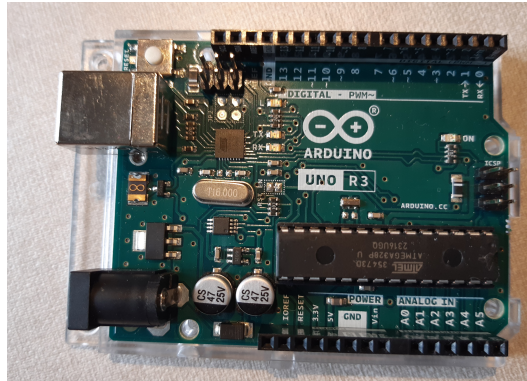
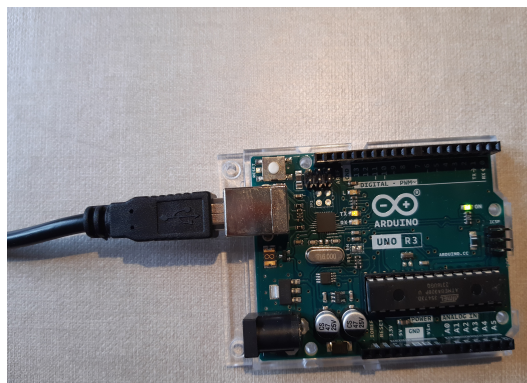


1 La carte arduino



La connexion de la carte à l'ordinateur:



2 Premier montage (lumière)

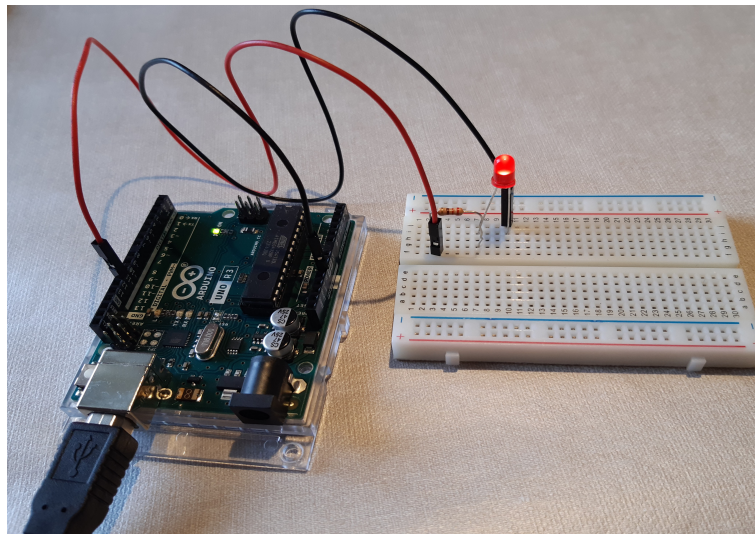
2.1 Les composants

- Une diode (rouge)
- Une résistance 220 ohm
- Des straps pour relier les composants à la carte Arduino.

La diode et la résistance sont mises en séries.

La borne (+) de la résistance, est reliée par un strap (en rouge sur la figure) à la broche digitale 11.

La borne (-) de la diode (patte courte de la diode) appelée aussi *cathode*) est reliée à la terre **GND** côté analogique.



Effectuez le branchement proposé ci-dessus.

2.2 le programme

Ouvrir un nouveau fichier *arduino* .Ce que l'on voit doit ressembler à :



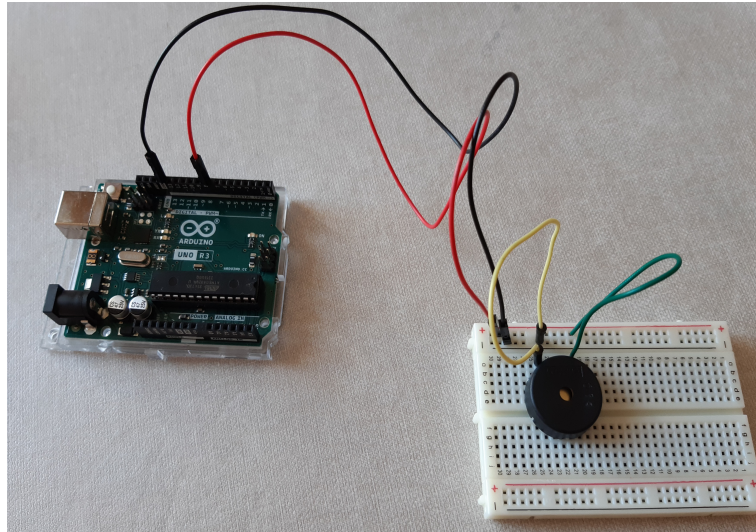
```
void setup() {
// code qui est exécuté une seule fois:

pinMode(11, OUTPUT);
}

void loop() {
// code destiné à tourner dans une boucle infinie:

digitalWrite(11,HIGH);
delay(1000);
digitalWrite(11,LOW);
delay(1000);
}
```

3 Deuxième montage (le son)



3.1 Les composants

- Un haut parleur 8 ohm.
- Des straps pour relier les composants à la carte Arduino.

Reliez une des bornes du buzzer au pin digital 8 et l'autre borne à la terre (GND).

3.2 le programme de la mélodie

```
#include "pitches.h"

// Définir la broche du haut-parleur
#define speakerPin 8

// Tableau des notes de la mélodie (do, sol, do, sol, do, ré, si, la, sol)

int notes[] = {
NOTE_C4, NOTE_G3, NOTE_C4, NOTE_G3, NOTE_C4, NOTE_D4, NOTE_B3, NOTE_A3, NOTE_G3
};

// Tableau des durées des notes (en millisecondes)
// La note de base est une croche (200 ms), sauf pour les deux dernières

int durations[] = {
200, 200, 200, 200, 200, 400, 400, 400, 800
//400 ,400 ,400, 400 ,400,800 ,800 ,800, 1600
};

// Nombre total de notes dans le morceau
int numberOfNotes = 9;

void setup() {

}

void loop() {
// Boucle à travers chaque note et sa durée
// Rien à initialiser dans le setup

for (int thisNote = 0; thisNote < numberOfNotes; thisNote++) {

// Joue la note sur la broche du haut-parleur pendant la durée spécifiée

tone(speakerPin, notes[thisNote], durations[thisNote]);

// Fait une pause entre chaque note pour ne pas les superposer

delay(durations[thisNote]);
noTone(speakerPin);

}

// Petite pause avant de rejouer le morceau

delay(3000);

noTone(speakerPin);

}
```