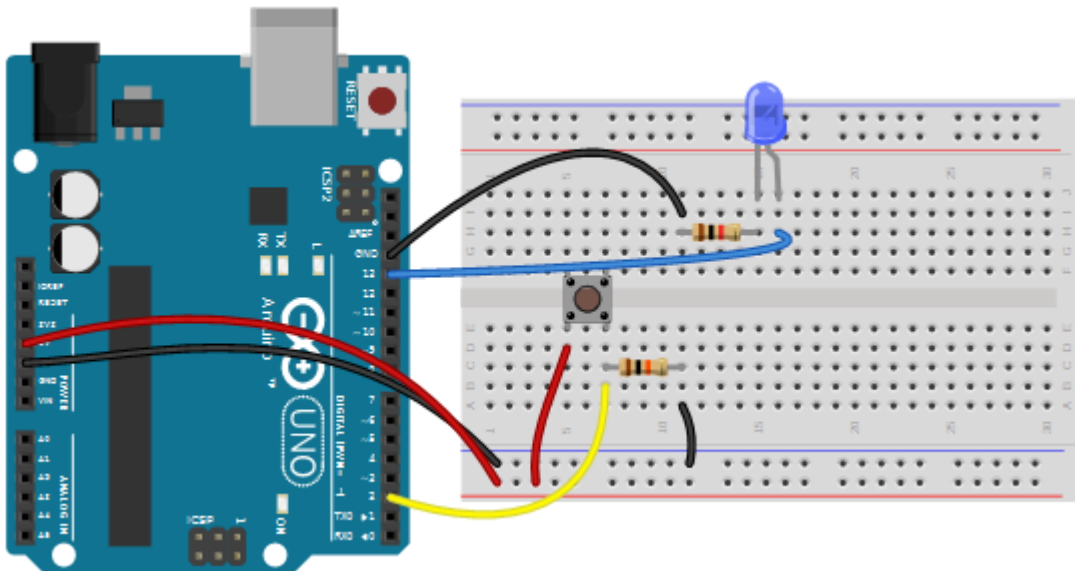


Capteur numérique

1. Montage



Pour le réaliser, vous aurez besoin de :

- Un Arduino
- Un câble USB
- Deux résistances de 1KΩ
- Des fils de prototypage
- Une platine de prototypage
- Un bouton poussoir
- Une LED de votre couleur préférée

2. Code

Cette suite d'instructions va allumer une LED branchée sur la broche 13 lorsque le bouton branché sur la broche 2 est appuyé. Lorsque vous utilisez le logiciel Arduino, le code peut être trouvé en cliquant sur *Fichier* → *Exemples* → *02.Digital* → *Button*.

```
/*
  Allume une LED branchée sur la broche 13 lorsque le bouton
  branché sur la broche 2 est appuyé.
  */

// Initialisation des constantes :
const int buttonPin = 2;    // Numéro de la broche à laquelle est connecté le
bouton poussoir
const int ledPin = 13;     // Numéro de la broche à laquelle est connectée la
LED

// le code dans cette fonction est exécuté une fois au début
void setup()
{
```

```
// indique que la broche ledPin est une sortie :
pinMode(ledPin, OUTPUT);
// indique que la broche buttonPin est une entrée :
pinMode(buttonPin, INPUT);
}

// le code dans cette fonction est exécuté en boucle
void loop()
{
  // lit l'état du bouton et stocke le résultat
  // dans buttonState :
  const int buttonState = digitalRead(buttonPin);

  // Si buttonState est à 5V (HIGH→bouton appuyé)
  if (buttonState == HIGH)
    // on allume la LED
    digitalWrite(ledPin, HIGH);
  else
    // sinon on éteint
    digitalWrite(ledPin, LOW);
}
```

Remarques : Copiez-collez ce code dans le simulateur pour ne pas avoir à tout retaper. Saviez vous que vous pouvez accéder à la documentation d'une fonction en cliquant avec le bouton droit sur celle-ci puis en cliquant sur *Trouvez dans la référence*.

3. Instructions

Voici une description des nouvelles fonctions utilisées (n'hésitez pas à cliquer sur les liens ci-dessous afin d'arriver sur [la référence Arduino](#)).

- **Déclaration d'une constante :** comme pour une [variable](#), on vient avec cette ligne stocker la valeur à droite du signe égal dans `led`.

```
const int led = 13;
```

Le mot clé `const` indique que l'on ne souhaite pas que la valeur de `led` puisse être modifiée dans le programme.

Les nouvelles instructions :

- `digitalRead` lit l'état d'une broche et renvoie la valeur `HIGH` si la broche est à la tension de l'alimentation ou `LOW` si la broche est à 0V.

```
digitalRead(buttonPin);
```

La valeur de retour de `digitalRead` peut être stockée dans une variable comme ceci :

```
buttonState = digitalRead(buttonPin);
```

- **if** permet de tester si une expression située entre parenthèse est vraie. Dans *Button*, nous cherchons à savoir si le bouton est appuyé, nous allons donc comparer `buttonState` à `HIGH` comme ceci :

```
if(buttonState == HIGH)
```

- **else** : le bloc situé après cet mot clé viendra être exécuté si le test précédent échoue. Dans *Button*, si le bouton **n'est pas** appuyé alors on viendra éteindre la LED.

4. Références

- [Référence mini du langage Arduino](#) par Xavier Hinault
- [Référence officielle du langage Arduino \(anglais\)](#) par l'équipe d'Arduino