

Pour le corrigé de cette activité vous trouverez le code dans les versions blockly et C/C++ (exercice 1) ou C/C++ (exercices 2 & 3) pour faire fonctionner les simulations qui correspondent aux exercices.

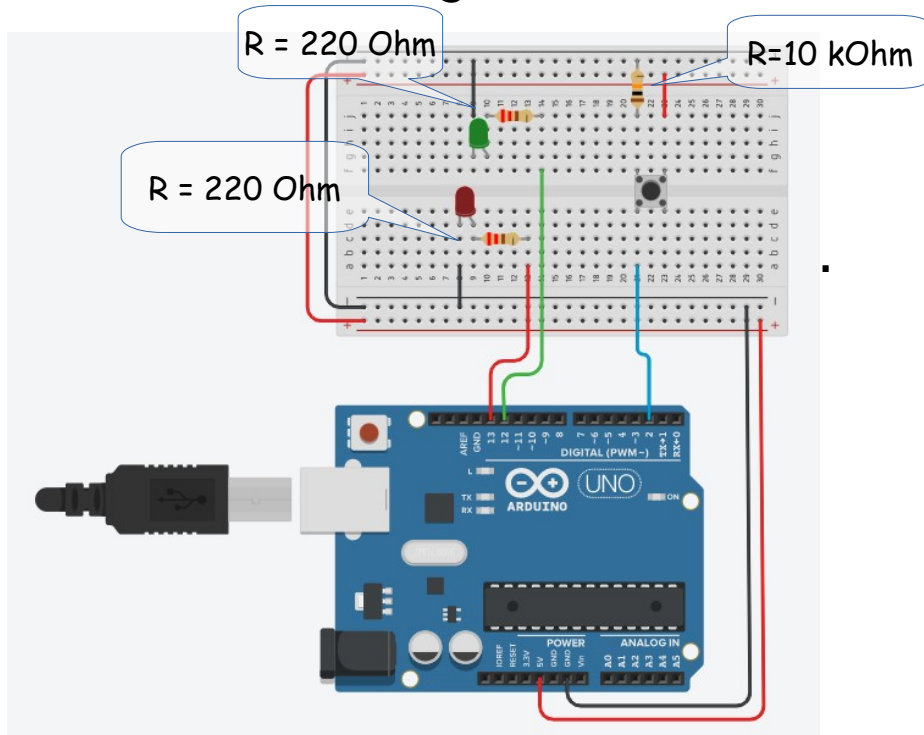
Les corrections que je vous propose peuvent être différentes de ce que vous avez fait, même si votre programme fonctionne. En programmation il n'y a que très rarement qu'une solution. Il y a simplement des solutions plus élégantes ou plus efficaces que d'autres. Le choix dépendra de vos priorités (vitesse d'exécution, encombrement mémoire, lisibilité du code...) Les commentaires expliquent le code.

Je n'ai volontairement pas mis de corrections pour l'exercice 4

Je rappelle que la lecture approfondie des documents ressources fournis est nécessaire pour bien assimiler les notions utilisées :

- Arduino Uno-01.pdf
- Arduino Uno-02.pdf

## Le schéma de câblage



## Mode de rendu

J'insiste sur le mode de rendu et sur la nécessité de respecter la consigne de nommage des simulations.

Exemple : Mon pseudo pour accéder à la classe est **sti3-patrick** , j'appelle alors ma simulation pour l'exercice 1  
« Ex1 sti3-patrick »

**Vous devez bien évidemment utiliser votre pseudo !**

## Correction

Exercice 1 : Tant que le bouton poussoir est enfoncé, la LED verte doit s'allumer. Au repos c'est la LED rouge qui s'allume.



```
1 int bp = 0;
2
3 void setup()
4 {
5   pinMode(2, INPUT);
6   pinMode(12, OUTPUT);
7   pinMode(13, OUTPUT);
8 }
9
10 void loop()
11 {
12   bp = digitalRead(2);
13   if (bp == HIGH) {
14     // Led Verte active
15     digitalWrite(12, HIGH);
16     digitalWrite(13, LOW);
17   } else {
18     // Led Rouge active
19     digitalWrite(12, LOW);
20     digitalWrite(13, HIGH);
21   }
22   delay(100); // Wait for 100 millisecond(s)
23 }
```

Exercice 2 : Quand on clique sur le bouton poussoir, la LED allumée doit changer.

Au départ la LED Rouge est allumée, si je clique sur le BP, elle s'éteint et la LED verte s'allume. Si j'appuie à nouveau sur BP la LED rouge s'allume et la verte s'éteint et ainsi de suite.

```
1  /* Exercice 2
2  ** **** */
3  int led_on;           // La led allumée
4  int bp;              // L'état du BP
5
6  void setup()
7  {   pinMode(2, INPUT);   // Entrée BP
8     pinMode(12, OUTPUT); // Sortie Led Verte
9     pinMode(13, OUTPUT); // Sortie Led Rouge
10    // Au départ la led rouge est allumée.
11    digitalWrite(13, HIGH);
12    digitalWrite(12, LOW);
13    led_on=0;          // 0 = Rouge - 1 = Verte
14  }
15
16  void loop()
17  { bp=digitalRead(2);
18    if (bp==HIGH)
19    { if (led_on)      // VERT
20      { digitalWrite(13, HIGH);
21        digitalWrite(12, LOW);
22        led_on=0;
23      }
24    else              // Rouge
25      { digitalWrite(13, LOW);
26        digitalWrite(12, HIGH);
27        led_on=1;
28      }
29    // Maintenant on attend que l'utilisateur relâche le BP
30    while (bp==HIGH)
31      { delay(150);
32        bp=digitalRead(2);
33      }
34  }
35  delay(100); // On attend 100 ms avant de recommencer
36 }
```

Exercice 3 : Quand on clique sur le bouton poussoir la LED allumée doit s'éteindre. Quand on clique à nouveau sur le BP on allume l'autre LED, et ainsi de suite.

```
1  /* Exercice 3
2  ** **** */
3
4  int led_color;           // La couleur de la led allumée
5  int is_led_on;          // Une led allumée=1, 0 sinon
6  int bp;                  // L'état du BP
7
8  void setup()
9  {  pinMode(2, INPUT);    // Entrée BP
10     pinMode(12, OUTPUT); // Sortie Led Verte
11     pinMode(13, OUTPUT); // Sortie Led Rouge
12
13     // Au départ la led rouge est allumée.
14     digitalWrite(13, HIGH);
15     digitalWrite(12, LOW);
16     led_color=0;         // 0 = Rouge - 1 = Verte
17     is_led_on=1;
18 }
19
20 void loop()
21 {
22     bp=digitalRead(2);
23     if (bp==HIGH)
24     {  if (is_led_on)
25         {  // Une Led est allumée
26             digitalWrite(12, LOW);
27             digitalWrite(13, LOW);
28             is_led_on=0;
29         }
30         else
31         {  // les deux leds sont éteintes
32             if (led_color)
33             {  // VERT
34                 digitalWrite(13, HIGH);
35                 led_color=0;
36             }
37             else
38             {  // Rouge
39                 digitalWrite(12, HIGH);
40                 led_color=1;
41             }
42             is_led_on=1;
43         }
44
45         // Maintenant on attend que l'utilisateur
46         // relâche le BP
47         while (bp==HIGH)
48         {  delay(150);
49             bp=digitalRead(2);
50         }
51     }
52     delay(100); // On attend 100 ms avant de recommencer
53 }
54
```

