

Introduction à la simulation TinkerCAD

<https://www.tinkercad.com/classrooms>

Code de la classe : CYA-HSP-BTJ

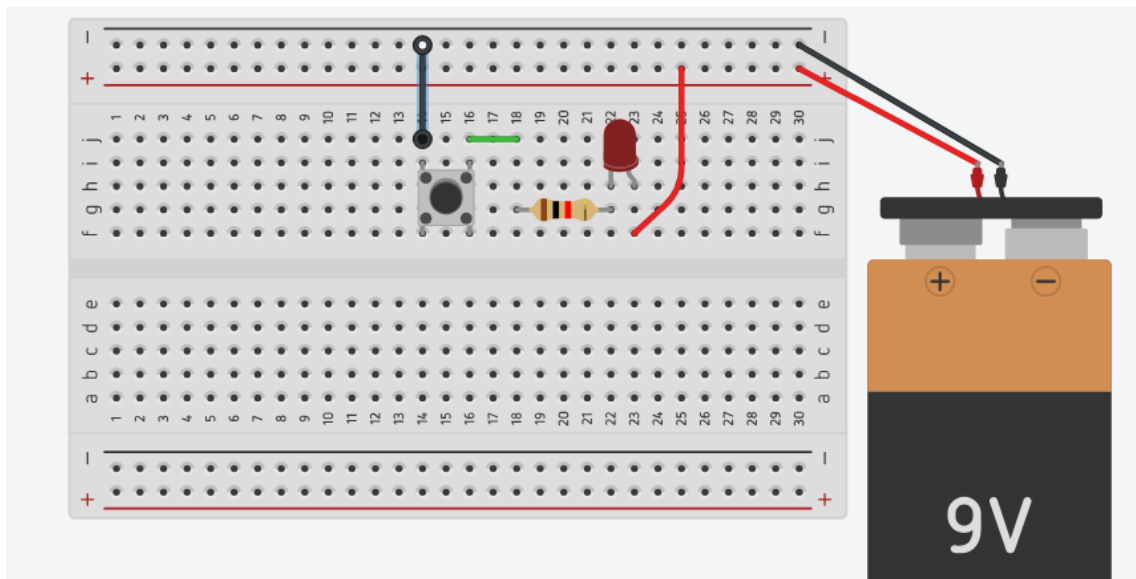
Pseudo :



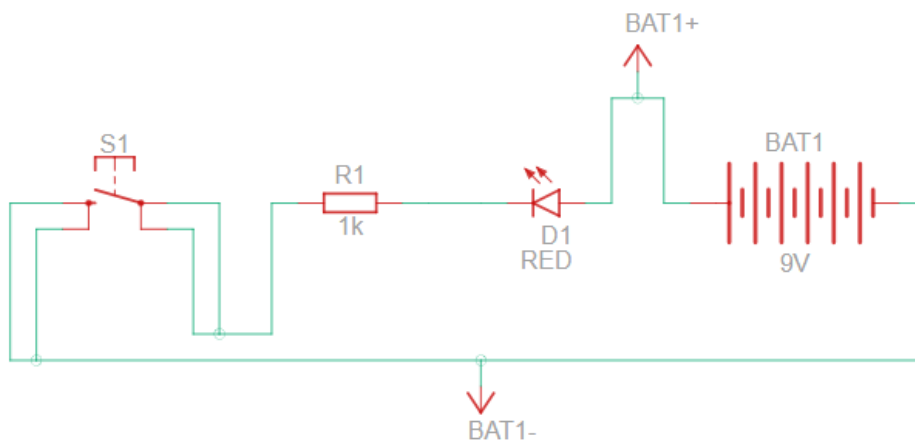
1. C'est quoi TinkerCAD ?

Le logiciel en ligne Circuit TINKERCAD permet de réaliser des circuits avec des composants (capteurs, servo-moteurs, voltmètres, ...) et d'en simuler le fonctionnement piloter par un programme en lignes de commande Arduino ou Scratch.

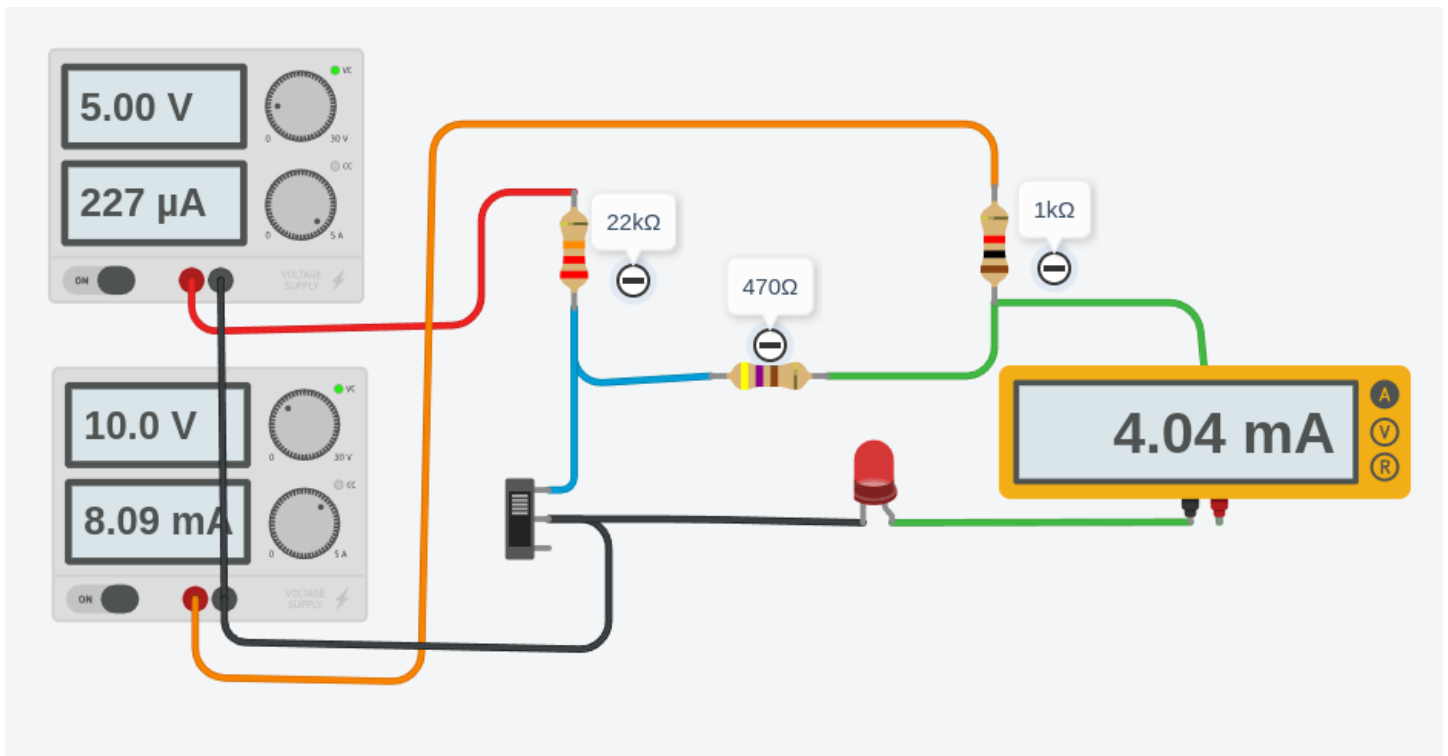
Vue de circuit :



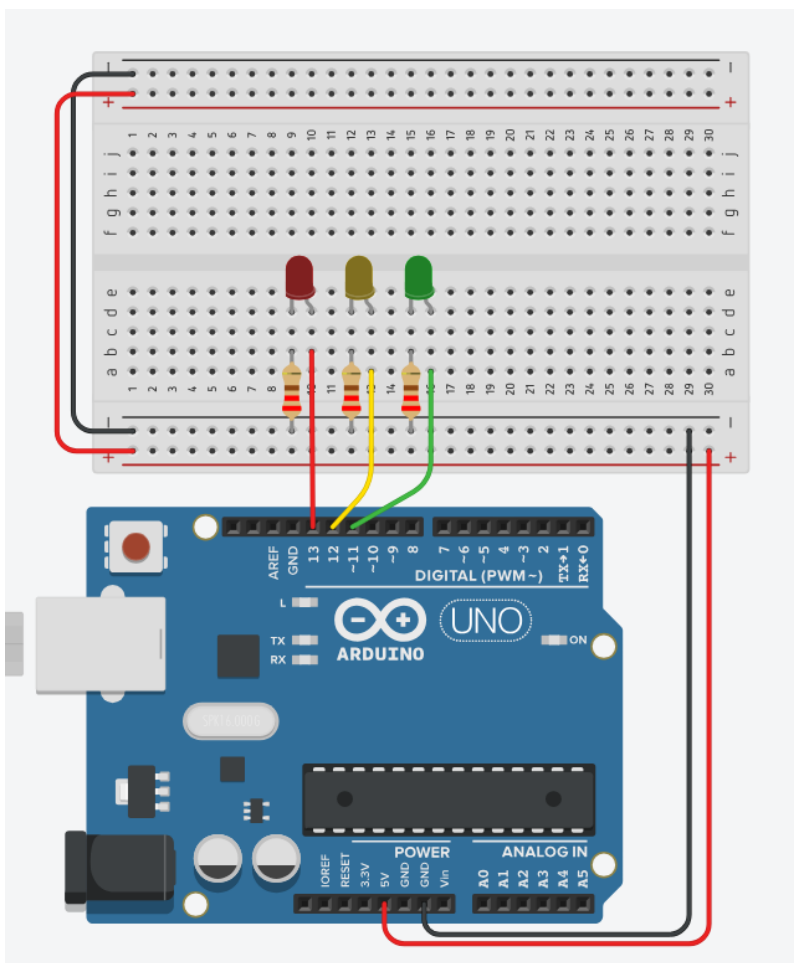
Vue schématique :



2. Réaliser un circuit



3. Réaliser un circuit et son code



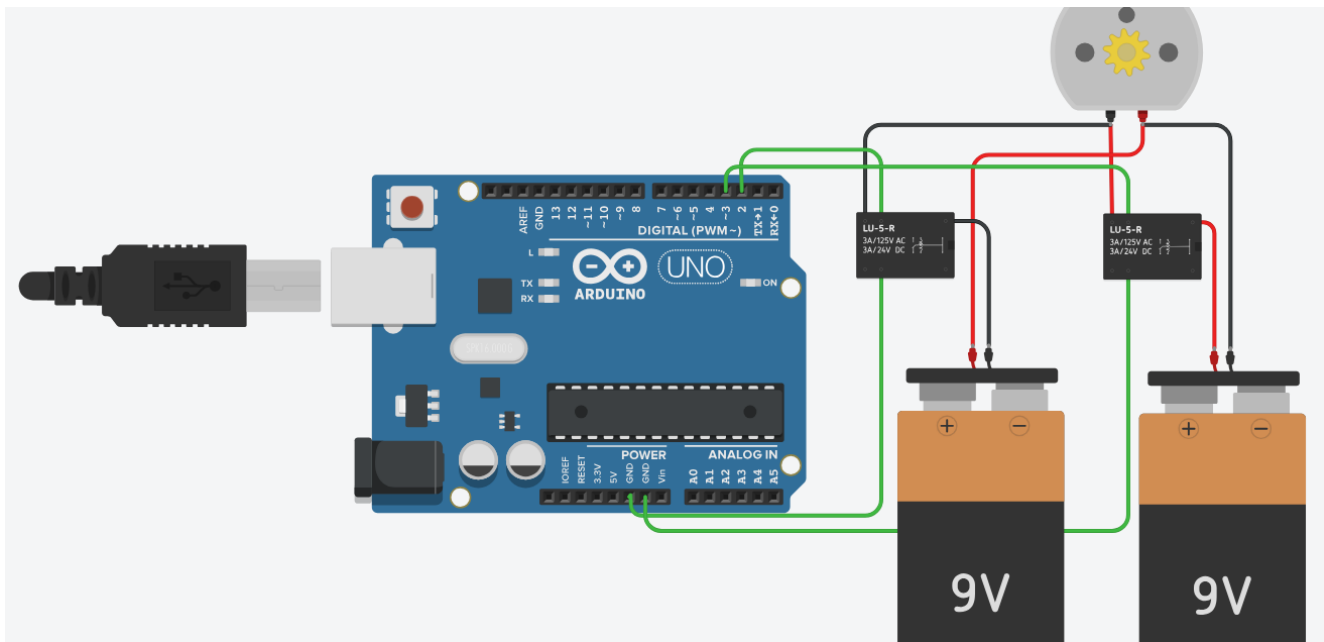
Code :

```
// C++ code
int animationSpeed = 0;

void setup()
{
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
}

void loop()
{
  animationSpeed = 400;
  digitalWrite(13, HIGH);
  delay(animationSpeed);
  digitalWrite(13, LOW);
  delay(animationSpeed);
  digitalWrite(12, HIGH);
  delay(animationSpeed);
  digitalWrite(12, LOW);
  delay(animationSpeed);
  digitalWrite(11, HIGH);
  delay(animationSpeed);
  digitalWrite(11, LOW);
  delay(animationSpeed);
}
```

4. Réaliser un circuit et son code



Code :

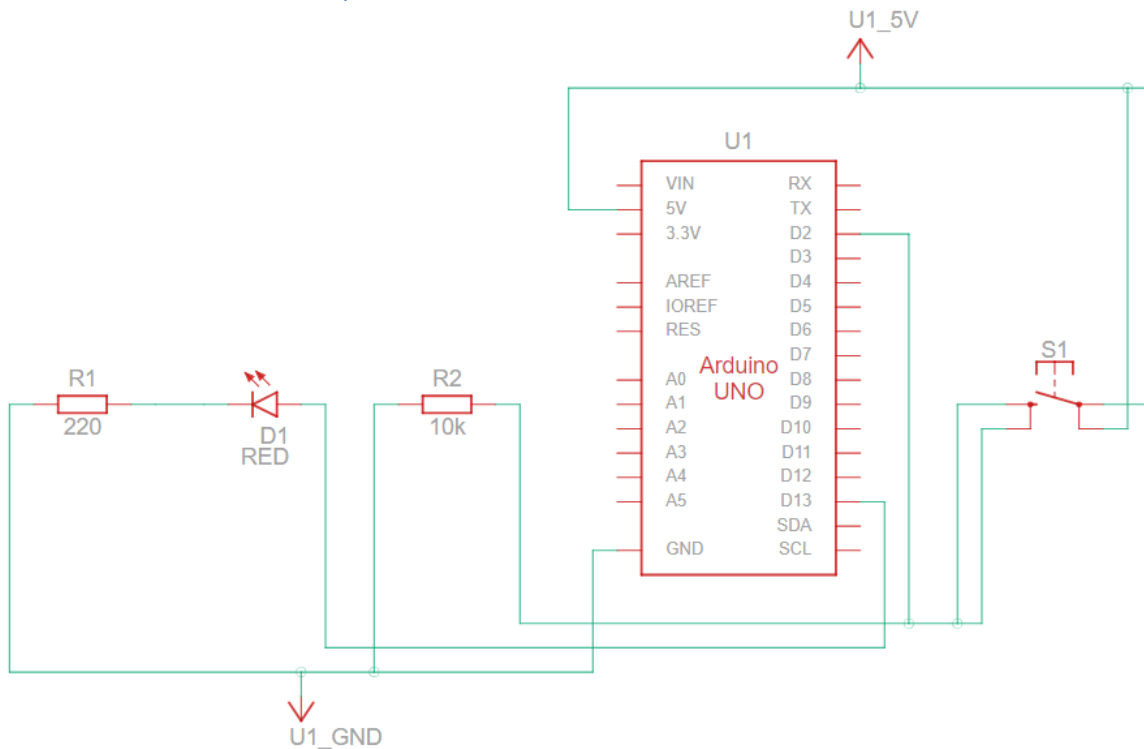
```
void setup()
{
  pinMode(3, OUTPUT);
  pinMode(2, OUTPUT);
}

void loop()
{
  digitalWrite(3, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(3, LOW);
  delay(1000); //wait just to see the result
  digitalWrite(2, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(2, LOW);
  delay(1000); //wait just to see the result
}
```

Liste de composants :

Nom	Quantité	Composant
M1	1	Moteur à courant continu
U2	1	Arduino Uno R3
K1 K2	2	Relais unipolaire bidirectionnel
BAT1 BAT2	2	Pile 9 V

5. Réaliser un circuit à partir du schéma et son code



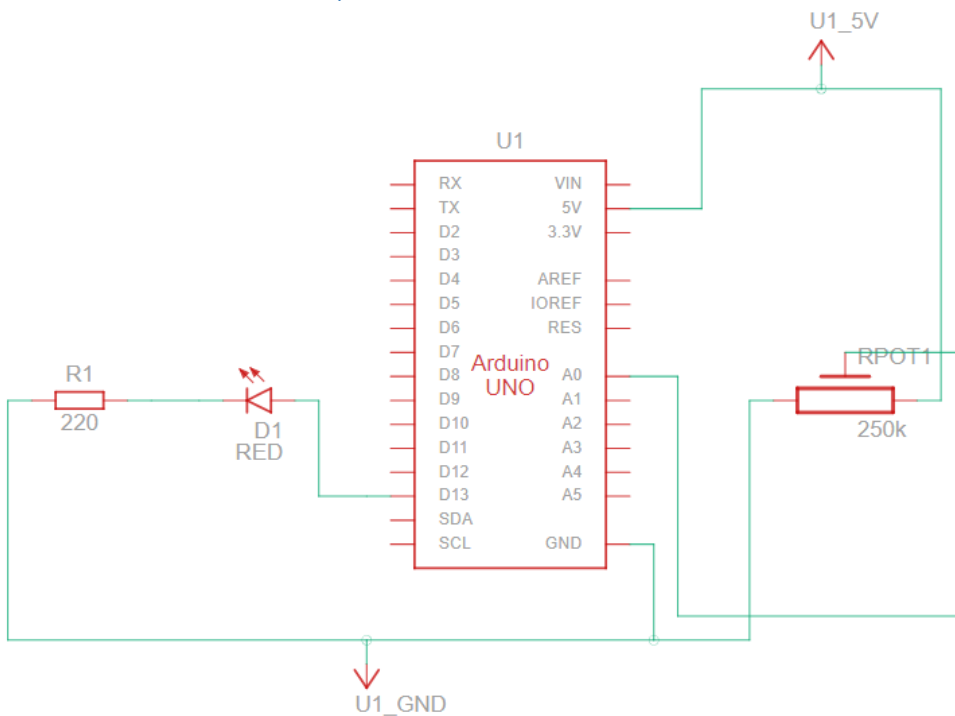
Code :

```
// C++ code
//
int buttonState = 0;

void setup()
{
  pinMode(2, INPUT);
  pinMode(13, OUTPUT);
}

void loop()
{
  // read the state of the pushbutton
  buttonState = digitalRead(2);
  // check if pushbutton is pressed. if it is, the button state is HIGH
  if (buttonState == HIGH) {
    digitalWrite(LED_BUILTIN, HIGH);
  }
  else {
    digitalWrite(LED_BUILTIN, LOW);
  }
  delay(10);
}
```

6. Réaliser un circuit à partir du schéma et son code



Code :

```
// C++ code
//
int sensorValue = 0;

void setup()
{
  pinMode(A0, INPUT);
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  // read the value from the sensor
  sensorValue = analogRead(A0);
  // turn the LED on
  digitalWrite(13, HIGH);
  // pause the program for <sensorValue> milliseconds
  delay(sensorValue); // Wait for sensorValue millisecond(s)
  // turn the LED off
  digitalWrite(13, LOW);
  delay(sensorValue);
}
```

7. Réaliser un circuit à partir de son code et la liste des composants

Code :

```
// C++ code
//
int sensorValue = 0;

void setup()
{
  pinMode(A0, INPUT);
  Serial.begin(9600);
  pinMode(9, OUTPUT);
}

void loop()
{
  // read the value from the sensor
  sensorValue = analogRead(A0);
  // map the sensor reading to a range for the LED
  analogWrite(9, map(sensorValue, 0, 1023, 0, 255));
  delay(100); // Wait for 100 millisecond(s)
}
```

Liste de composants :

Nom	Quantité	Composant
U1	1	Arduino Uno R3
D1	1	Rouge LED
R1	1	220 Ω Résistance
R2	1	Photorésistance
R6	1	4.7 k Ω Résistance

Fonction du prototype :

On mesure l'intensité lumineuse avec une photorésistance. Selon cette valeur on éclaire plus ou moins la LED.