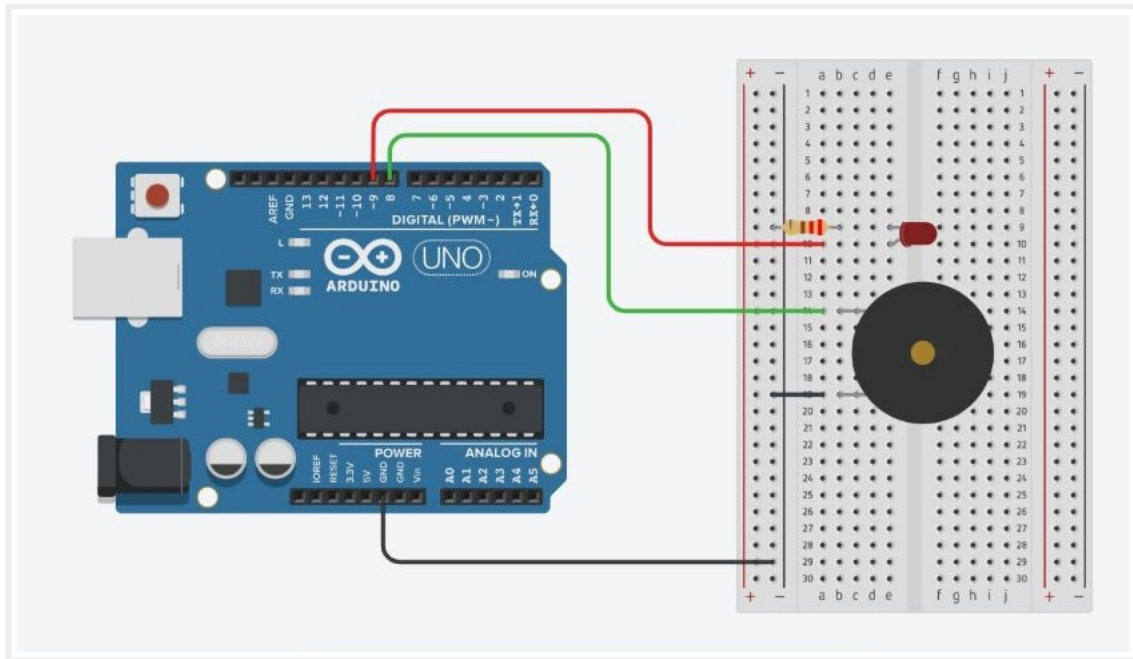


Beep

(Faire clignoter une DEL et produire un "beep" synchrone)



. Liste des composants :

- . 1 DEL rouge
- . 1 résistance de 220 Ω (résistance de protection de la DEL)
- . 1 buzzer
- . 1 plaque d'essai
- . Fils de connexion

. Objectif

Dans cette activité, le programme des activités d'apprentissage qui permet de faire clignoter une DEL est modifié pour commander la production d'un signal sonore ("un beep"), émis par un buzzer ou un petit haut-parleur, synchrone avec le clignotement de la diode.

Le programme

Le programme de l'activité pourra être modifié pour voir l'influence des variables (durée d'allumage et d'extinction, fréquence en Hz de l'onde sonore) :

```
Beep

// Déclaration des constantes et variables

const int PinLED = 9;
const int PinTone = 8;
const int FreqTone = 440;
const int TimeSleep1 = 500;
const int TimeSleep2 = 500;

// Initialisation des entrées et sorties

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

// Fonction principale en boucle

void loop()
{
  digitalWrite(PinLED, HIGH);
  tone(PinTone, FreqTone);
  delay(TimeSleep1);
  digitalWrite(PinLED, LOW);
  noTone(PinTone);
  delay(TimeSleep2);
}
```

Déroulement du programme :

– Déclaration des constantes et variables :

. **const int PinLed = 9** (constante nombre entier correspondant au n° de la broche sur laquelle la DEL rouge est connectée)

. **const int PinTone = 8** (constante nombre entier correspondant au n° de la broche sur laquelle le buzzer est connecté)

. **const int FreqTone = 440** (constante nombre entier correspondant à la fréquence en Hz de l'onde sonore)

. **const int TimeSleep1 = 500** (constante nombre entier correspondant à la durée d'allumage de la DEL en ms)

. **const int TimeSleep2 = 500** (constante nombre entier correspondant à la durée d'extinction de la DEL en ms)

– Initialisation des entrées et sorties :

. La broche de la DEL est initialisée en sortie digitale. Des données seront donc envoyés depuis le microcontrôleur vers cette broche :

pinMode(PinLed, OUTPUT)

– Fonction principale en boucle :

. Niveau haut sur la broche de la Del : **digitalWrite(PinLed, HIGH)**

. Emission de l'onde sonore en continu : **tone(PinTone,FreqTone)**

. Attente pendant TimeSleep1 ms : **delay(TimeSleep1)**

. Niveau bas sur la broche de la Del : **digitalWrite(PinLed, Low)**

. Arrêt de l'émission sonore : **notone(PinTone)**

. Attente pendant TimeSleep2 ms : **delay(TimeSleep2)**

Rappel - Gestion du son en langage Arduino

. La fonction tone() :

Le signal électrique appliqué par l'Arduino sur une de ses sorties digitales ou analogiques, sur laquelle est connecté le piezo ou le haut-parleur et qui va créer l'onde sonore, est réalisé avec la fonction **tone()**.

Cette fonction génère une onde carrée (onde symétrique avec « duty cycle » (niveau haut/période) à 50%) à la fréquence spécifiée en Hertz (Hz) sur une broche. La durée peut être précisée, sinon l'impulsion continue jusqu'à l'appel de l'instruction **noTone()**.

Une seule note peut être produite à la fois. Si une note est déjà jouée sur une autre broche, l'appel de la fonction **tone()** n'aura aucun effet (tant qu'une instruction **noTone()** n'aura pas eu lieu).

Si la note est jouée sur la même broche, l'appel de la fonction **tone()** modifiera la fréquence jouée sur cette broche.

Enfin, l'utilisation de **tone()** rend impossible l'utilisation des broches **D3** et **D11** en PWM avec **analogWrite()**.

. Syntaxe :

tone(broche, fréquence)

tone(broche, fréquence, durée)

. Paramètres :

Broche : la broche sur laquelle la note est générée

Fréquence : la fréquence de la note produite, en hertz (Hz)

Durée : la durée de la note en millisecondes (optionnel)

. La fonction noTone() :

La fonction **noTone()** stoppe la génération d'impulsion produite par l'instruction **tone()**. Elle n'a aucun effet si aucune impulsion n'a été générée.

. Syntaxe :

noTone(broche)

. Paramètres :

broche: la broche sur laquelle il faut stopper la note.