

Objectif

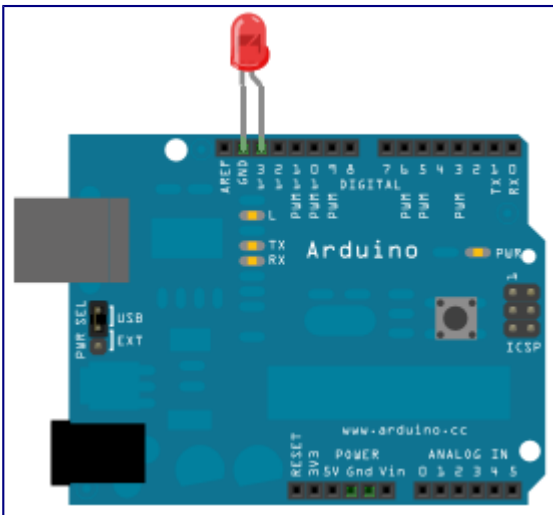
Faire clignoter une LED

Matériel

- une LED de la couleur que vous voulez

Montage

Réalisez ce montage :



Petite précision : sur la pin 13 de l'arduino, il y a déjà une résistance c'est pourquoi on peut se permettre d'y connecter directement la led!

Programmation

Copiez-collez ce code dans l'IDE Arduino :

Vous trouvez aussi le code par le menu Fichier->Exemples->Basics->Blink

```
/*  
Allume la LED pendant 1 seconde, puis l'éteint pendant 1 seconde.  
*/  
// Numéro de la broche à laquelle est connectée la LED  
int led = 13;  
// le code dans cette fonction est exécuté une fois au début  
void setup() {  
  
  // on va modifier sa tension  
  pinMode(led, OUTPUT);    // indique que la broche de la LED une sortie  
}  
// le code dans cette fonction est exécuté en boucle  
void loop() {  
  digitalWrite(led, HIGH); // allumer la LED, tension 5V sur la broche  
  
  delay(1000); // attendre 1000ms=1s  
  digitalWrite(led, LOW); // éteindre la LED, tension 0V sur la broche  
  delay(1000); // attendre 1 seconde  
}
```

Objectif

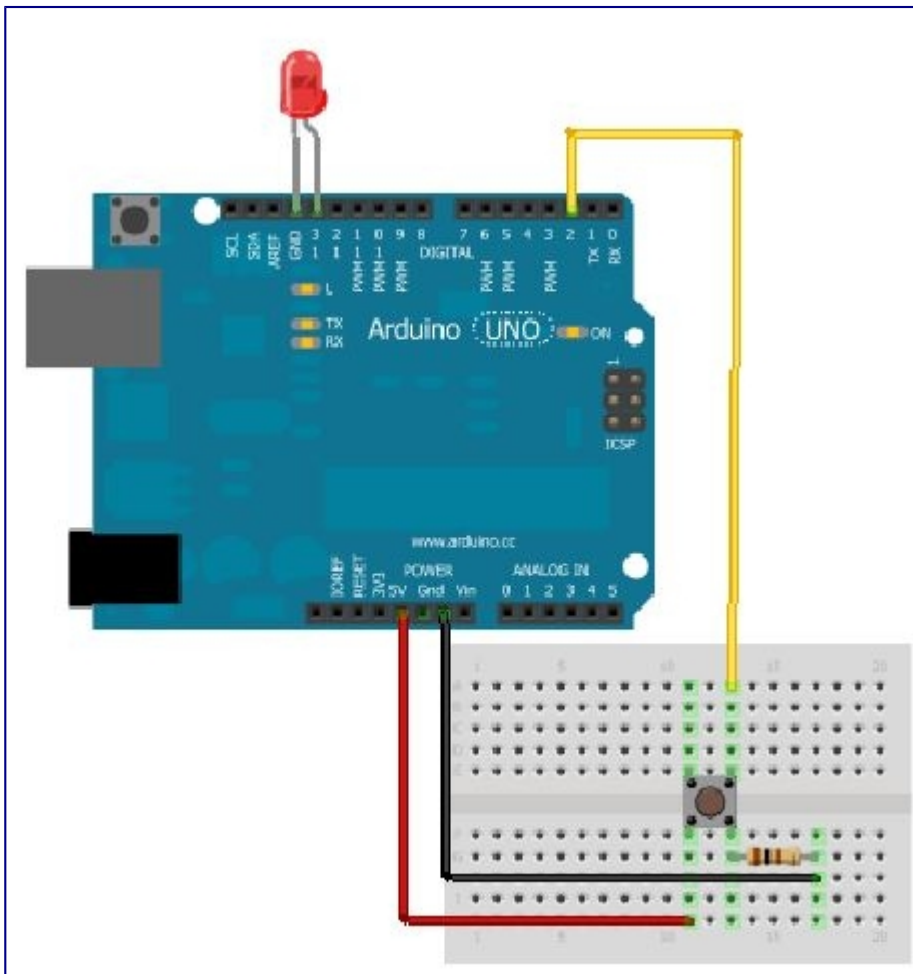
Détection quand un bouton est appuyé pour allumer une LED.

Matériel

- une LED
- un bouton poussoir
- une breadboard (plaque à trous pour connecter des composants)
- des fils
- une résistance de 10 k Ω

Montage

Réalisez ce montage :



Programmation

```
/*
Allumage de LED par bouton.
Allume la LED quand on appuie sur le bouton.
*/

// Numéro de la broche à laquelle est connecté le bouton
const int boutonPin = 2;
// Numéro de la broche à laquelle est connectée la LED
const int ledPin = 13;

void setup() {
// indique que la broche de la LED est utilisée en sortie :
// on va modifier sa tension
pinMode(ledPin, OUTPUT);

// indique que la broche du bouton est utilisée en entrée :
// on va lire sa tension
pinMode(boutonPin, INPUT);
}

void loop() {
// lit l'état du bouton : appuyé si on lit 5V
// éteint si on lit 0 V
int boutonState = digitalRead(boutonPin);
delay(20); // anti rebond

// Condition sur l'état du bouton
if(boutonState == HIGH) {
// s'il est à 5V (HIGH)
// on allume la LED
digitalWrite(ledPin, HIGH);
} else {
// sinon
// on éteint la LED
digitalWrite(ledPin, LOW);
}
}
```

Lisez un peu le code pour comprendre ce qu'il fait. Ensuite appuyez sur «**Téléverser**».

Essayez d'appuyer sur le bouton : la LED devrait s'allumer.

Astuces

- La **breadboard** est pratique pour faire les branchements : les trous proches sont connectés électriquement en colonnes (traces vertes sur le schéma), ce qui permet de brancher des composants ensemble.
- Toutes les broches de l'Arduino peuvent être utilisées soit en **sortie** (on modifie la tension), soit en **entrée** (on lit la tension).

Objectif

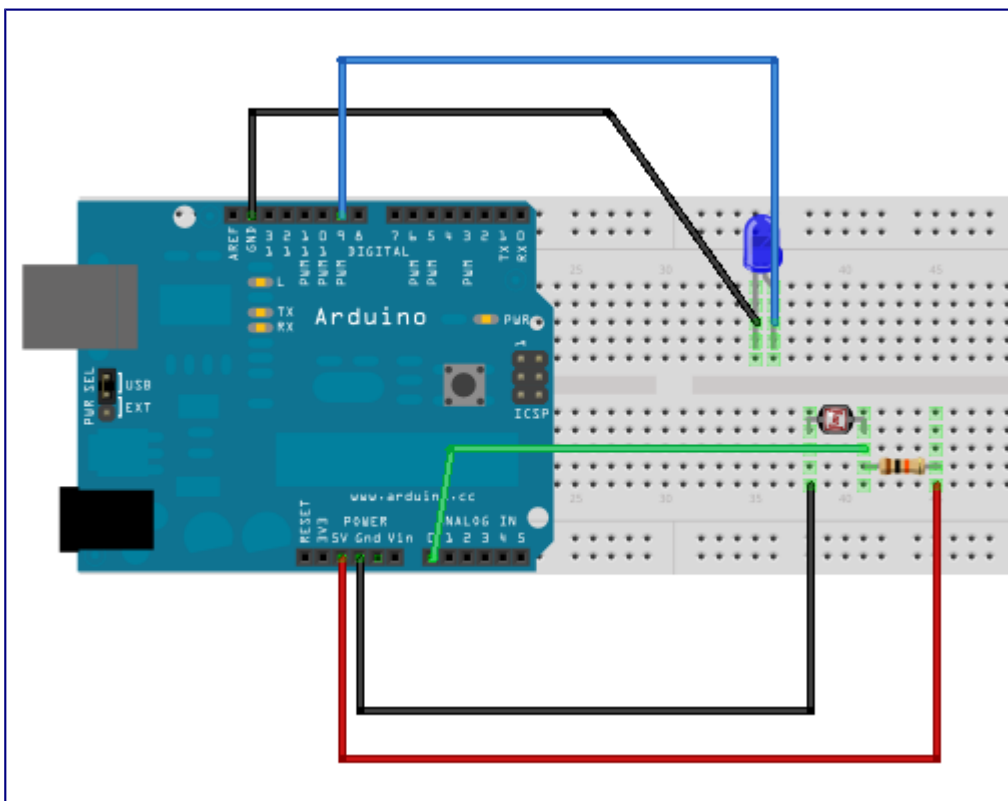
Faire clignoter une LED plus ou moins vite selon la luminosité ambiante.

Matériel

- une breadboard
- des fils
- une résistance de 10 k Ω
- une photo-résistance
- une LED

Montage

Réalisez ce montage :



Lisez un peu le code pour comprendre ce qu'il fait. Ensuite appuyez sur «**Téléverser**».

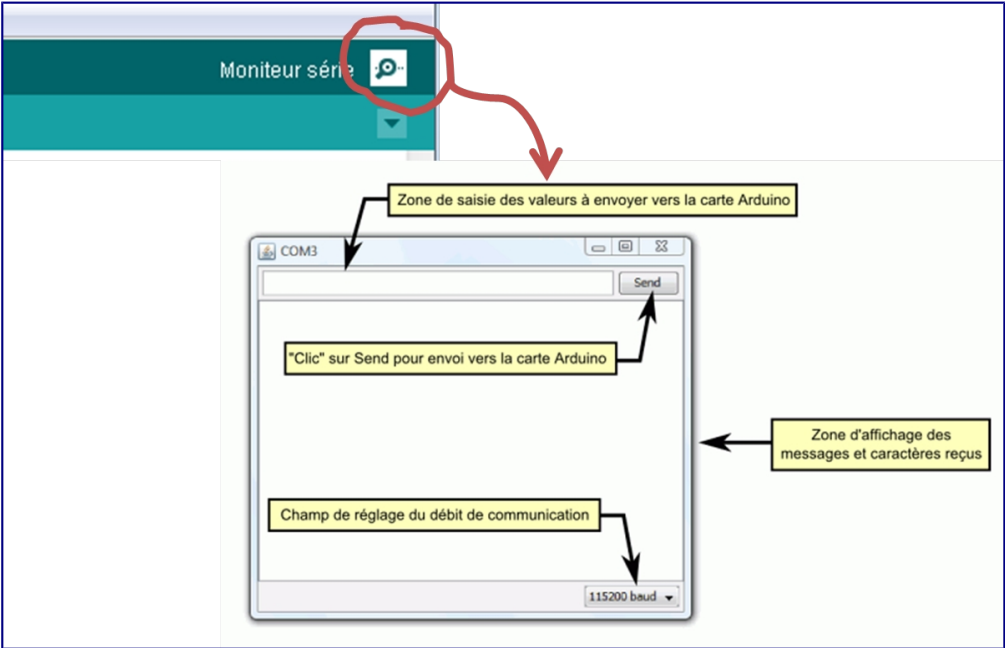
Essayez de cacher le capteur avec vos doigts : la LED devrait clignoter plus ou moins vite.

Programmation

```
/*  
  Lit la luminosité et  
  fait clignoter une LED à vitesse variable.  
*/  
  
// Numéro de la broche du capteur de luminosité  
const int luminosityPin = A0;  
// Numéro de la broche de la LED  
const int ledPin = 9;  
  
void setup() {  
  // Initialise la communication avec l'ordinateur  
  Serial.begin(9600);  
  // Indique que la broche de la LED est une sortie  
  pinMode(ledPin, OUTPUT);  
}  
  
void loop() {  
  // Lit la valeur du capteur de luminosité  
  int sensorValue = analogRead(luminosityPin);  
  // affiche la valeur du capteur dans le "moniteur" sur l'ordinateur  
  Serial.print(" Valeur du Capteur = " );  
  Serial.print(sensorValue);  
  
  // Convertit la valeur en durée d'attente pour la LED  
  // il sera nécessaire d'adapter les valeurs du "mapping" en  
  // fonction de la luminosité ambiante  
  int waitTime = map(sensorValue, 0, 1023, 20, 1000);  
  
  // Envoie le résultat vers l'ordinateur  
  Serial.print("\t Delai de clignotement (millisecondes) = ");  
  Serial.println(waitTime);  
  
  // Fait clignoter la LED au rythme de la luminosité  
  digitalWrite(ledPin, HIGH); // allume la LED  
  delay(waitTime); // attends la durée choisie  
  digitalWrite(ledPin, LOW); // éteinds la LED  
  delay(waitTime); // attends la durée choisie  
}
```

Astuces

- **Les broches A0 à A5 sont spéciales : elles permettent de lire une valeur analogique : non pas 0 ou 1 mais une valeur de 0 à 1023.**
- La photo-résistance est une résistance dont la valeur change en fonction de la luminosité reçue.
- Pour voir les données envoyées à l'ordinateur, appuyez sur la petite loupe en haut à droite de l'IDE Arduino. Les valeurs affichées vous permettront de calibrer votre fonction :



Objectif

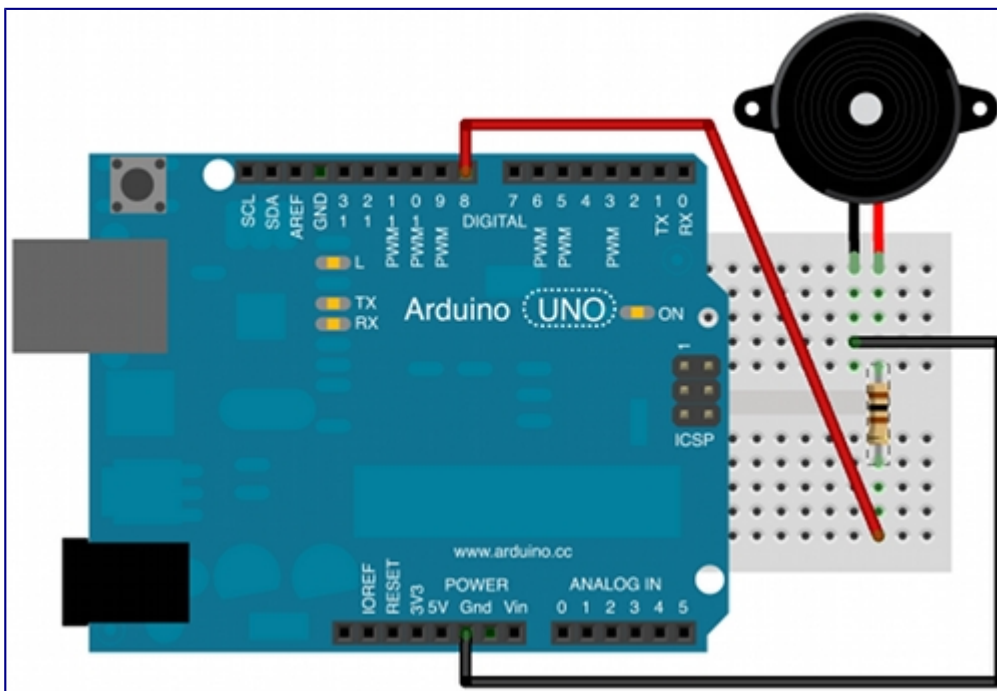
Jouer une mélodie avec un buzzer.

Matériel

- un buzzer
- une breadboard
- des fils
- une résistance de 100 Ω

Montage

Réalisez ce montage :



Programmation

Copiez-collez ce code dans l'IDE Arduino :

Vous trouvez aussi le code par le menu Fichier->Exemples->Basics->toneMelody

```
/*
  Plays a melody
  circuit: 8-ohm speaker on digital pin 8
  */
#include "pitches.h"
// notes in the melody:
int melody[] = {
  NOTE_C4, NOTE_G3,NOTE_G3, NOTE_A3, NOTE_G3,0, NOTE_B3, NOTE_C4};
// note durations: 4 = quarter note, 8 = eighth note, etc.:
int noteDurations[] = {
  4, 8, 8, 4,4,4,4,4 };
void setup() {
  // iterate over the notes of the melody:
  for (int thisNote = 0; thisNote < 8; thisNote++) {

    // to calculate the note duration, take one second
    // divided by the note type.
    //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
    int noteDuration = 1000/noteDurations[thisNote];
    tone(8, melody[thisNote],noteDuration);
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    // stop the tone playing:
    noTone(8);
  }
}
void loop() {
  // no need to repeat the melody.
}
```


Lisez un peu le code pour comprendre ce qu'il fait. Ensuite appuyez sur « **Téléverser** » .

Vous devriez entendre de la musique.

Astuces

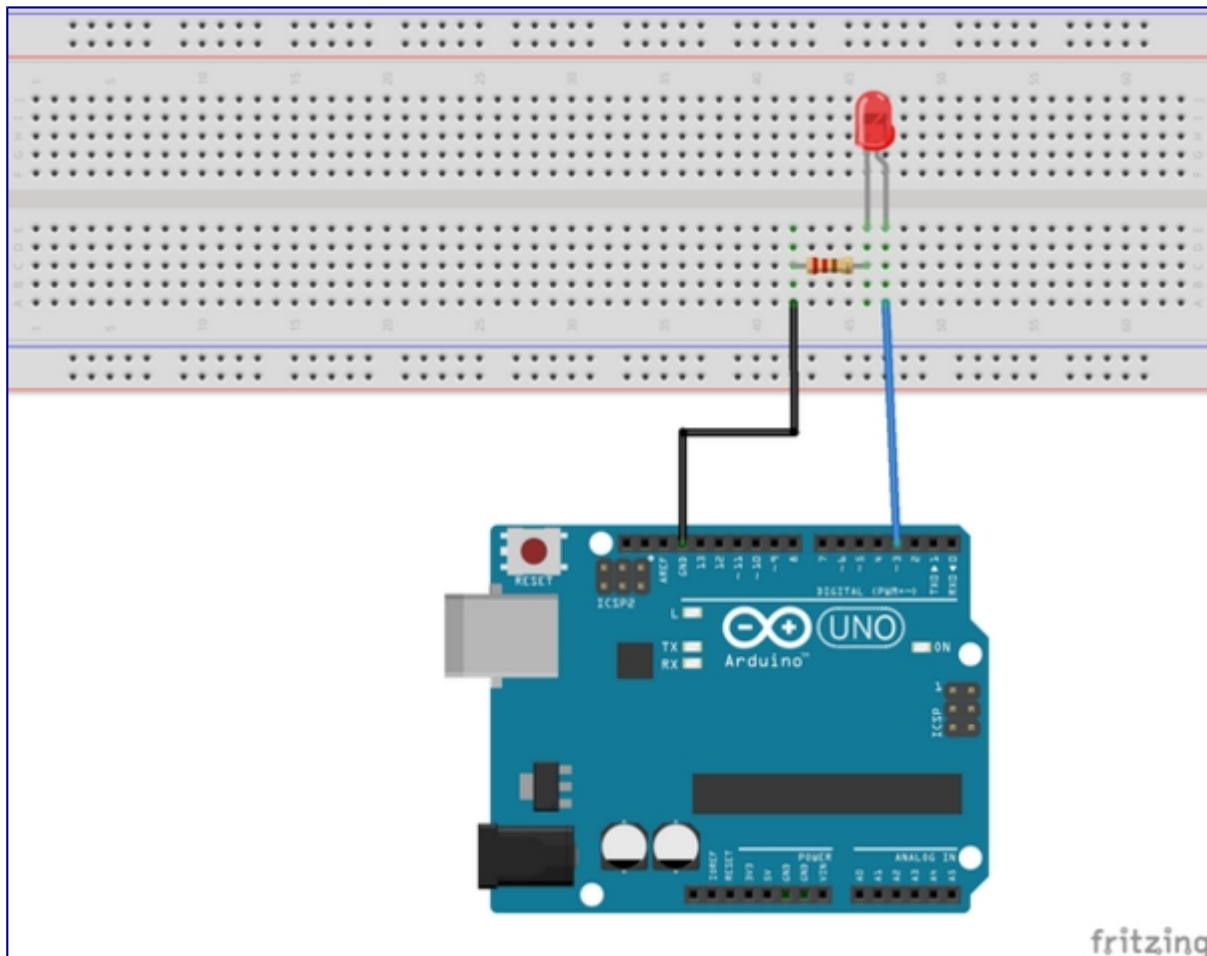
- **Le buzzer a un sens !**
- **La résistance sert à diminuer le volume.**
- La fonction *tone* permet de régler la fréquence d'un signal carré généré en sortie.

Modifier l'intensité lumineuse d'une LED

Matériel

- Arduino Uno
- Plaque d'essai (breadbord)
- LED
- Bouton Poussoir
- Résistance 1/4W 220Ω ou 330Ω peuvent également convenir
- Straps ou fils de liaison

Montage



Programme

Voici le programme qu'il faut éditer dans l'IDE Arduino, puis compiler et charger dans la carte Arduino

```
/*  
  
  Ce programme allume progressivement une LED branchée sur la broche 3  
  
*/  
  
int led = 3; // broche de la LED  
int i = 0; // variable de la boucle for  
  
// setup est déroulé une seule fois après la remise à zéro  
void setup() {  
  pinMode(led, OUTPUT); // la broche led (3) est initialisée en sortie  
}  
  
// loop est déroulé indéfiniment  
void loop() {  
  for (i=0; i<255;i++) // i varie de 1 à 255  
  {  
    analogWrite(led,i); // génère une impulsion sur la broche de largeur i => la luminosité augmente  
    delay(20); // pause de 20 ms entre chaque pas  
  }  
}
```

Informations :

broche: la broche utilisée pour "écrire" l'impulsion. Cette broche devra être une broche ayant la fonction PWM, Par exemple, sur la UNO, ce pourra être une des broches 3, 5, 6, 9, 10 ou 11.

- **valeur:** la largeur du "duty cycle" (proportion de l'onde carrée qui est au niveau HAUT) : entre 0 (0% HAUT donc toujours au niveau BAS) et 255 (100% HAUT donc toujours au niveau HAUT).

Il n'est pas nécessaire de faire appel à l'instruction pinMode() pour mettre la broche en sortie avant d'appeler la fonction analogWrite().