

BOUTON POUSSOIR

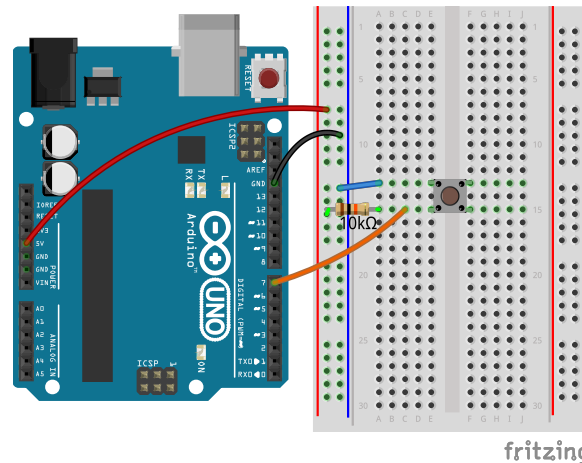
Montage

Sortez votre matériel et câblez le circuit suivant :

Matériel nécessaire :

- Arduino
- Plaque d'essai
- 1 bouton poussoir (touche)
- 1 résistance de 10 kΩ
- 4 câbles

Du matériel supplémentaire sera nécessaire pour les applications.



Programme

Nous allons ici utiliser le *moniteur série* pour lire sur l'ordinateur l'état du bouton poussoir.

- **Serial.begin** définit la vitesse de communication. Elle se mesure en bits par seconde. **La même vitesse doit être utilisée dans le moniteur série !**
- **Serial.print** permet d'écrire dans le moniteur. **Serial.println** rajoute en plus un retour à la ligne. Avec des guillemets, le texte est affiché tel quel, **sans guillemets, c'est la valeur de la variable qui est affichée.**
- Cette fois ci, il faut savoir si du courant passe, nous allons mettre la broche Touche en *entrée* : c'est le rôle de **pinMode(Touche, INPUT)**, puis nous allons *lire* l'état de la broche : c'est ce que fait **digitalRead(Touche)**. La valeur est mise dans la variable Etat.

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

Appuyez sur l'icône permettant d'afficher le *moniteur série* (en haut à droite).

Que remarquez-vous ?

Maintenant **enlevez la résistance**. Que remarquez-vous ?

CODE ARDUINO

```
// Croquis P4-Touche.ino
//----- CONSTANCE -----
#define Touche 7 // TOUCHE sur la ligne d'E/S 7
//----- VARIABLE -----
int Etat = 0; // Etat logique de la TOUCHE
//----- PROCEDURE D'INITIALISATION -----
void setup() {
  // Vitesse de communication (port série)
  Serial.begin(9600);
  // Ligne de la TOUCHE en entrée
  pinMode(Touche, INPUT);
}
//----- BOUCLE PRINCIPALE -----
void loop() {
  // Lecture de l'état de la TOUCHE
  Etat = digitalRead(Touche);
  // Affichage sur le moniteur série
  Serial.print("Etat de la touche : ");
  Serial.println(Etat);
}
```

Bouton poussoir, ou « touche »

Un bouton poussoir est un composant qui laisse passer le courant lorsqu'il est pressé.

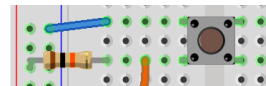
Avec le montage réalisé, lorsque le bouton est appuyé, l'entrée 7 est à 0 V (puisque reliée à GND) et est donc



lue comme étant à l'état bas (**LOW** ou 0). Mais lorsqu'il n'est pas appuyé, l'entrée n'est pas définie, elle est *en l'air* et elle peut être lue comme étant aussi bien à l'état haut (**HIGH** ou 1) qu'à l'état bas.

Pour résoudre ce problème, il faut connecter une résistance de tirage vers le haut (*pullup*) d'au moins 4,7 kΩ entre l'entrée et l'alimentation. Nous prendrons une résistance de 10 kΩ :

Lorsque le bouton est appuyé, l'entrée est à 0 V et un courant circule dans la résistance. Lorsque le bouton n'est pas appuyé, l'entrée est à 5 V et il ne circule aucun courant dans la résistance.



Remettez la résistance et vérifiez que tout fonctionne parfaitement avec le moniteur série.

Maintenant nous allons afficher sur le moniteur « Le bouton est appuyé » ou « Le bouton n'est pas appuyé » suivant le cas. Nous avons donc besoin de faire un test et d'agir en conséquence...

Instructions conditionnelles « if ... else »

Notre programme doit donc procéder ainsi :

- Si le bouton est appuyé, alors afficher « Le bouton est appuyé » ;
- sinon afficher « Le bouton n'est pas appuyé ».

C'est le « if ... else ».

La syntaxe est telle que ci-dessous. La partie avec **else** n'est pas obligatoire, tout dépend de ce que l'on veut faire.

Pour la condition, il existe différents opérateurs :

- == : est égale à
- != : n'est pas égal à
- > : est supérieur à
- < : est inférieur à
- >= : est supérieur ou égal à
- <= : est inférieur ou égal à
- && : et *logique*
- || : ou *logique*

CODE ARDUINO

```
if ( condition ) {
    // choses à réaliser si la condition est vraie
}
else {
    // choses à réaliser si la condition n'est pas vraie
}
```

Applications

1. Modifiez la boucle principale de votre programme en complétant les pointillés. Enregistrez votre programme sous le nom P4-Touche_Q1.
2. Rajoutez une LED sur la broche n° 5 ainsi que sa résistance de protection de 220 Ω (voir la fiche n° 1 ou la fiche n° 2).
Écrire un programme qui allumera la LED lorsque le bouton sera appuyé (P4-Touche_Q2).
3. Rajoutez un second bouton dont l'état sera lu sur la broche n° 2.
Modifiez le programme précédent pour que la LED s'allume uniquement lorsque **les deux boutons** sont appuyés en **même temps** (P4-Touche_Q3).

CODE ARDUINO

```
//----- BOUCLE PRINCIPALE -----
void loop(){
    // Lecture de l'état de la TOUCHE
    Etat = digitalRead(Touche);
    if (Etat == ...) {
        Serial.println("Le bouton est appuyé");
    }
    else {
        Serial.println(".....");
    }
}
```

