



Initiation Arduino #4

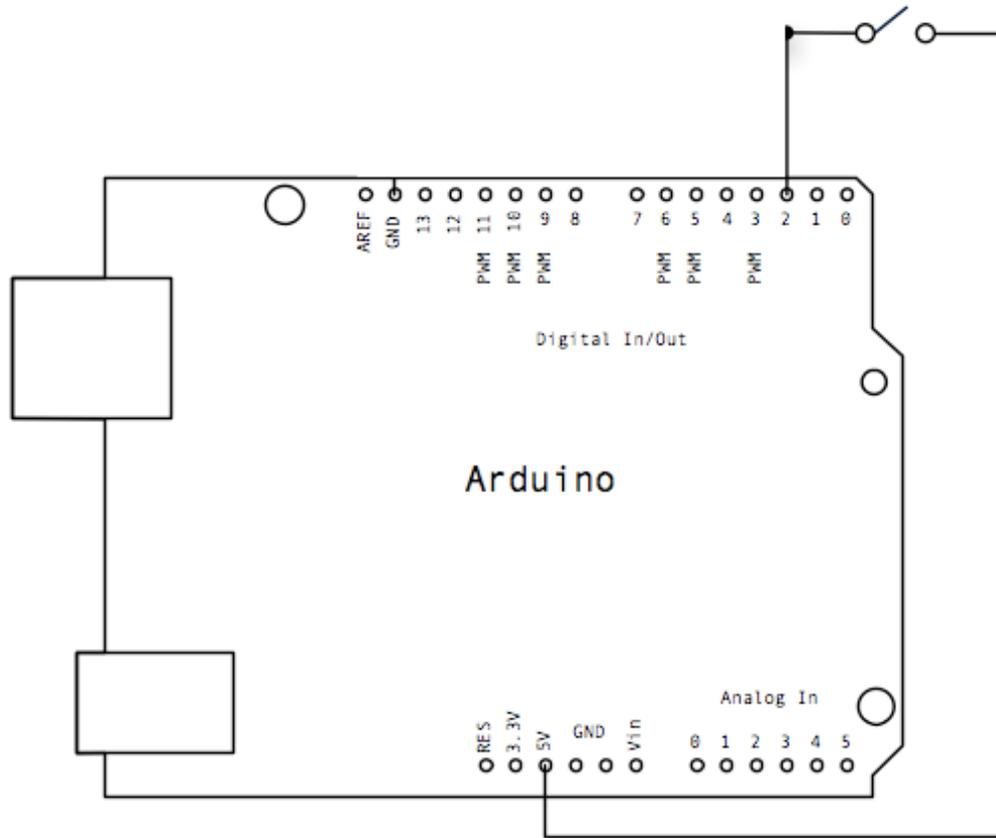
Club astro QF – Pascal ANDRE mars 2016

Objectif

- Détecter l'appui d'un bouton poussoir sur la broche digital 2 de l'arduino.
- Lever les incertitudes d'état avec les résistances de rappel pull down, pull up,
Utiliser les résistances de rappel pull-up intégrées
- Comprendre et modifier le code arduino

En première approche on pourrait penser à un montage simple comme cela

Pin 2 : ETAT ?? Quand ouvert



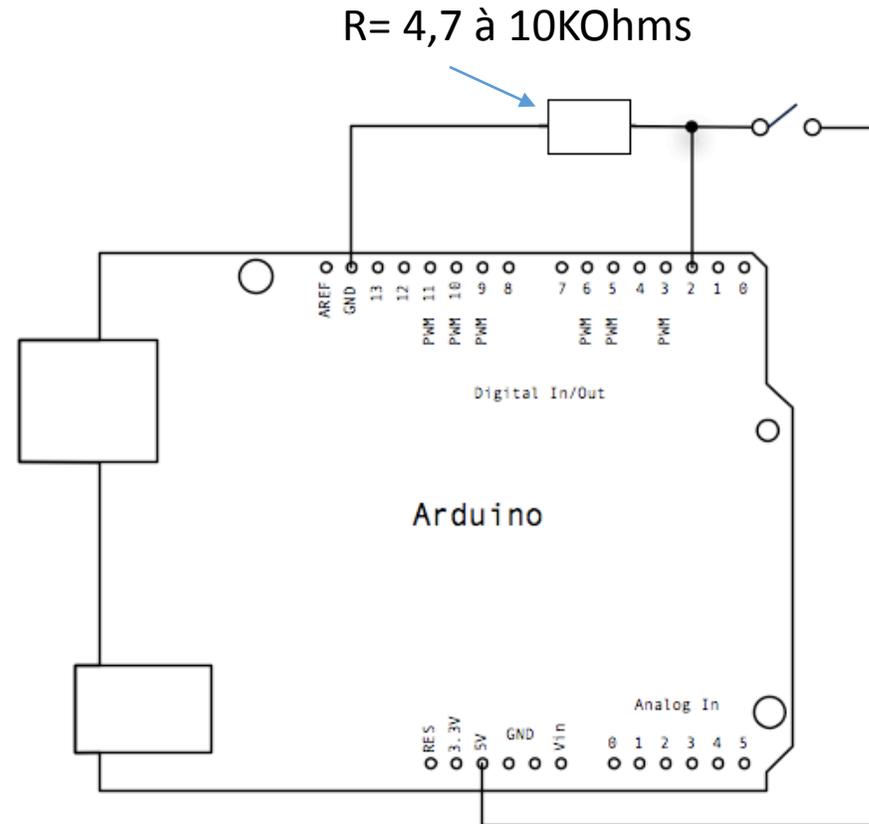
Lorsque le bouton est pressé, l'entrée est à 5 V, **mais lorsque le bouton n'est pas pressé, elle n'est pas définie car elle est "en l'air"**. Les entrées du microcontrôleur équipant l'Arduino sont très sensibles et elles réagissent à la tension qui leur est appliquée. En conséquence, l'entrée laissée en l'air peut avoir n'importe quelle valeur comprise entre 0 et 5 V et être interprétée comme un LOW ou un HIGH par l'instruction `digitalRead(...)`

La solution la « résistance pull down »

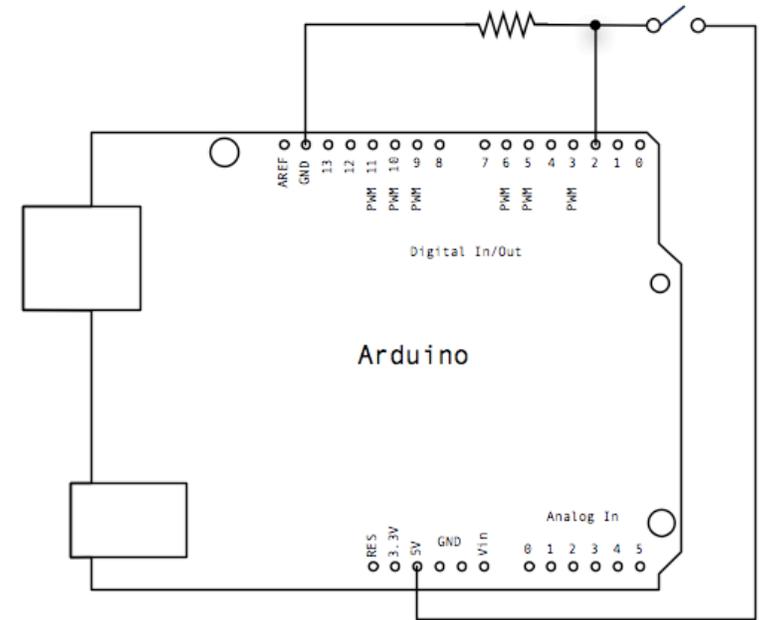
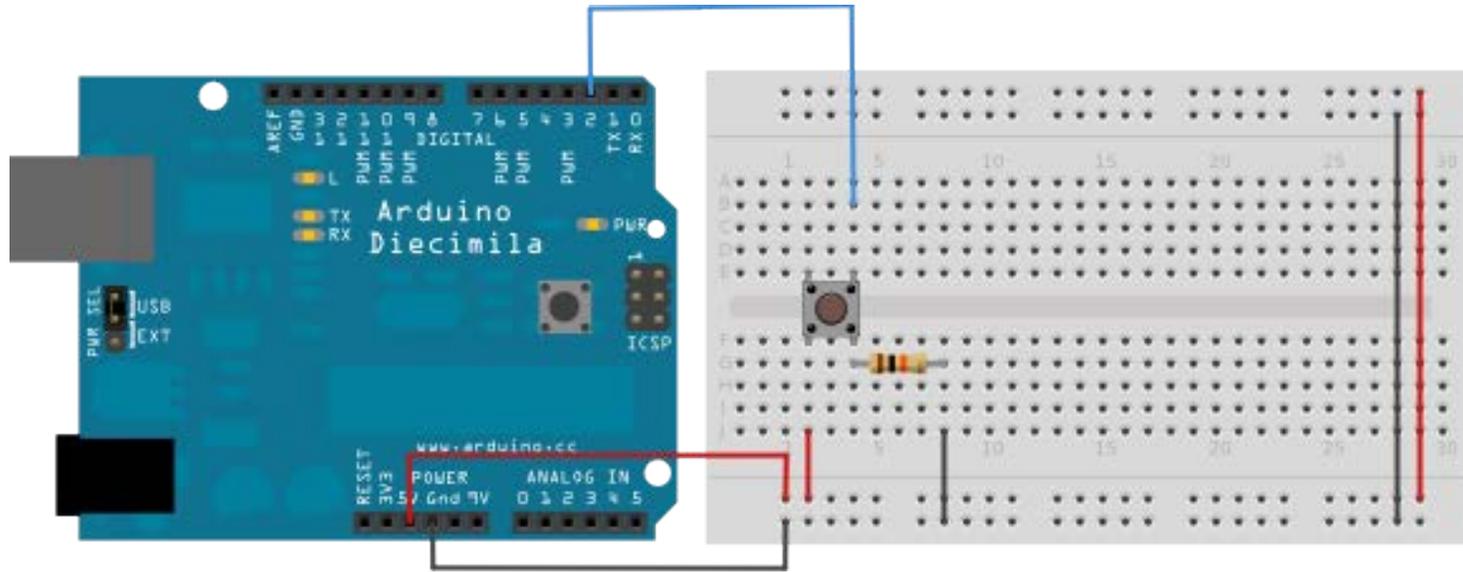
Pour fixer la tension lorsque le bouton n'est pas pressé, on insère une résistance *pull down* (résistance de rappel vers le bas) entre l'entrée et la masse (GND) dont le rôle est d'assurer que l'entrée est à 0 V lorsque le poussoir n'est pas pressé. Lorsque le poussoir est pressé, un courant égal à $5 \text{ V} / R$ circule dans la résistance.

Note : Pour une alimentation de 5 V, une bonne valeur pour R est d'au moins 4.7 Kilo-Ohm, le courant sera alors de 1.1 mA

Pin 2 : ETAT =LOW Quand ouvert ; HIGH(5V) bouton appuyé



Montage platine d'essai



Le code :

Exemples 02 Digital Button

```
/* Button
   Allume et éteint une diode reliée à la broche 13
   lorsque que l'on presse un bouton poussoir relié à la broche 2
*/
const int buttonPin = 2; // le bouton est relié à la broche 2
const int ledPin = 13; // led (intégrée) sur sur la broche 13
int buttonState = 0; // variable pour lire le statut du bouton

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop() {
  buttonState = digitalRead(buttonPin);

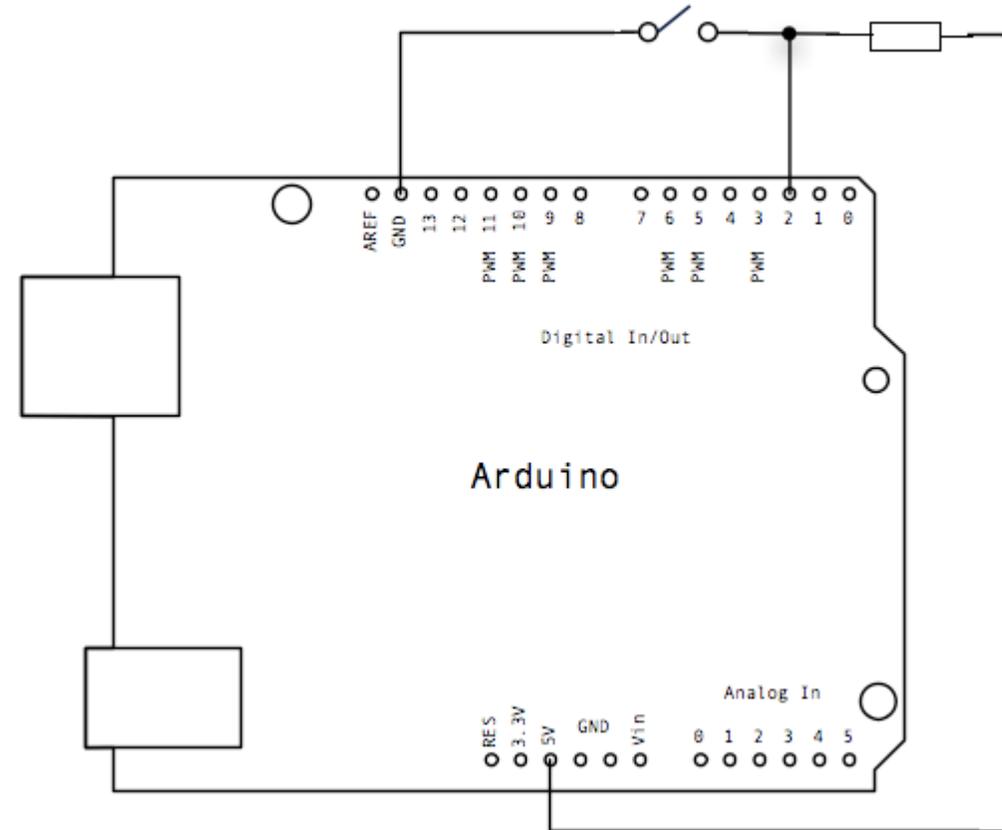
  // teste si le bouton est appuyé si oui le statut du bouton vaut High
  if (buttonState == HIGH) {
    // alors on allume la led
    digitalWrite(ledPin, HIGH);
  } else {
    // sinon on éteint la led
    digitalWrite(ledPin, LOW);
  }
}
```

La solution de choix « résistance pull up »

Pour des raisons liées aux technologies des circuits logiques, on préfère utiliser une résistance pull up (résistance de rappel vers le haut) entre l'entrée et l'alimentation (5 V dans le schéma ci-après) et connecter le poussoir comme suit.

Note : Pour une alimentation de 5 V, une bonne valeur pour R est d'au moins 4.7 Kilo-Ohm, le courant sera alors de 1.1 mA

Pin 2 : ETAT =HIGH Quand ouvert ; LOW (0V) bouton appuyé



Dans le programme, **il faut inverser le résultat retourné** par `digitalRead` lorsque le bouton est pressé :
`if (digitalRead (2) == LOW)`

Résistances pull up intégrées à l'Arduino

l'Arduino dispose de résistances pull up internes qui évitent l'utilisation de résistances externes. Ces résistances internes ont une valeur comprise entre 20 et 50 Kilo-Ohm, valeur plus élevée que la valeur recommandée pour une résistance externe, mais suffisante. Le schéma ci-après illustre le câblage du poussoir sans résistance pull up externe, la résistance pull up interne étant représentée *en pointillé* et à l'intérieur de la carte Arduino,

Pour activer la résistance pull up interne d'une entrée, par exemple la no 2, il suffit d'écrire, dans la fonction setup :

```
pinMode (2, INPUT_PULLUP);
```

À la place de `pinMode(2, INPUT);`

