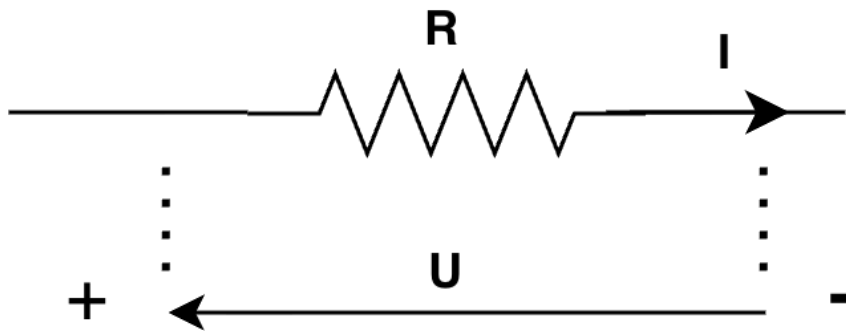


SESSION 03 · DÉBUTANT ● - Boutons et logique

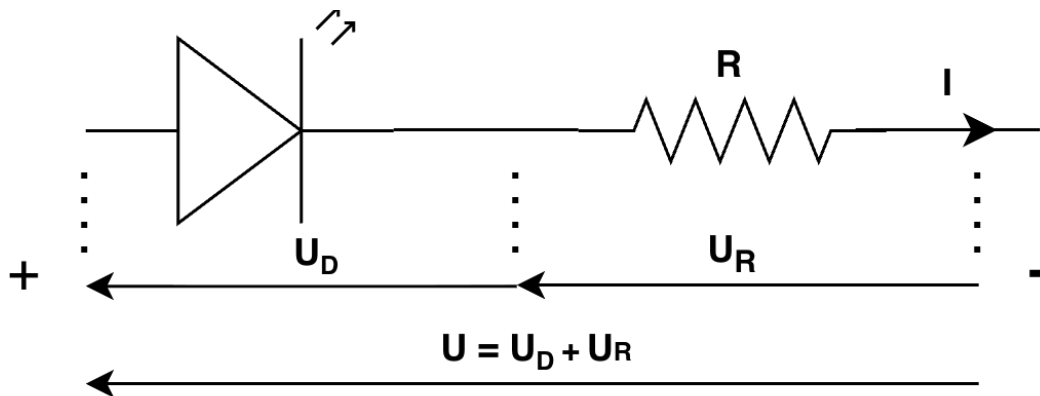
🕒 Rappels des sessions précédentes

📝 Loi d'ohm (simple résistance)



📌 $U \text{ (volts)} = R \text{ (ohms)} * I \text{ (A)}$

📝 Résistance et LED en série



📌 $U = U_D + U_R = U_D + R * I$

$$R = (U - U_D) / I$$

$$R = (5 - 1,8) / 0,015 = 213 \text{ ohms}$$

=> On choisit $R = 220 \text{ ohms}$ (valeur normalisée)

Programmation d'une broche en sortie

```
int maLed = 8; // On déclare la variable qui contient le numéro de broche (pin)

pinMode(maLed, OUTPUT); // On déclare la broche en mode SORTIE

digitalWrite(maLed, HIGH); // On met le niveau de la broche à 5V

digitalWrite(maLed, LOW); // On met le niveau de la broche à 0V
```

Objectifs de la session

- Lire l'état d'un bouton avec `digitalRead()` et `INPUT_PULLUP`
- Comprendre pourquoi un bouton retourne LOW quand il est pressé avec `INPUT_PULLUP`
- Gérer l'anti-rebond (bounce) avec un délai
- Utiliser une variable booléenne pour mémoriser un état (bascule)
- Communiquer avec le PC via le Serial Monitor

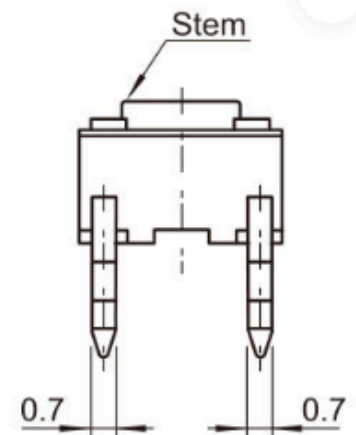
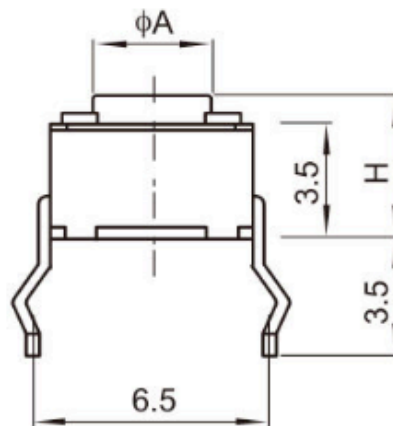
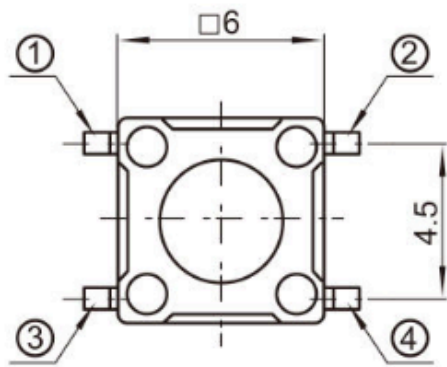
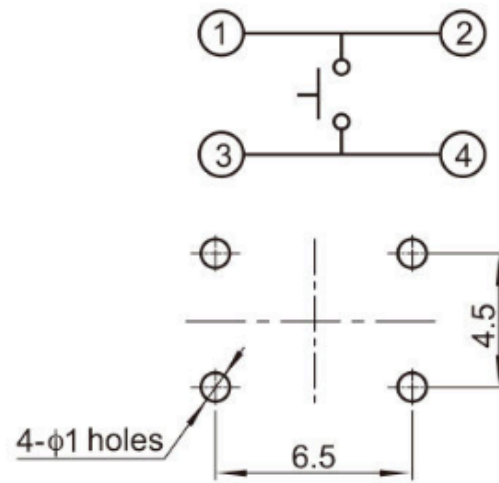
Matériel

Composant	Qté	Note
Arduino Uno + câble USB	1	Du kit
Breadboard	1	Du kit
Bouton poussoir	1	Du kit
LED rouge	1	Du kit
Résistance 220Ω	1	Code : rouge-rouge-marron
Câbles Dupont	4	Du kit

Le bouton poussoir



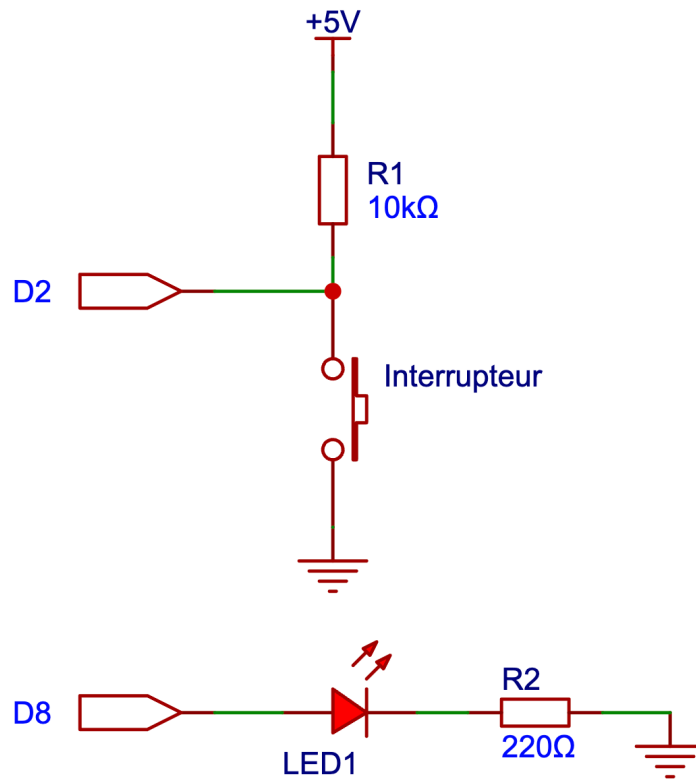
H	A
4.3	3.5
5	3.5
7	3.3
9.5	3
13	3
17	3



⚡ Montage 1 — Bouton → LED

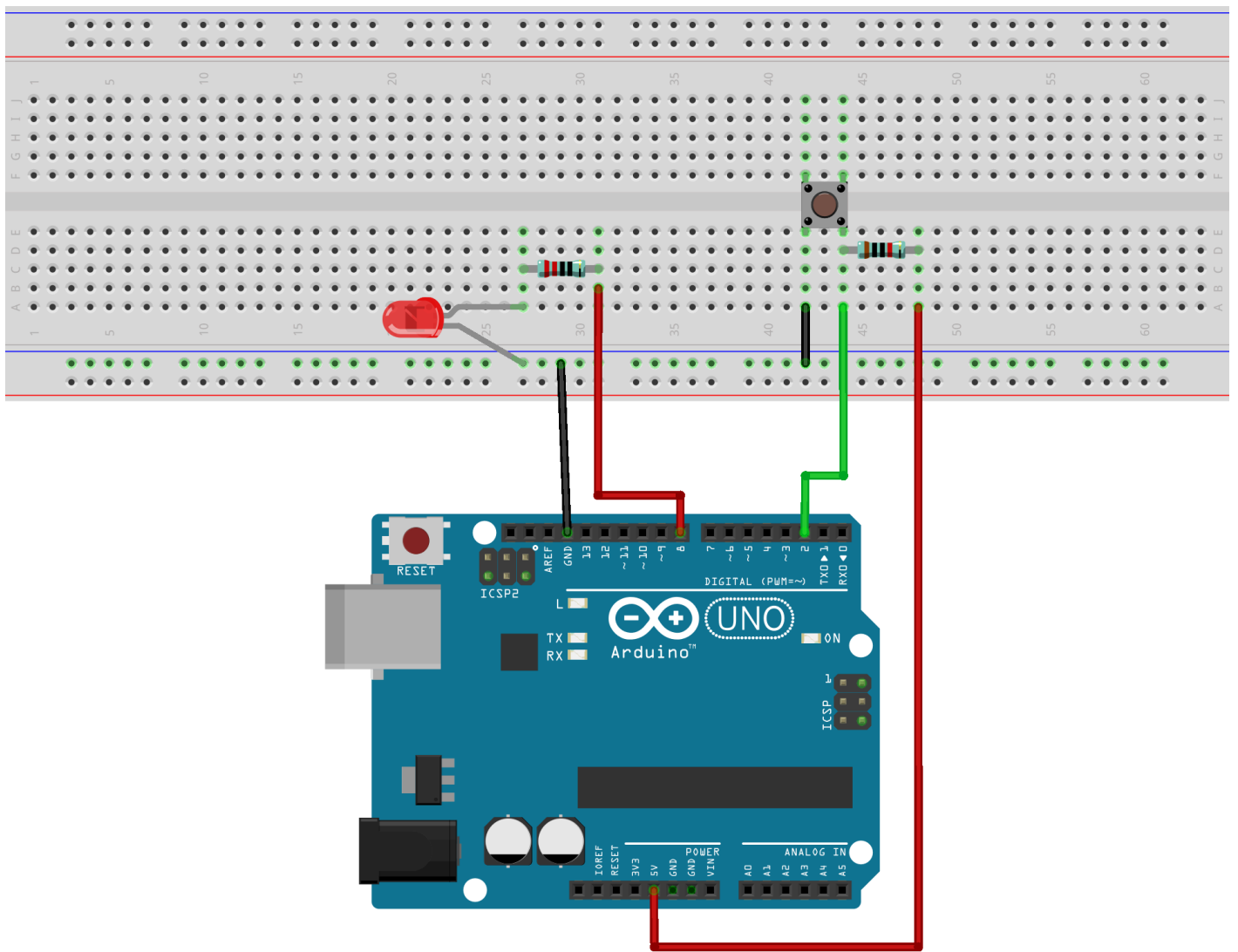
Allumage d'une LED par appui sur un bouton poussoir avec une résistance de rappel (+5V).

🔌 Schéma de principe



- Bouton non appuyé => tension en D2 = 5V
- Bouton appuyé => tension en D2= 0V

Schéma de câblage



fritzing

⚡ Code

```
int bouton = 2;      // Bouton connecté à la pin 2
int led      = 8;    // LED connectée à la pin 8

void setup() {
  pinMode(bouton, INPUT);      // Pin 2 en entrée
  pinMode(led, OUTPUT);        // Pin 8 en sortie
}

void loop() {
  if (digitalRead(bouton) == LOW) { // Lecture du niveau sur la pin 2, LOW = pressé!
    digitalWrite(led, HIGH);        // On allume la LED
  } else {
    digitalWrite(led, LOW);          // On éteint la LED
  }
}
```

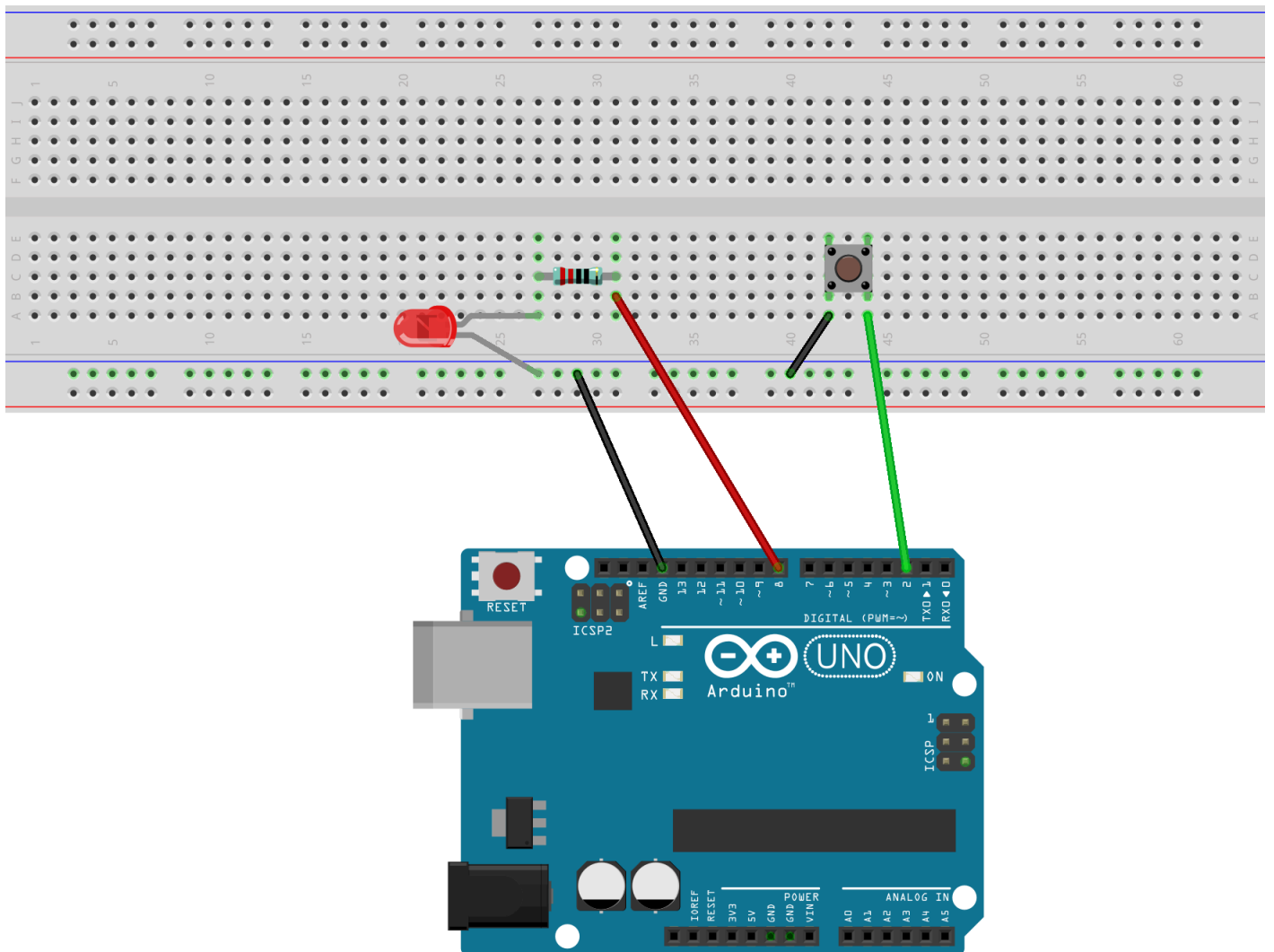
```
}  
}
```

Structure du test if else ()

⚡ Montage 2 (avec résistance de rappel interne)

Le principe reste identique au montage 1: l'allumage de la LED se fait par appui sur un bouton poussoir mais résistance de rappel interne à l'Arduino.

🔌 Schéma de câblage



fritzing

⚡ Code

```

int bouton = 2;      // Bouton connecté à la pin 2
int led     = 8;      // LED connectée à la pin 8

void setup() {
  pinMode(bouton, INPUT_PULLUP); // Pin 2 en entrée et résistance interne activée
  pinMode(led, OUTPUT);          // Pin 8 en sortie
}

void loop() {
  if (digitalRead(bouton) == LOW) { // Lecture du niveau sur la pin 2, LOW = pressé!
    digitalWrite(led, HIGH);        // On allume la LED
  } else {
    digitalWrite(led, LOW);         // On éteint la LED
  }
}

```

Le pull-up interne

 Avec INPUT_PULLUP, une résistance interne de ~50kΩ relie la pin au 5V. Sans appui : pin = HIGH. Avec appui (bouton relie pin à GND) : pin = LOW.

Montage 3 — Bascule ON/OFF

Le montage est identique mais le fonctionnement est le suivant: - Un appui sur le bouton déclenche l'allumage de la LED et la LED reste allumée - l'appui suivant éteint la LED et elle reste éteinte

```

int bouton    = 2;
int led       = 8;
bool etat     = false;          // Mémoire si la LED est allumée
bool ancienEtatBouton = HIGH;   // Mémoire la position du bouton (LOW=pressé)

void setup() {
  pinMode(bouton, INPUT_PULLUP); // Bouton non pressé, l'entrée est maintenue à HIGH
  pinMode(led, OUTPUT);
}

void loop() {

  // Lecture de la position du bouton
  // Bouton pressé => LOW, non pressé = HIGH

```

```

bool nouvelEtatBouton = digitalRead(bouton);

// Détecter le front descendant (appui sur le bouton)
if (nouvelEtatBouton == LOW && ancienEtatBouton == HIGH) {
    etat = !etat;           // Inverser l'état de la LED
    digitalWrite(led, etat); // On change effectivement l'état de la LED
    delay(50);              // Anti-rebond
}

// On mémorise l'état du bouton
ancienEtatBouton = nouvelEtatBouton;
delay(50);
}

```

Montage 3 — Utilisation du Moniteur Série (Serial Monitor)

Le moniteur série est une fenêtre de l'IDE Arduino pouvant recevoir des messages de l'Arduino (et en émettre également). Ces messages peuvent permettre de tracer l'exécution de votre code en phase de mise au point.

```

int bouton    = 2;           // Bouton connecté à la pin 2
int compteur = 0;           // Compteur initialisé à 0

void setup() {
    pinMode(bouton, INPUT_PULLUP); // Pin 2 en entrée
    Serial.begin(9600);           // Programmer le moniteur série à 9600 bauds
    delay(1000);                  // Attente de 1 seconde
    Serial.println("Prêt ! Appuyez sur le bouton..."); // Envoi d'un message vers le moni
}

void loop() {
    if (digitalRead(bouton) == LOW) { // Si le bouton est appuyé
        compteur = compteur + 1;      // On ajoute 1 au compteur
        Serial.print("Détection appui bouton numéro : "); // Affichage sans retour à la li
        Serial.println(compteur);     // Affichage avec retour à la li
        delay(300);                   // Attente 300 ms
    }
}

```



Ouvrir le Serial Monitor : **Outils** → **Moniteur série** (ou Ctrl+Maj+M). S'assurer que la vitesse est réglée sur **9600 bauds**.