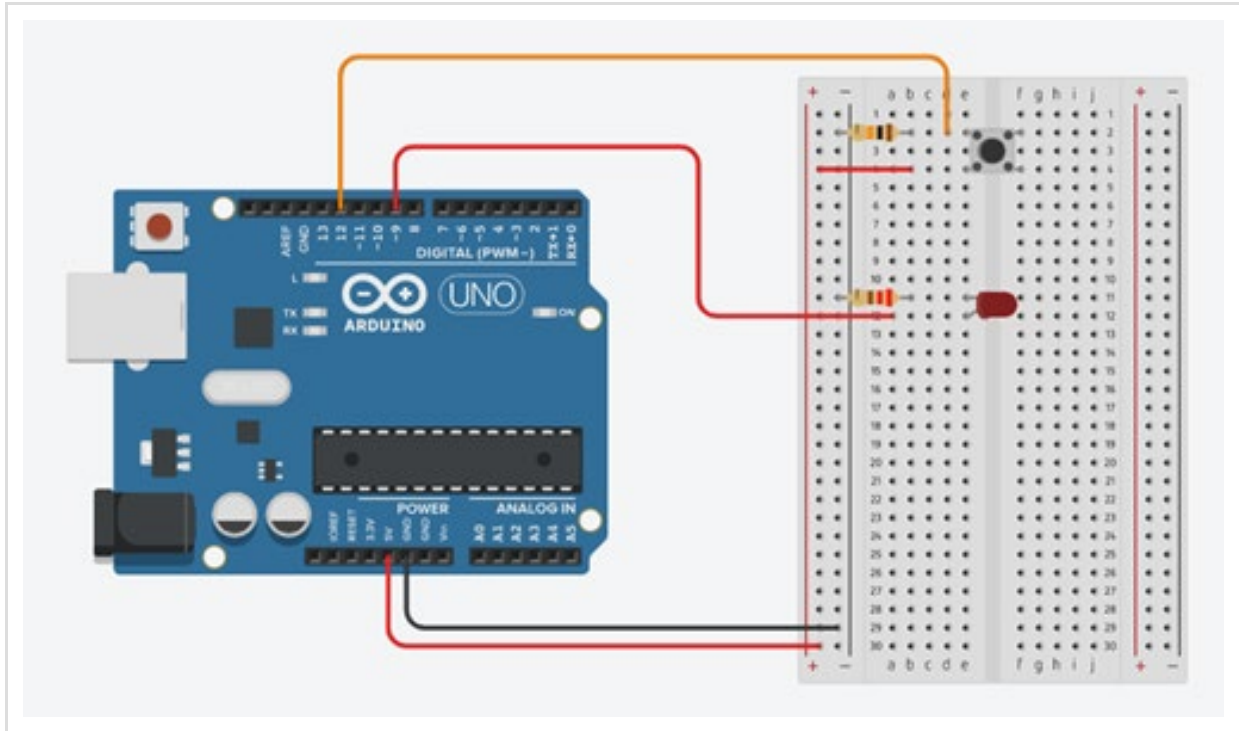


# *Bouton-poussoir (Allumer une DEL)*



## Liste des composants :

- . 1 DEL rouge
- . 1 résistance de  $220\ \Omega$  (résistance de protection de la DEL)
- . 1 bouton poussoir
- . 1 résistance de  $10\ k\Omega$  (résistance du circuit du bouton poussoir)
- . 1 plaque d'essai
- . Fils de connexion

## Objectif

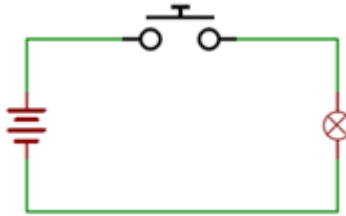
Dans cette activité, la DEL rouge s'allume en appuyant sur le bouton poussoir et s'éteint si on le relâche. L'objectif est de se familiariser avec les entrées numériques de l'Arduino.

## . Le bouton poussoir normalement ouvert

Un bouton poussoir normalement ouvert est un dispositif mécanique doté de 4 broches et d'une lamelle métallique qui met en contact toutes les broches lorsqu'on appui sur la tête du bouton. Un ressort de rappel ramène la tête du bouton lorsqu'il est relâché.



Un bouton poussoir normalement ouvert n'est jamais qu'un fil qui est connecté au circuit électrique ou non selon sa position :

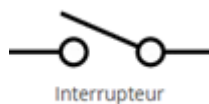
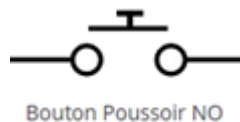


. **Relâché** : le courant ne passe pas dans le circuit électrique, le circuit est "ouvert" .

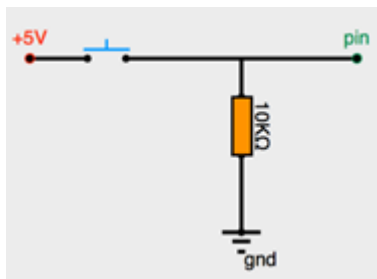
. **Appuyé** : le courant passe, on dit que le circuit est "fermé".

Les mêmes effets peuvent être produits avec un interrupteur sauf que circuit reste fermé ou ouvert tant que la position de l'interrupteur n'a pas été changé.

Le bouton poussoir et l'interrupteur ne possèdent pas le même symbole pour les schémas électroniques :



Avec un Arduino, il est principalement utilisé pour envoyer une « impulsion de commande » avec ce circuit électrique :



Quand le bouton poussoir est appuyé, le potentiel de sa broche connectée à la broche « **pin** » de l'Arduino passe à 5 V (le circuit est fermé) et quand il est relâché, celui-ci passe à 0 V (le circuit est ouvert).

Avec notre circuit, on pourra alors demander à l'Arduino, d'allumer ou d'éteindre la DEL connectée sur la broche 9 en fonction de la valeur du potentiel de la broche 12 de l'Arduino.

Avec le bouton poussoir, nous allons donner, à l'Arduino, l'ordre d'effectuer une action. D'où le terme « impulsion de commande ».

### Attention :

**Il est impératif d'utiliser une résistance de 10 kΩ en série avec le bouton poussoir dans un montage « impulsion de commande ».**

**Ainsi, le courant dans le circuit est très faible ( $I = U/R = 0,5 \text{ mA}$ ) quand le circuit est fermé (bouton poussoir appuyé), et il n'y a pas de risque pour l'Arduino.**

## . Gestion des entrées numériques

En langage Arduino, pour lire l'état logique d'une entrée numérique, on utilise la fonction :

### **digitalRead()**

Lit l'état (= le niveau logique) d'une broche initialisée en entrée numérique, et renvoie la valeur HIGH (HAUT en anglais) ou LOW (BAS en anglais).

### . Syntaxe :

**digitalRead(broche)**

### . Paramètre :

broche: le numéro de la broche de la carte Arduino

### . Valeur retournée :

renvoie la valeur HIGH (1) ou LOW (0)

## . Le programme

Dans [le programme de cette activité](#), on demande à l'Arduino d'interroger l'état logique (**niveau haut ou bas**) de la broche du bouton poussoir qui a été déclaré comme une entrée numérique, on peut alors savoir si celui-ci est appuyé ou pas et donc lui donner l'ordre d'allumer ou d'éteindre la DEL.

```
Bouton_poussoir
// Déclaration des constantes et variables

const int PinLED = 9;
const int PinButton = 12;

int ValButton = 0;

// Initialisation des entrées et sorties

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

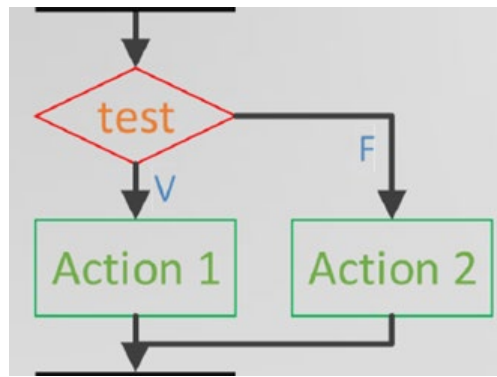
// Fonction principale en boucle

void loop() {
  ValButton = digitalRead(PinButton);
  if (ValButton == HIGH) {
    digitalWrite(PinLED, HIGH);
  }
  else {
    digitalWrite(PinLED, LOW);
  }
}
```

Rappel sur la programmation d'une condition if – else:

### Condition IF (Test)

```
if (test)
{
  // Action 1;
}
else
{
  // Action 2;
}
```



### 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 PinButton = 12** (constante nombre entier correspondant à la durée d'allumage de la Del en ms)
- . **int ValButton = 0** (variable nombre entier pour stocker la valeur de la broche du bouton poussoir)

– 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)**

- . La broche du bouton poussoir est initialisée en entrée digitale. Des données seront donc envoyés depuis cette broche vers le microcontrôleur :

**pinMode(PinButton, INPUT)**

– Fonction principale en boucle :

