

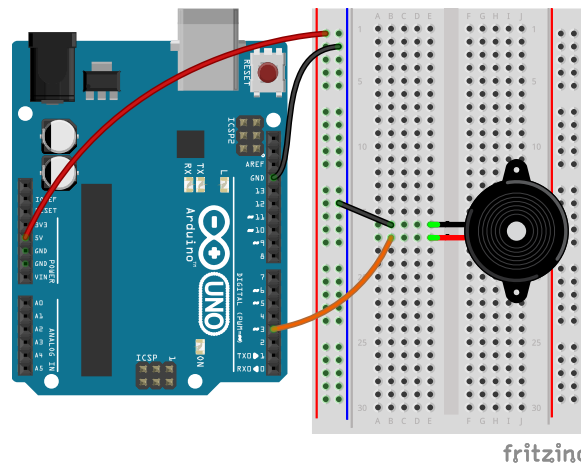
BUZZER

Montage

Sortez votre matériel et câblez le circuit suivant :

Matériel nécessaire :

- Arduino
- Plaque d'essai
- 1 buzzer

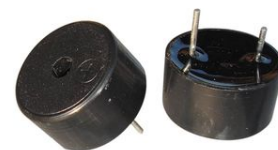


fritzing

Le buzzer (ou bipeur...)

Le buzzer possède deux broches : une broche + et une broche -. En général, la broche + est repérée sur le dessus ou le dessous du buzzer.

Un **buzzer** est un composant électromécanique. Quand un courant le parcourt, il émet un « clic ». Quand on émet des milliers de clic par seconde, on crée une note d'une certaine *fréquence*.



Le langage de l'Arduino possède l'instruction **tone**(broche, fréquence, durée) qui émet des clics à une certaine fréquence (en Hz) pendant un certain temps (en ms). Ci-dessous figure les correspondances entre les notes de l'octave 3 en français et en anglais, et leur fréquence :

Note	Do	Ré	Mi	Fa	Sol	La	Si	DO
	c	d	e	f	g	a	b	C
Fréquence (Hz)	262	294	330	349	392	440	494	523

Vous trouverez les fréquences sur plusieurs octaves, ici par exemple :

fr.wikipedia.org/wiki/Note_de_musique#Fr.C3.A9quence

Programme

Branchez l'Arduino au câble USB et *téléversez* le *croquis* ci-dessous.



CODE ARDUINO

```
// Croquis P8-Buzzer.ino
//----- CONSTANTES -----
#define Buzzer 3    // BUZZER sur la ligne d'E/S 3
//----- VARIABLES -----
int duree = 500;    // durée des notes
int pause = duree * 1.30; // Pause entre les notes
// (30 % de plus que la durée des notes)
//----- PROCEDURE D'INITIALISATION -----
void setup() {
    pinMode(Buzzer, OUTPUT); // Ligne Buzzer en sortie
}
//----- BOUCLE PRINCIPALE -----
void loop(){
    tone(Buzzer, 262, duree);
    delay(pause);
```

CODE ARDUINO

```
    tone(Buzzer, 294, duree);
    delay(pause);
    tone(Buzzer, 330, duree);
    delay(pause);
    tone(Buzzer, 349, duree);
    delay(pause);
    tone(Buzzer, 392, duree);
    delay(pause);
    tone(Buzzer, 440, duree);
    delay(pause);
    tone(Buzzer, 494, duree);
    delay(pause);
    tone(Buzzer, 523, duree);
    delay(1000);
}
```

Vous devez entendre la gamme...

Il est possible d'améliorer ce programme à l'aide d'une boucle, car ce sont *presque* les mêmes instructions qui se répètent dans la boucle principale.

Le type char

Nous allons utiliser les notes en anglais, qui sont notées avec un seul caractère. En C, et donc en langage Arduino, les caractères ont leur propre type : le type **char** qui prend un octet de mémoire.

Un caractère doit être écrit entre guillemets simples, tandis que plusieurs caractères (chaîne de caractères, aussi appelé string) doivent être écrits entre guillemets doubles.

CODE ARDUINO

```
char monCaractere = 'A';
char maChaine = "coucou";
```

En fait, le type **char** est un type *signé* pour des nombres allant de -128 à 127 (le type **byte** lui aussi codé sur un octet n'est pas signé et utilise les nombres allant de 0 à 255).

Définir une fonction

Le langage de l'Arduino permet de définir ses propres fonctions. Une fonction renvoie une valeur d'un certain type. Par exemple, on peut définir une fonction qui renvoie la somme de deux nombres. Le nombre renvoyé sera ici de type **int**.

CODE ARDUINO

```
int add(int terme1, int terme2) {
    int somme;
    somme = terme1 + terme2;
    return(somme);
}
```

Dans un programme, l'instruction `add(24,6)` vaudra 30.



Programme

Nous allons définir une fonction qui, à une note, va associer sa fréquence (voir page 1).

CODE ARDUINO

```
// Croquis P8-Buzzer-Fonction.ino
//----- CONSTANTES -----
#define Buzzer 3    // BUZZER raccordé à la ligne d'E/S 3
//----- VARIABLES -----
int duree = 500;    // durée des notes
int pause = duree * 1.30; // Pause entre les notes
char gamme[] = "cdefgabC"; // la gamme
//----- PROCEDURE D'INITIALISATION -----
void setup() {
    pinMode(Buzzer, OUTPUT); // Ligne Buzzer en sortie
}
//----- BOUCLE PRINCIPALE -----
void loop(){
    for (int i = 0; i < sizeof(gamme)-1; i++){
        tone(Buzzer, donneFrequence(gamme[i]), duree);
        delay(pause);
    }
    delay(1000);
}
//----- DEFINITION DE LA FONCTION -----
int donneFrequence(char note) { // on recherche la fréquence de "note"
    char noms[] = {'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' }; // nom des notes
    int frequences[] = {262, 294, 330, 349, 392, 440, 494, 523}; // fréquences associées
    // Recherche de la fréquence associée à "note"
    for (int i = 0; i < 8; i++) { // 8 car il y a 8 notes
        if (noms[i] == note) { // on parcourt les noms jusqu'à trouver "note"
            return(frequences[i]); // on a trouvé, on retourne la fréquence
        }
    }
}
}
```

Applications

1. Écrire (ou modifier) un croquis afin de tester quelques fréquences différentes.
2. Réaliser un montage et écrire un programme qui fera bipper le buzzer selon la valeur d'un potentiomètre.
3. Pour jouer des mélodies, voir cette page :

http://www.wikidebrouillard.org/index.php/Lecteur_de_mélodie

Dans le monde réel...

Tous les appareils électroniques qui émettent des *bips* utilisent un buzzer, comme les ordinateurs par exemple.

