SESSION 04 · DÉBUTANT 🟢 - Buzzers passif et actif

🎯 Objectifs

- Apprendre à utiliser un buzzer actif et un buzzer passif

🧰 Matériel

Composant	Qté	Note
Arduino Uno + câble USB	1	
Breadboard	1	
Buzzer actif	1	
Buzzer passif	1	
Câbles Dupont	4	

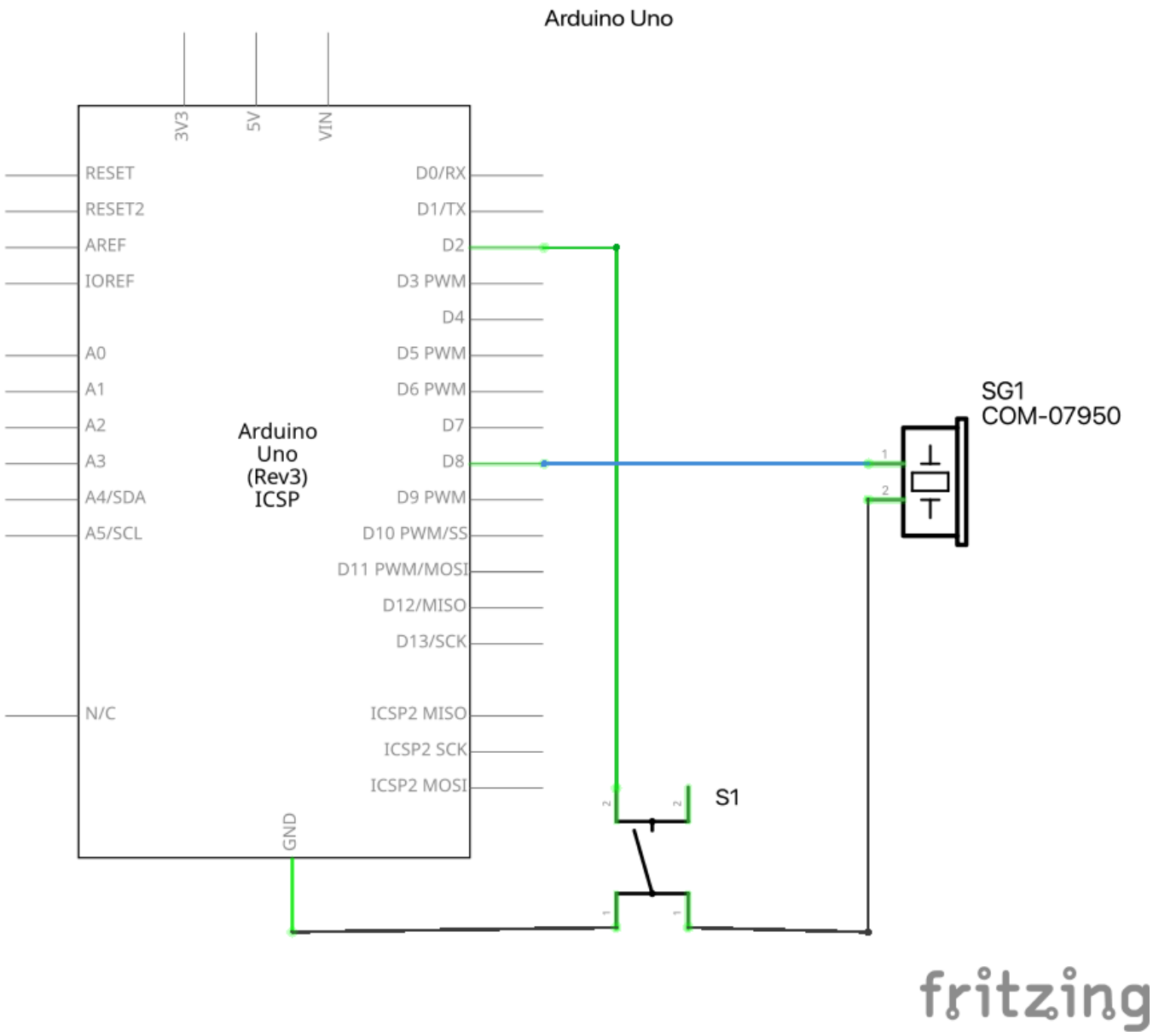
🧰 Montage 1 - Le buzzer actif



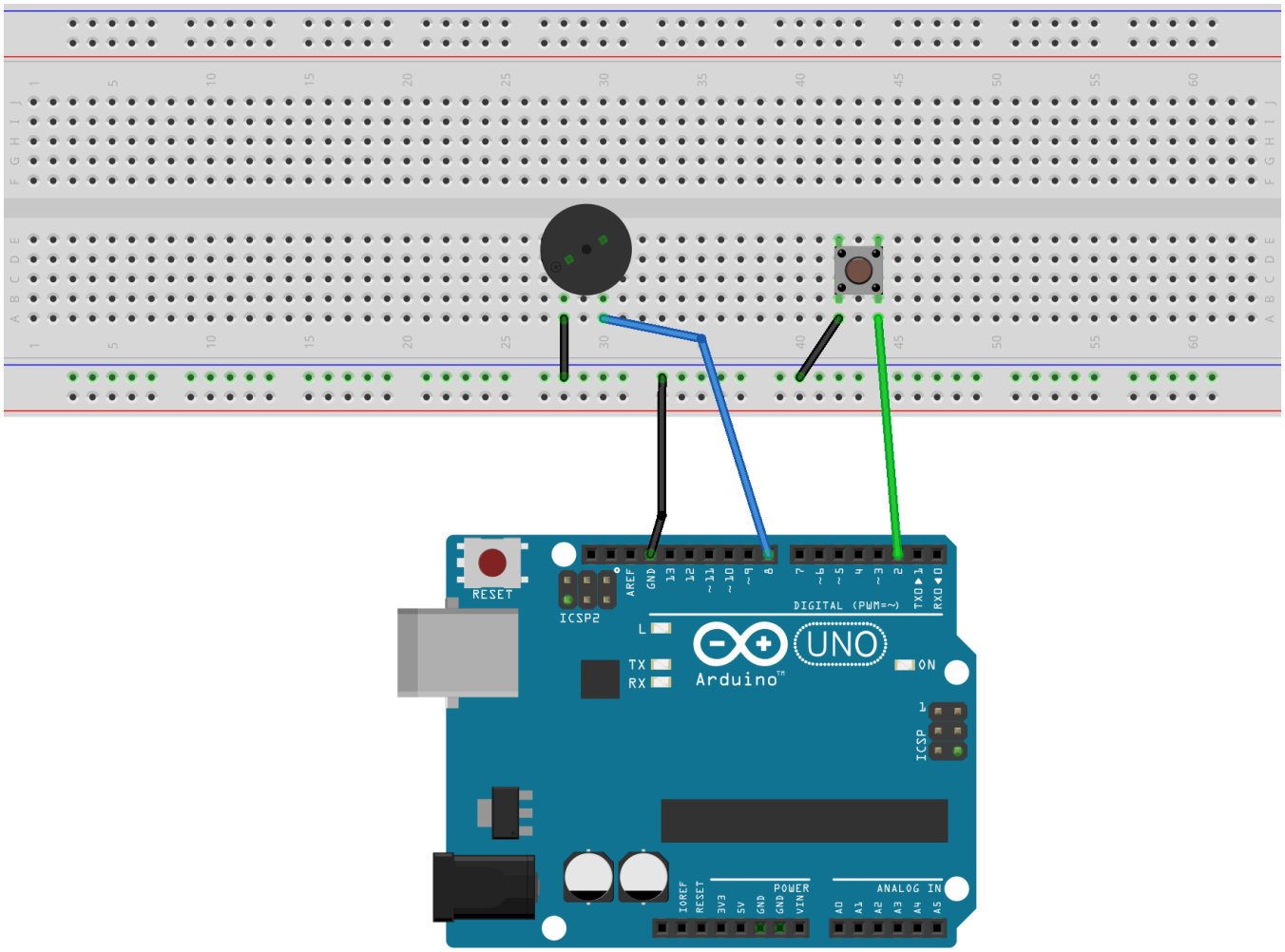
Un buzzer actif contient son propre oscillateur interne. Il suffit de lui appliquer une tension pour qu'il produise un **son à fréquence fixe**.

🔌 Schéma de principe

Dans ce montage, l'idée est d'activer le buzzer lorsque le bouton est pressé.


Arduino Uno
3V3
5V
VIN
RESET
RESET2
AREF
IOREF
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
ICSP2 MISO
ICSP2 SCK
ICSP2 MOSI
GND
Arduino
Uno
(Rev3)
ICSP
SG1
COM-07950
S1
fritzing

🔌 Schéma de câblage



⚡ Code buzzer actif

```
// Variables désignant les broches utilisées par le bouton et le buzzer
int  bouton = 2;
int  buzzer = 8;

void setup() {
  // Configuration de la broche du bouton en entrée avec résistance de rappel (par défaut:
  pinMode(bouton, INPUT_PULLUP);
  // Broche buzzer configurée en sortie
  pinMode(buzzer, OUTPUT);
}

void loop() {
  // Test sur la position du bouton
  if (digitalRead(bouton) == LOW) { // LOW = pressé!
    // Bouton pressé: entrée à LOW: le buzzer est alimenté en 5V
    digitalWrite(buzzer, HIGH);
  } else {
    // Bouton relaché: entrée à HIGH par défaut: le buzzer n'est pas alimenté
    digitalWrite(buzzer, LOW);
  }
}
```

🧰 Montage 2 - Le buzzer passif

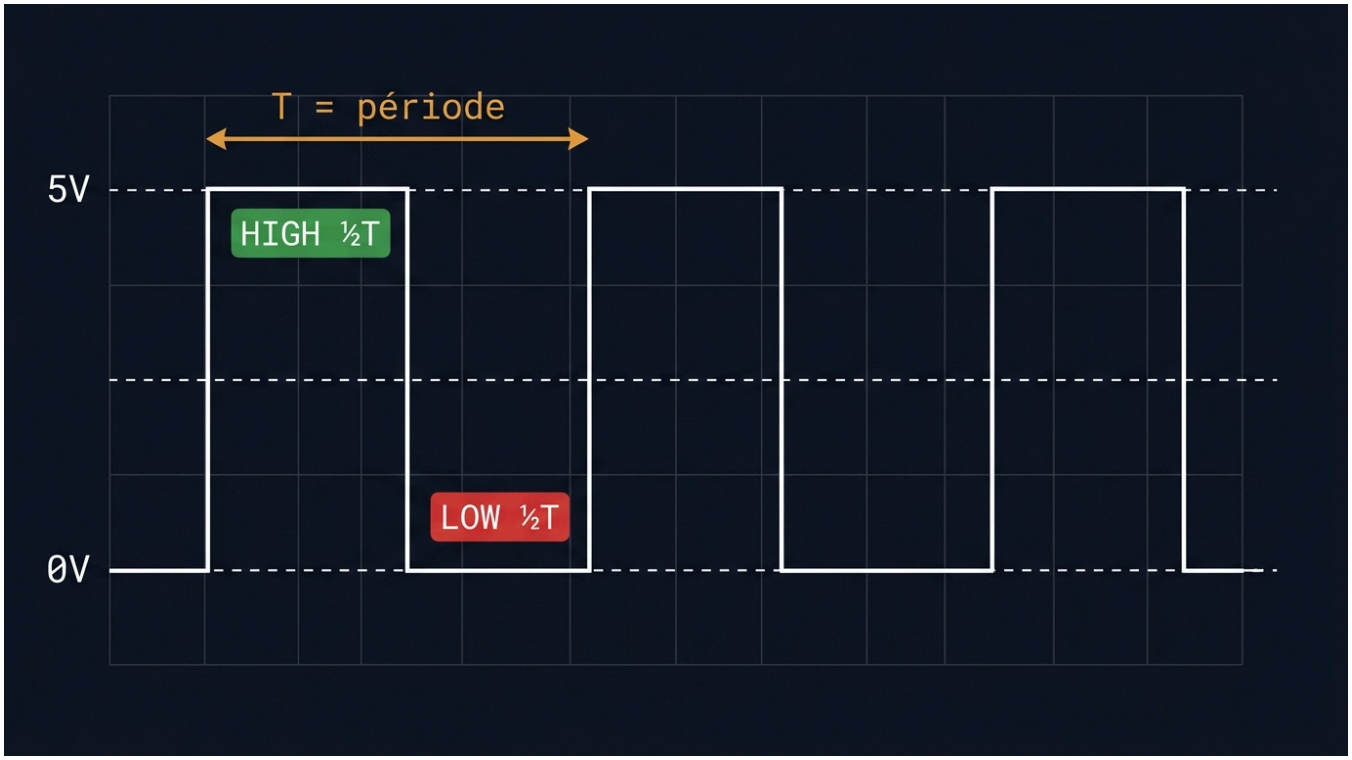


Un buzzer passif est juste un transducteur piézoélectrique **sans oscillateur**.

> 📍 **Comment fonctionnent les transducteurs piézoélectriques ?**
> Certains matériaux (souvent des cristaux ou des céramiques spéciales) génèrent une charge électrique lorsqu'on leur applique une contrainte mécanique. Le processus fonctionne également en sens inverse pour faire vibrer une membrane (son) à partir d'une alimentation électrique

C'est le microcontrôleur Arduino qui doit générer le signal à la fréquence souhaitée.

Pour un signal de fréquence 440 Hz (La4), la période T = 1/F = 2,27 ms


T = période
5V
HIGH ½T
LOW ½T
0V

Code simple

```
/* Cette fois, la broche du buzzer est déclarée par une MACRO
 Le texte BUZZER sera partout remplacé par 8 dans tout le code avant la phase de compilati
*/
#define BUZZER 8

int freq = 440;          // Frequence en Hz
long T = 1000000/freq;  // Période en micro-seconde


void setup() {
  // Configuration de la broche connectée au buzzer
  pinMode(BUZZER, OUTPUT);
}

void loop() {

  digitalWrite(BUZZER, HIGH);   // Niveau 5V
  delayMicroseconds(T/2);       // Attente 1/2 période
  digitalWrite(BUZZER, LOW);    // Niveau 0V
  delayMicroseconds(T/2);       // Attente 1/2 période

}
```

Plus de flexibilité avec une fonction

```
#define BUZZER 8

/* Fonction jouerNote()
  Joue une note à la fréquence donnée pendant 'duree' ms
  Paramètres d'entrée:
    freq: (entier) fréquence à jouer
    duree: (entier) durée de la note
*/
void jouerNote(int freq, int duree) {

  long T = 1000000/freq;            // Période en micro-seconde

  // Calcul de l'heure de fin de la période
  long fin = millis() + duree;      // millis() renvoie le nombre de millisecondes depuis

  // Pour toute la durée souhaitée:
  while (millis() < fin) {
    digitalWrite(BUZZER, HIGH);   // Niveau haut
    delayMicroseconds(T/2);       // Attente demi-période
    digitalWrite(BUZZER, LOW);    // Niveau bas
    delayMicroseconds(T/2);       // Attente demi-période
  }
}

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

void loop() {
  jouerNote(440, 500); // La4 pendant 500 ms
  jouerNote(494, 500); // Si4 pendant 500 ms
  jouerNote(523, 500); // Do5 pendant 500 ms

  delay(1000);    // Attente 1 seconde
}
```

Encore plus facile avec la fonction tone()

https://docs.arduino.cc/language-reference/en/functions/advanced-io/tone/