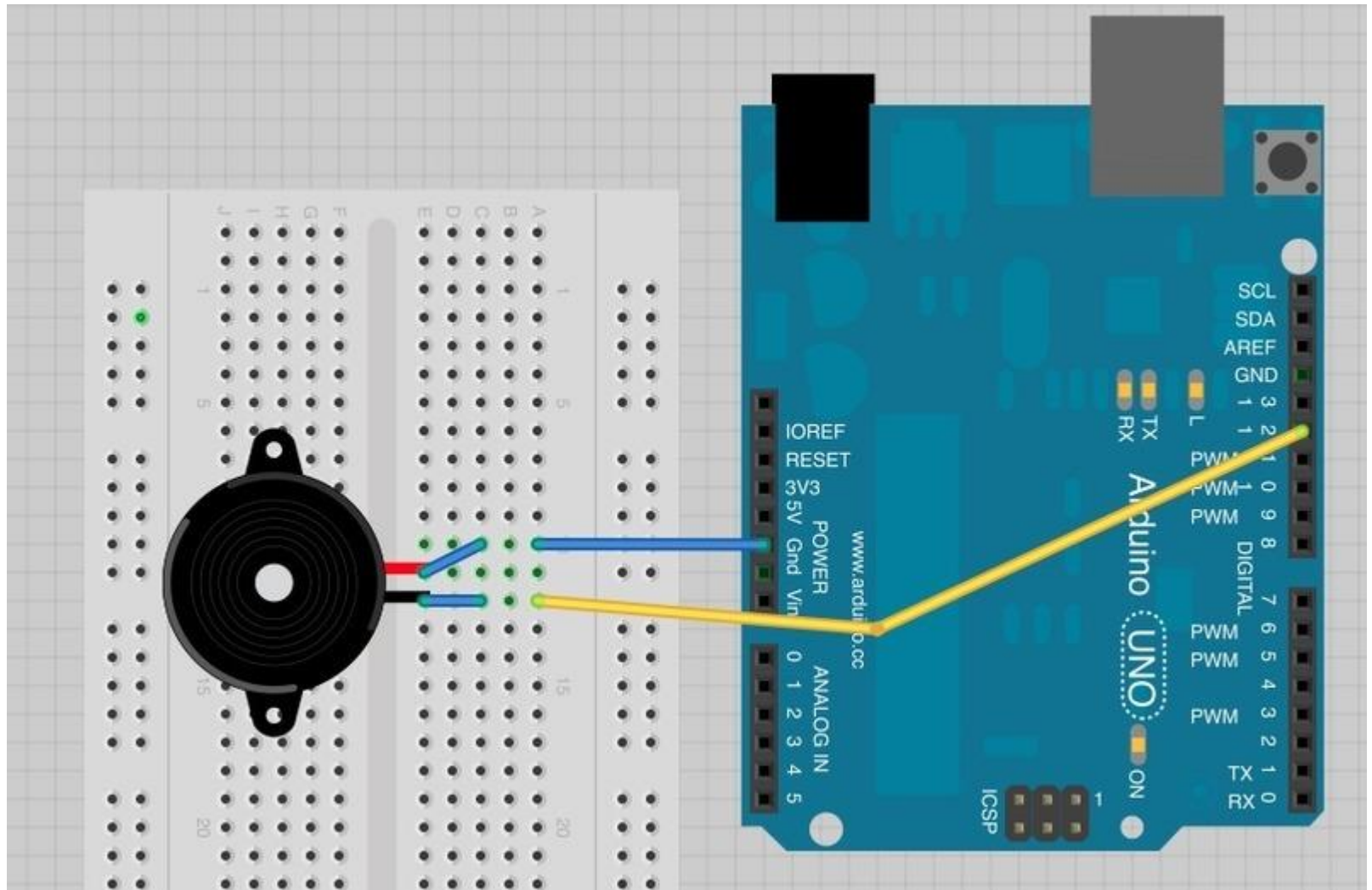
# L'interface avec l'Arduino

- L'Arduino vient avec une fonction « `tone()` » qui génère un signal carré, d'une fréquence donnée, sur la pin spécifiée
- Deux utilisations:
  - `tone(pin, fréquence);`
  - `tone(pin, fréquence, durée);`
  - Fréquence: en Hertz
  - Durée: en millisecondes
- Une fonction « `noTone()` » existe également pour arrêter le son

# Schéma avec un buzzer



# Schéma avec un buzzer (détails)



# Le code

```
/* Code pour schéma avec un buzzer */

int speakerPin = 12;
int numTones = 10;

int tones[] = {261, 277, 294, 311, 330, 349, 370, 392, 415, 440};
//          mid C  C#  D   D#  E   F   F#  G   G#  A

void setup(){
    for (int i = 0; i < numTones; i++) {
        tone(speakerPin, tones[i]);
        delay(500);
    }
    noTone(speakerPin);
}

void loop(){
}
```

# Le code

```
/* Code pour schéma avec un buzzer */
```

```
int speakerPin = 12;  
int numTones = 10;
```

```
int tones[] = {261, 277, 294, 311, 330, 349, 370, 392, 415, 440};  
//           mid C  C#  D   D#  E   F   F#  G   G#  A
```

```
void setup(){  
    for (int i = 0; i < numTones; i++) {  
        tone(speakerPin, tones[i]);  
        delay(500);  
    }  
    noTone(speakerPin);  
}
```

```
void loop(){  
}
```

Tableau => comme une liste



# Le code


```
/* Code pour schéma avec un buzzer */

int speakerPin = 12;
int numTones = 10;

int tones[] = {261, 277, 294, 311, 330, 349, 370, 392, 415, 440},
//           mid C  C#  D   D#  E   F   F#  G
void setup(){
    for (int i = 0; i < numTones; i++) {
        tone(speakerPin, tones[i]);
        delay(500);
    }
    noTone(speakerPin);
}

void loop(){
}
```

Une boucle « for » compte de 0 à 9 avec le variable « i »





# Le code

```
/* Code pour schéma avec un buzzer */


int speakerPin = 12;
int numTones = 10;

int tones[] = {261, 277, 294, 311, 330, 349, 370, 392, 415, 440};
//          mid C  C#  D   D#  E   F   F#  G

void setup(){
    for (int i = 0; i < numTones; i++) {
        tone(speakerPin, tones[i]);
        delay(500);
    }
    noTone(speakerPin);
}

void loop(){
}
```

A chaque étape de la boucle, on prend la prochaine fréquence dans la liste. On accède à chaque élément dans la liste avec le variable « i »



# Le code

```
/* Code pour schéma avec un buzzer */


int speakerPin = 12;
int numTones = 10;

int tones[] = {261, 277, 294, 311, 330, 349, 370, 392, 415, 440};
//          mid C  C#  D   D#  E   F   F#  G

void setup(){
    for (int i = 0; i < numTones; i++) {
        tone(speakerPin, tones[i]);
        delay(500);
    }
    noTone(speakerPin);
}

void loop(){
}
```

La fonction « tone » prend en entrée deux paramètres: le premier est la broche à laquelle on va envoyer le ton; le deuxième est la fréquence



# Le code

```
/* Code pour schéma avec un buzzer */

int speakerPin = 12;
int numTones = 10;

int tones[] = {261, 277, 294, 311, 330, 349, 370, 392, 415, 440};
//           mid C  C#  D   D#  E   F   F#  G   G#  A

void setup(){
    for (int i = 0; i < numTones; i++) {
        tone(speakerPin, tones[i]);
        delay(500);
    }
    noTone(speakerPin);
}

void loop(){
}
```

Quand toutes les fréquences ont été envoyées, on sort de la boucle et la fonction « noTone » est appelée. Cette fonction prend en entrée un paramètre: la broche associée.

# Le code

```
/* Code pour schéma avec un buzzer */

int speakerPin = 12;
int numTones = 10;

int tones[] = {261, 277, 294, 311, 330, 349, 370, 392, 415, 440};
//           mid C  C#  D   D#  E   F   F#  G   G#  A

void setup(){
    for (int i = 0; i < numTones; i++) {
        tone(speakerPin, tones[i]);
        delay(500);
    }
    noTone(speakerPin);
}

void loop(){
}
```

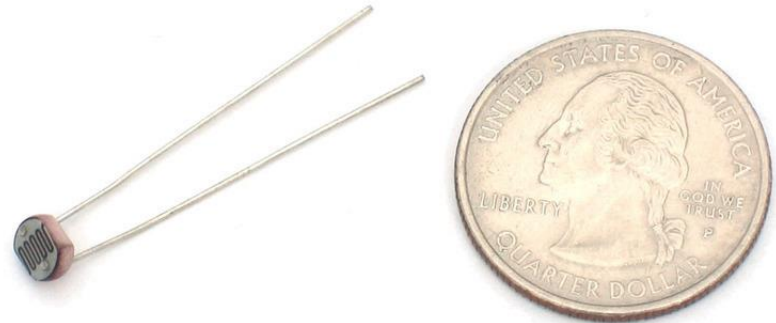
Cette procédure a été mise dans le « setup ». Ca veut dire qu'elle ne va être exécutée qu'une seule fois à chaque reset. Si on le mettais dans la fonction « loop », la procédure tournerait en boucle.

# Sommaire

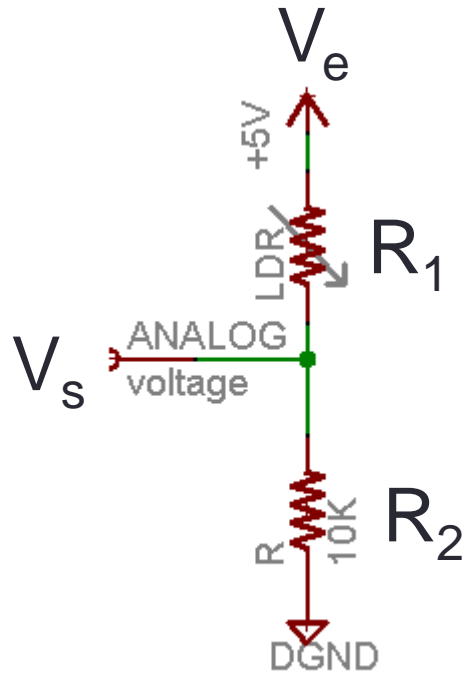
- Petit rappel
- Le buzzer
- La photorésistance

# La photorésistance

- Détection de la lumière
- Une résistance qui change sa valeur de résistance en fonction de la lumière sur sa façade
- Pour cette photorésistance, la résistance varie de  $200\text{k}\Omega$  (sombre) à  $12\text{k}\Omega$  (éclairé)

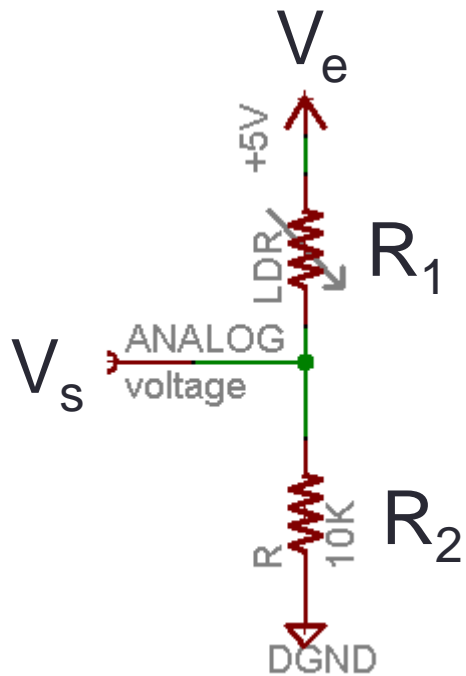


# Comment l'utiliser : la division de tension



$$V_s = V_e \times \frac{R_2}{R_1 + R_2}$$

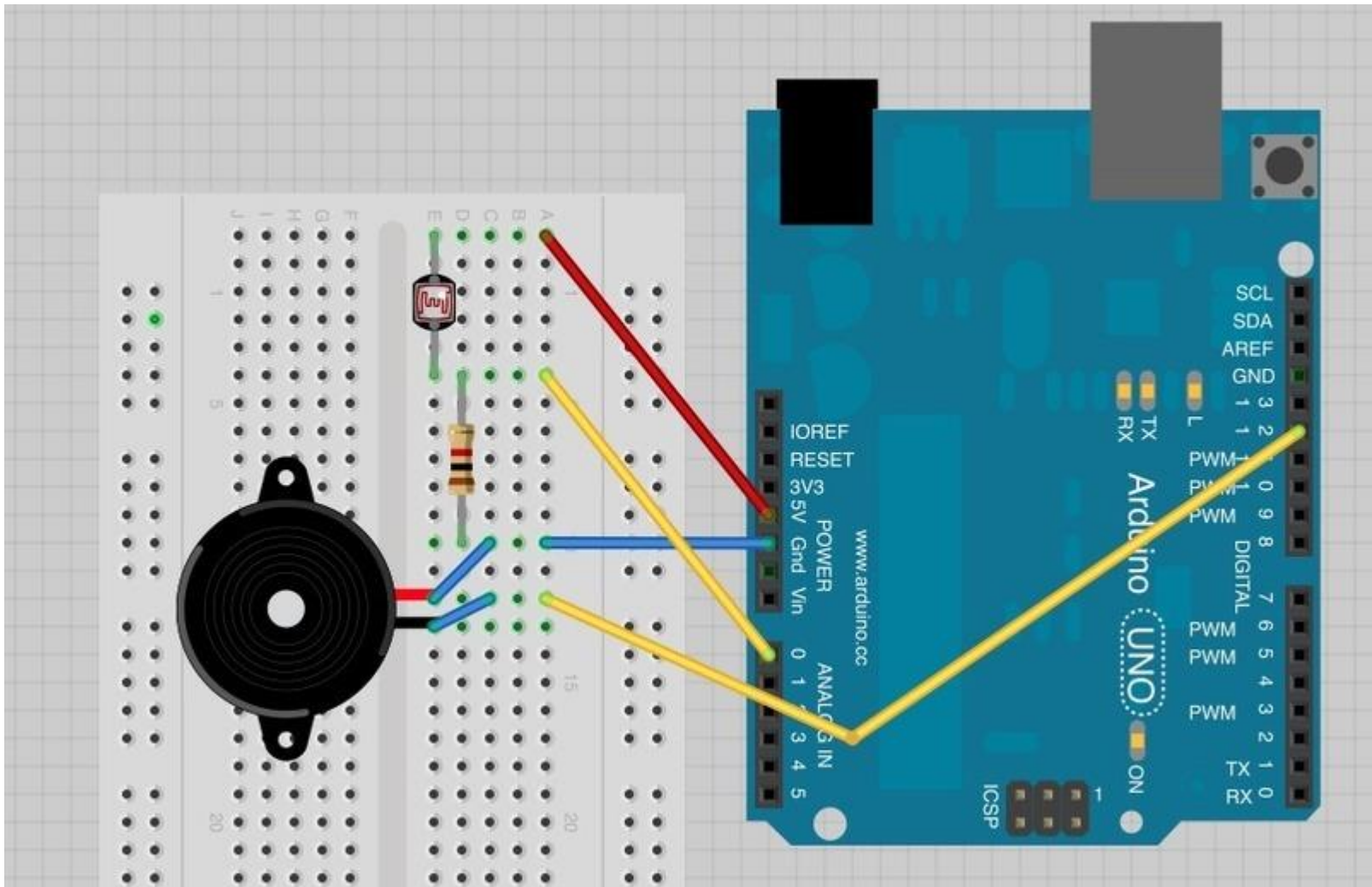
# Comment l'utiliser : la division de tension



Niveau de la lumière	$R_1$	$V_s$
Eclairé	12k $\Omega$	2,27V
Sombre	200k $\Omega$	0,24V



# Comment l'utiliser: le circuit



# Le code Arduino

```
int brocheBuzzer = 12;
int brochePhotoResistance = 0;

void setup(){
}

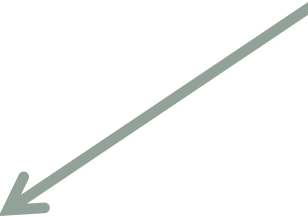
void loop(){
    int lecture = analogRead(brochePhotoResistance);
    int frequence = 200 + lecture / 4;
    tone(brocheBuzzer, frequence);
}
```

# Le code Arduino

```
int brocheBuzzer = 12;
int brochePhotoResistance = 0;

void setup(){
}

void loop(){
    int lecture = analogRead(brochePhotoResistance);
    int frequence = 200 + lecture / 4;
    tone(brocheBuzzer, frequence);
}
```




La fonction `analogRead(<N>)` lit l'entrée de la broche « N », lié à l'intensité de la lumière.

# Le code Arduino

```
int brocheBuzzer = 12;  
int brochePhotoResistance = 0;  
  
void setup(){  
}  
  
void loop(){  
    int lecture = analogRead(brochePhotoResistance);  
    int frequency = 200 + lecture / 4;  
    tone(brocheBuzzer, frequency);  
}
```

On force la fréquence de base d'être 200 Hz




# Le code Arduino

```
int brocheBuzzer = 12;
int brochePhotoResistance = 0;

void setup(){
}

void loop(){
    int lecture = analogRead(brochePhotoResistance);
    int frequency = 200 + lecture / 4;
    tone(brocheBuzzer, frequency);
}
```

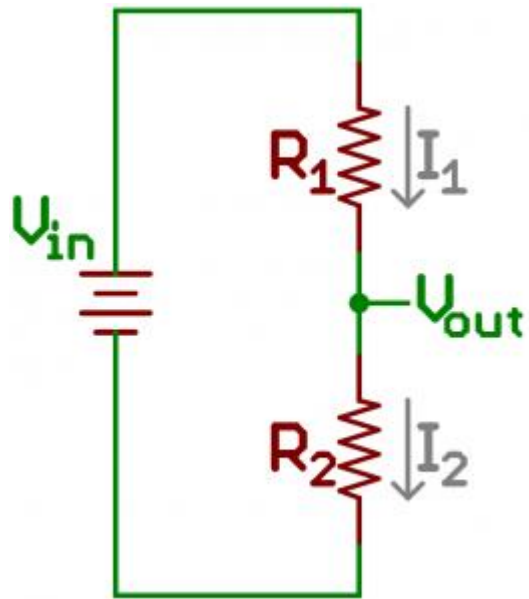
On envoie cette fréquence  
calculée vers le buzzer  
(attaché à la brocheBuzzer)



# Le code Arduino

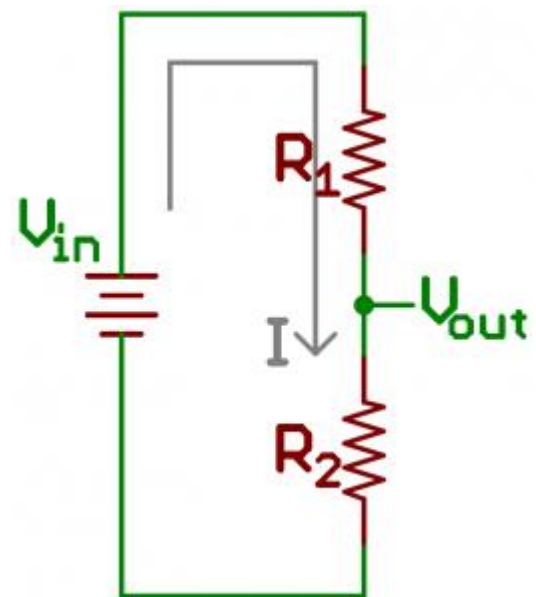
- Changez la valeur 4 pour changer la tranche des fréquences
- Changez les valeurs des fréquences dans le premier exercice pour jouer de la musique

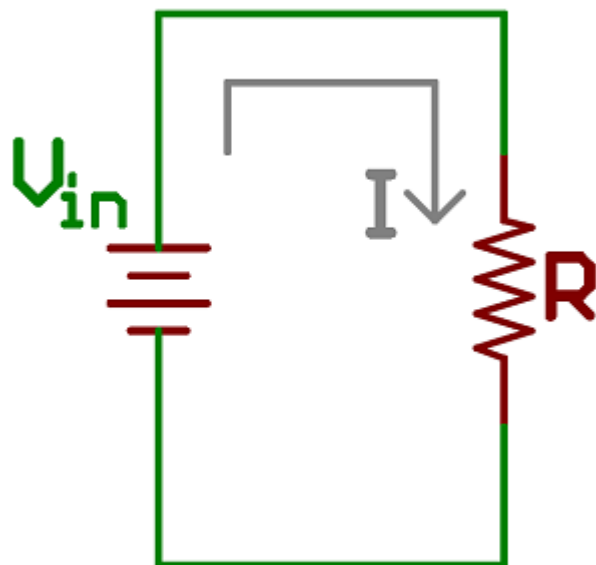
# Annexe



$$V_{out} = I_2 \cdot R_2$$







$$R = R_1 + R_2$$

$$I = \frac{V_{in}}{R_1 + R_2}$$

$$V_{out} = R_2 \cdot \frac{V_{in}}{R_1 + R_2}$$