

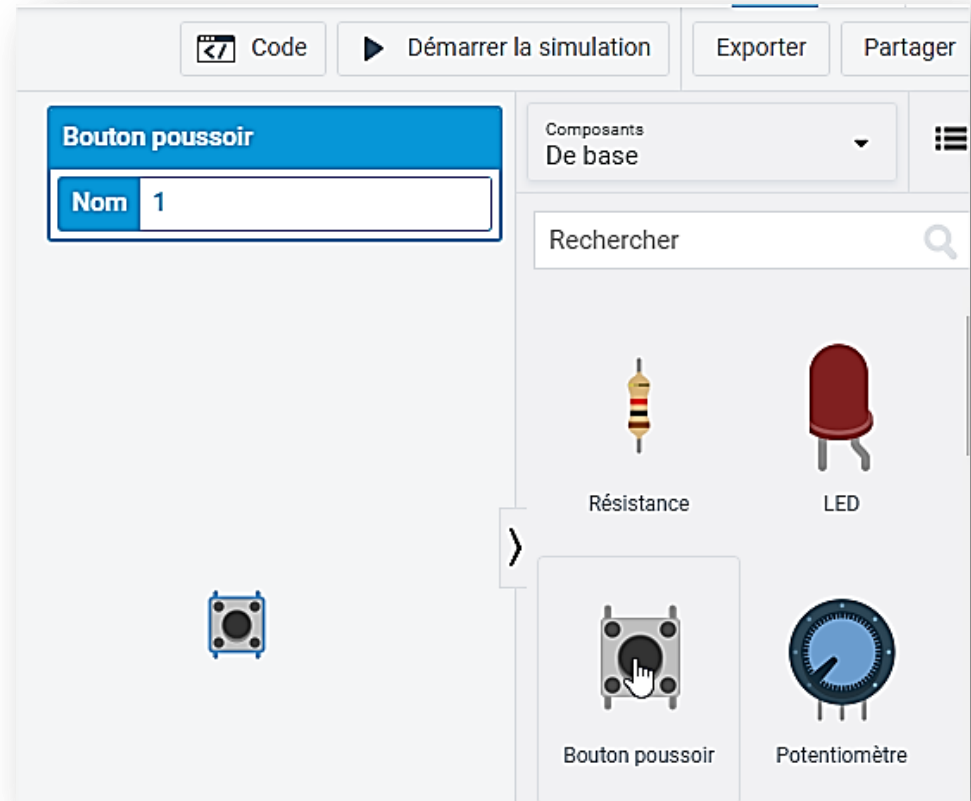


Lancer  
Tinkercad et  
créer un  
nouveau circuit.



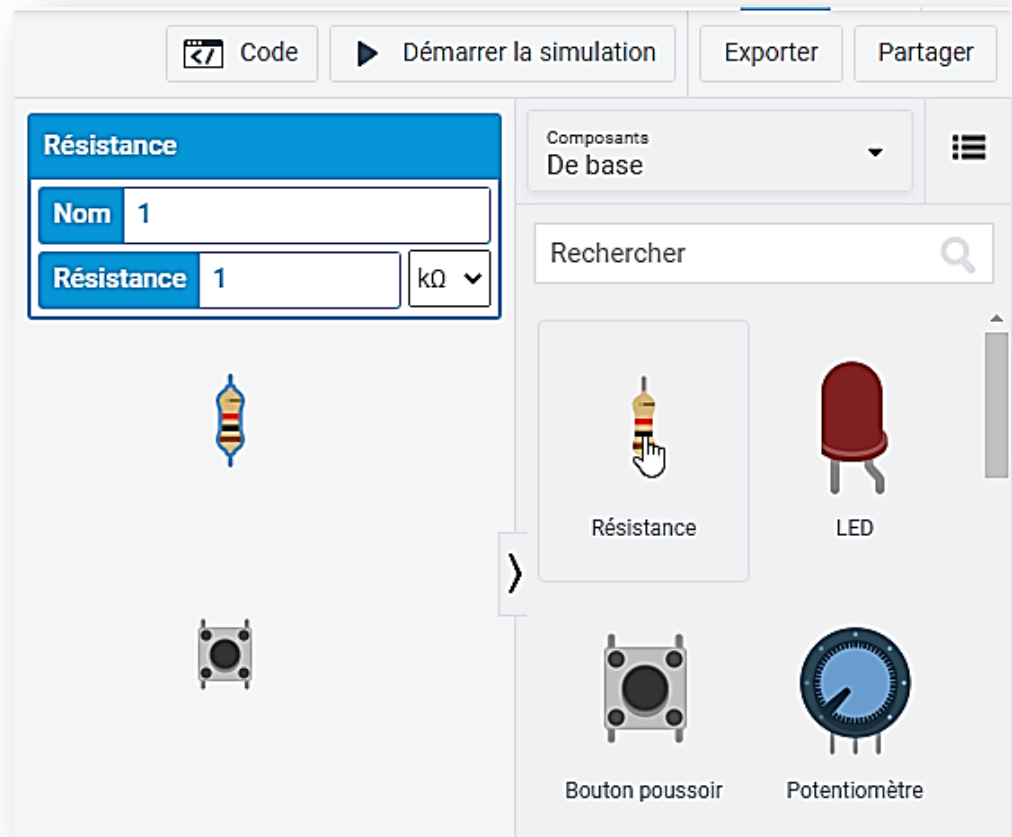


Placer un  
bouton sur la  
feuille



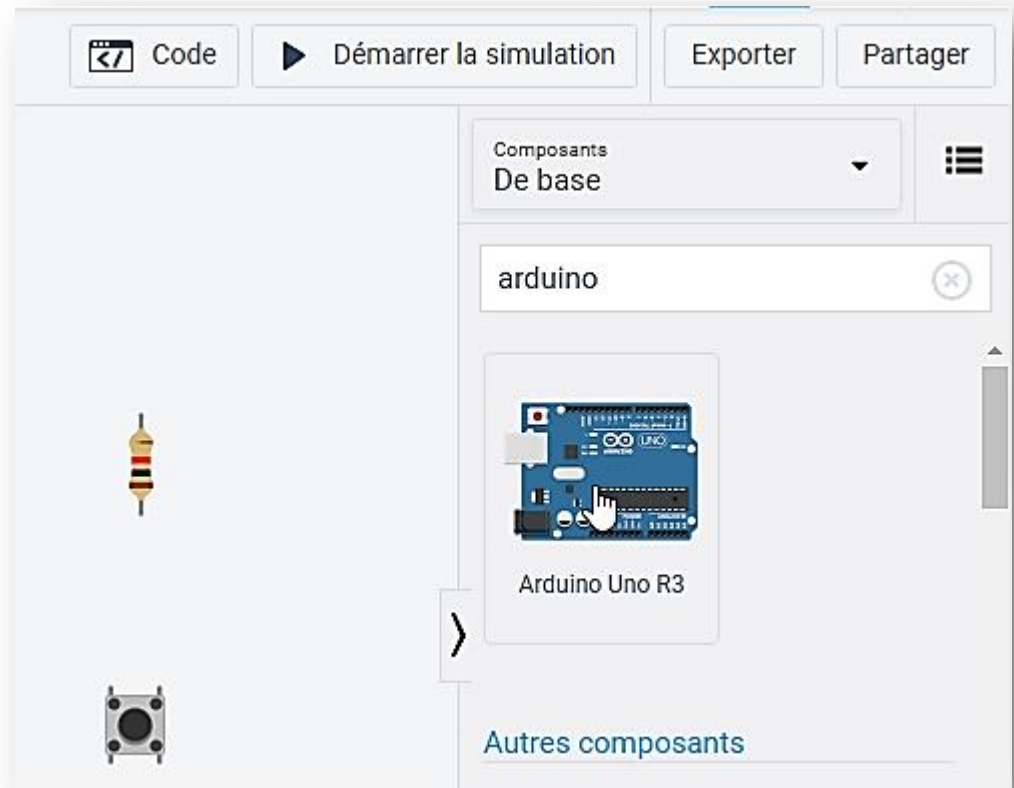


Placer une  
résistance sur  
la feuille



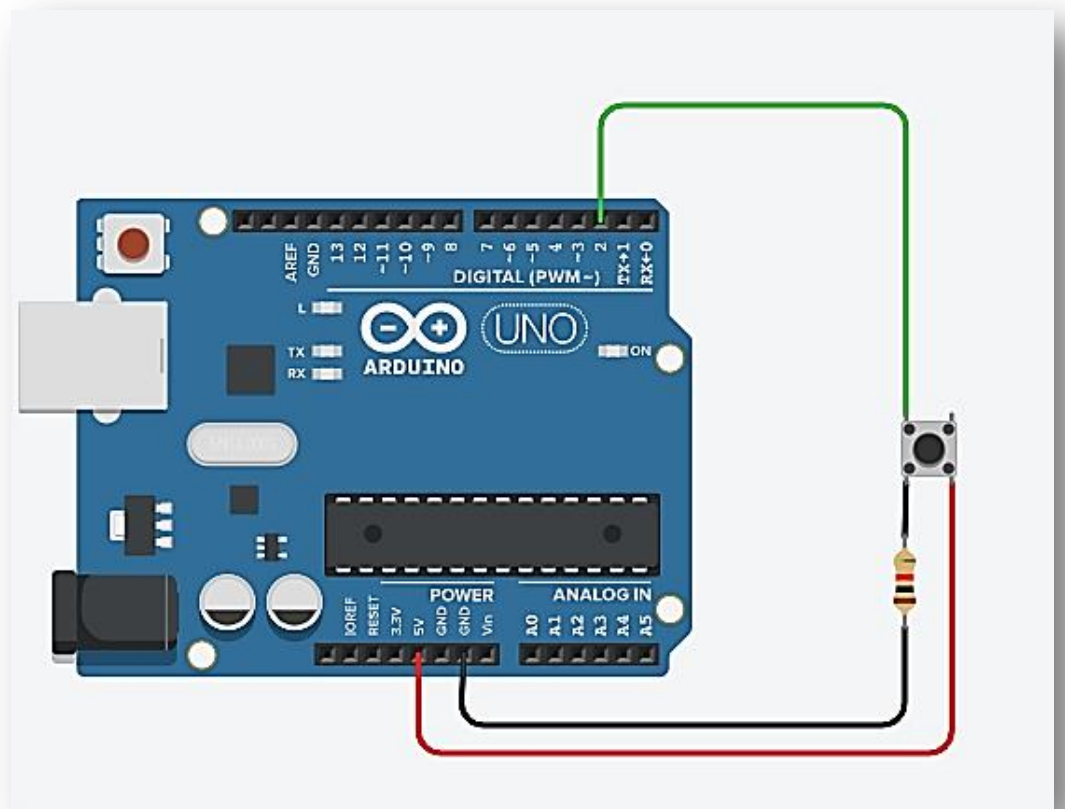


Puis placer  
une carte  
arduino.



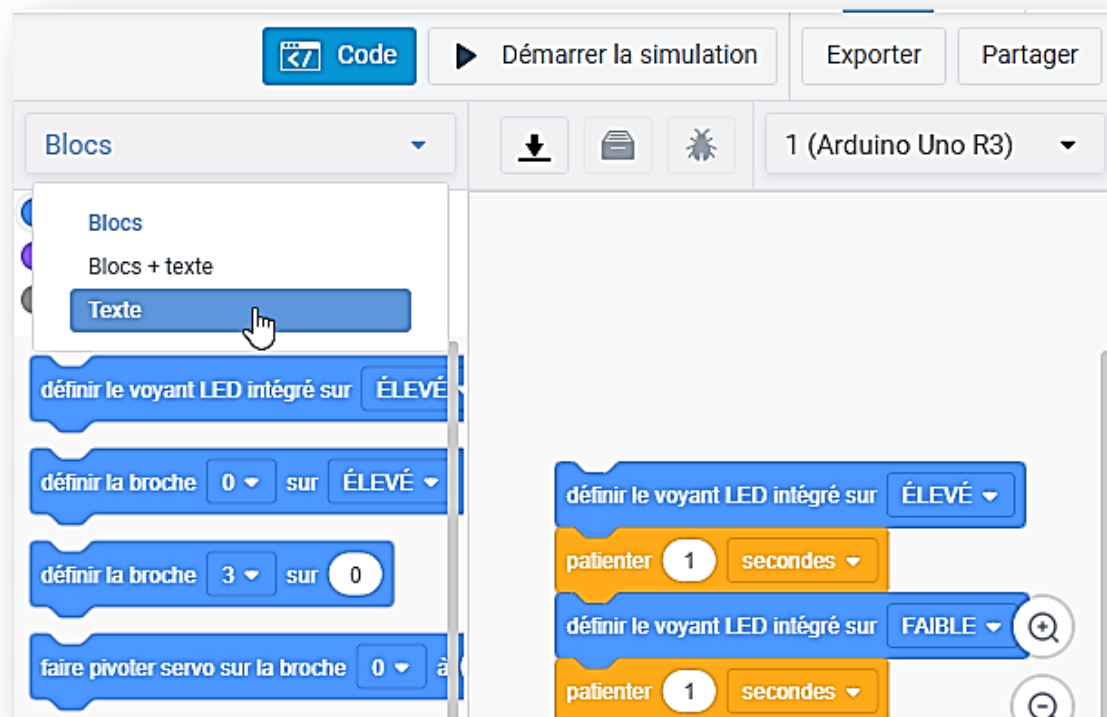


Câbler ces trois composants comme ci-contre.





Cliquer sur le bouton **Code** puis **Texte**.





Le programme par défaut est Blink.

A screenshot of the TINKER CAD web interface. The top bar contains a 'Code' button with a code icon, a 'Démarrer la simulation' button with a play icon, and an 'Exporter' button. Below this is a toolbar with a 'Texte' dropdown menu, a download icon, a save icon, and a bug icon. To the right of the toolbar, it says '1 (Arduino U'. The main area is a code editor with a line number column on the left. The code is the standard Arduino Blink program for pin 13.

```
1 void setup()  
2 {  
3   pinMode(13, OUTPUT);  
4 }  
5  
6 void loop()  
7 {  
8   digitalWrite(13, HIGH);  
9   delay(1000); // Wait for 1000 millisecond(s)  
10  digitalWrite(13, LOW);  
11  delay(1000); // Wait for 1000 millisecond(s)  
12 }
```



Le programme Prog1 doit afficher sur la console l'état de la broche 2, donc indirectement l'état du bouton.

Copier coller le  
programme  
Prog1

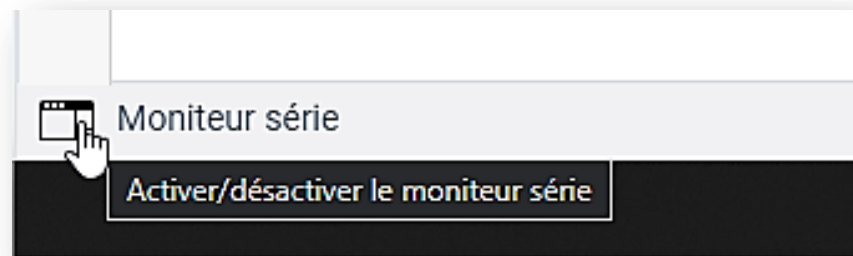
The image shows the Arduino IDE interface. At the top, there are buttons for 'Code', 'Démarrer la simulation', 'Exporter', and 'Partager'. Below these, there is a 'Texte' dropdown menu, a download icon, a file icon, a bug icon, and a dropdown menu showing '1 (Arduino Uno R3)'. The main area contains the following C++ code:

```
1 int bouton = 0; // Variable pour lire l'état du bouton
2
3 void setup()
4 {
5     pinMode(2, INPUT); // La broche 2 est une entrée
6     Serial.begin(9600); // Configurer la console
7 }
8
9 void loop()
10 {
11     bouton = digitalRead(2); // Lire l'état de la broche 2
12     Serial.println(bouton); // Afficher cet état sur la console
13 }
```



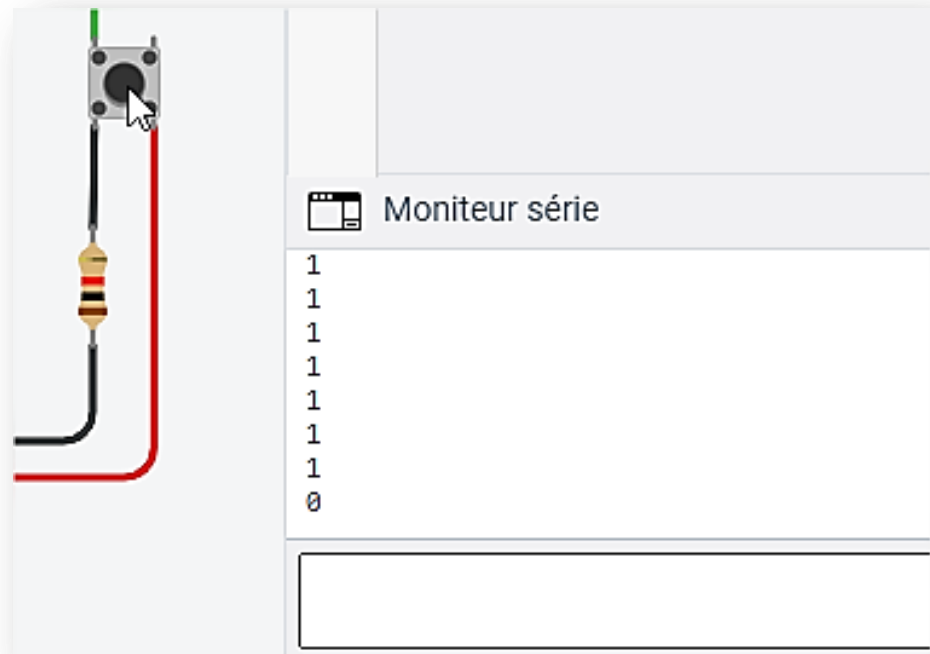


Ouvrir le  
moniteur série  
et démarrer la  
simulation.





Vérifier si le programme affiche bien l'état du bouton.





Le programme Prog2 doit afficher sur la console une valeur qui augmente de 1 à chaque action sur le bouton.

Copier coller le  
programme  
Prog2

```
1  int bouton = 0; // Variable pour lire l'état du bouton
2  int compteur = 0; // Variable pour compter
3
4  void setup()
5  {
6      pinMode(2, INPUT); // La broche 2 est une entrée
7      Serial.begin(9600); // Configurer la console
8  }
9
10 void loop()
11 {
12     Serial.println(compteur); // Afficher le compteur
13     bouton = digitalRead(2); // Lire l'état de la broche 2
14     if (bouton == 1) // Vérifier si le bouton est actionné
15     { // Si s'est le cas augmenter le compteur
16         compteur = compteur + 1;
17     }
18
19 }
```



Ce programme ne fonctionne pas car le compteur augmente trop vite. Pour résoudre ce problème, on peut ajouter une temporisation.

```
int bouton = 0; // Variable pour lire l'état du bouton
int compteur = 0; // Variable pour compter

void setup()
{
  pinMode(2, INPUT); // La broche 2 est une entrée
  Serial.begin(9600); // Configurer la console
}

void loop()
{
  Serial.println(compteur); // Afficher le compteur
  bouton = digitalRead(2); // Lire l'état de la broche 2
  if (bouton == 1) // Vérifier si le bouton est actionné
  { // Si s'est le cas augmenter le compteur
    compteur = compteur + 1;
  }
}
```



Ce programme ne fonctionne pas car le compteur augmente trop vite. Pour résoudre ce problème, on peut ajouter une temporisation.

Ajouter une temporisation de 100ms juste après avoir augmenter le compteur.

Vérifier son bon fonctionnement.

```
int bouton = 0; // Variable pour lire l'état du bouton
int compteur = 0; // Variable pour compter

void setup()
{
  pinMode(2, INPUT); // La broche 2 est une entrée
  Serial.begin(9600); // Configurer la console
}

void loop()
{
  Serial.println(compteur); // Afficher le compteur
  bouton = digitalRead(2); // Lire l'état de la broche 2
  if (bouton == 1) // Vérifier si le bouton est actionné
  { // Si s'est le cas augmenter le compteur
    compteur = compteur + 1;
    ???
  }
}
```



On souhaite maintenant utiliser un deuxième bouton pour diminuer la valeur du compteur.

Ajouter un autre bouton avec sa résistance et modifier le programme.

Vérifier son bon fonctionnement.

Compléter le document  
réponse DR1



Les valeurs du compteur doivent être comprises entre 0 et 59.

Modifier le programme pour limiter les valeurs du compteur.

Vérifier son bon fonctionnement.

Compléter le document  
réponse DR2



# Compléter le document réponse DR3

Modifier le  
programme.

Vérifier son bon  
fonctionnement.

Une autre solution consiste à utiliser un bouton pour augmenter les unités et l'autre pour augmenter les dizaines. Les boutons doivent être indépendant, c'est-à-dire que les unités ne doivent pas provoquer l'augmentation des dizaines.

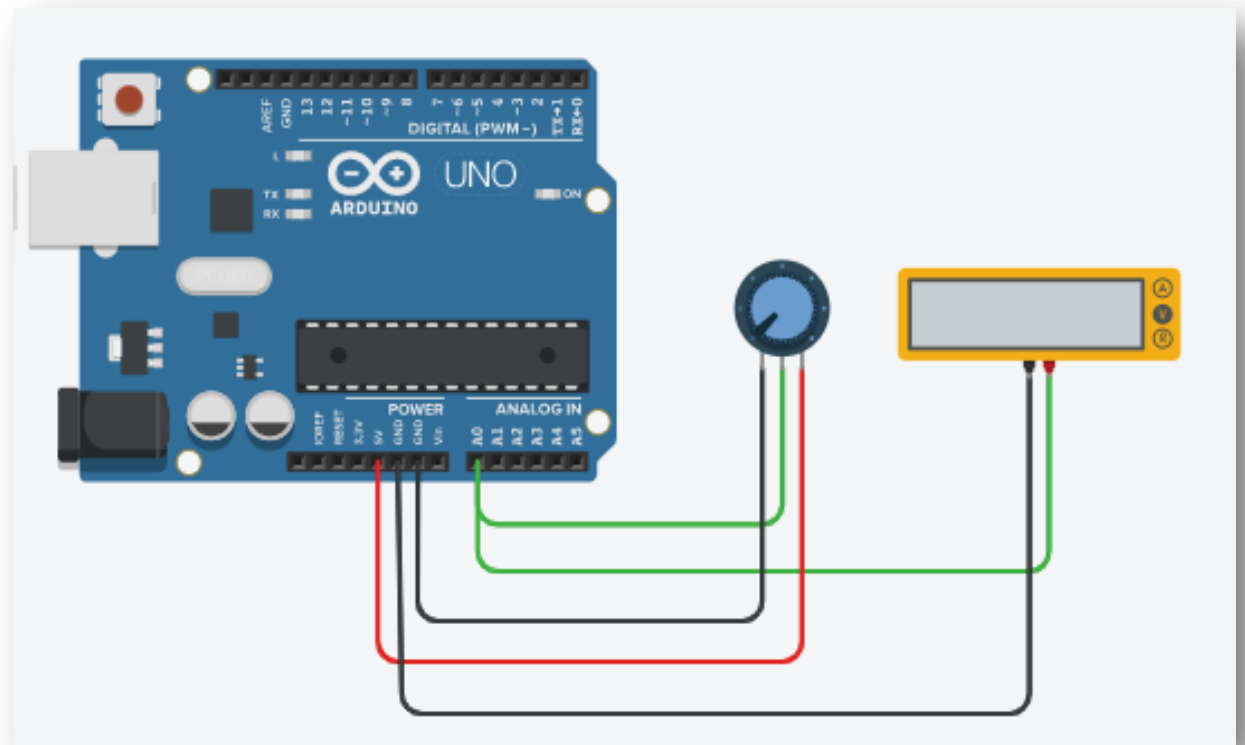
Le chiffre des unités doit repasser à 0 si il dépasse 9 et celui des dizaines repasse à 0 si il dépasse 5.





Un potentiomètre est un bouton qui une fois alimenté délivre une tension analogique. Si le bouton est alimenté en 5 V, il pourra délivrer une tension comprise entre 0 et 5V.

Réaliser le câblage suivant.





Un potentiomètre est un bouton qui une fois alimenté délivre une tension analogique. Si le bouton est alimenté en 5 V, il pourra délivrer une tension comprise entre 0 et 5V.

Copier coller le  
programme  
Prog3

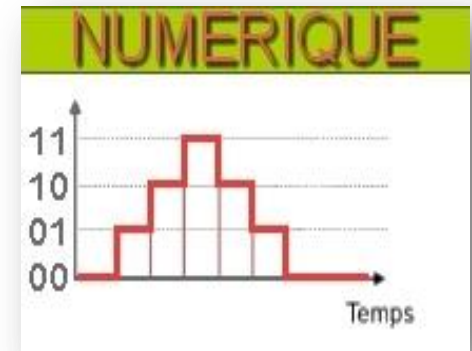
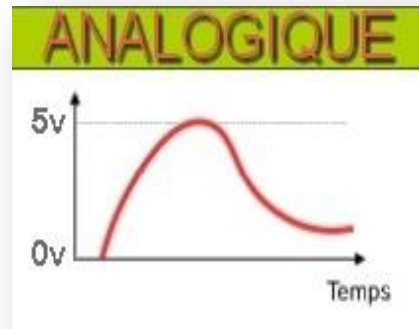
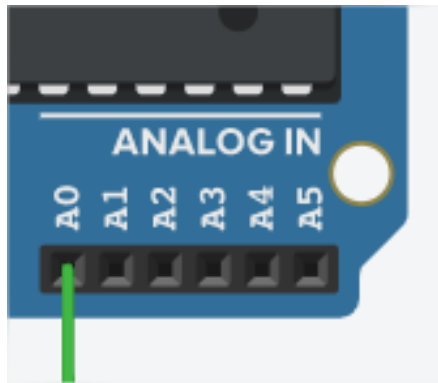
```
int capteur = 0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  capteur = analogRead(0);
  Serial.println(capteur);
}
```

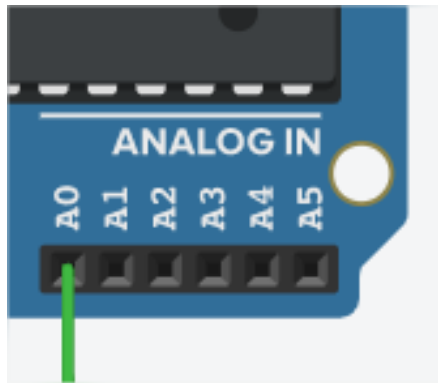


Certaines broches de la carte Arduino disposent d'un Convertisseur Analogique Numérique (CAN). Ces convertisseurs sont nécessaires car les microcontrôleur ne peuvent pas traiter les signaux analogiques.

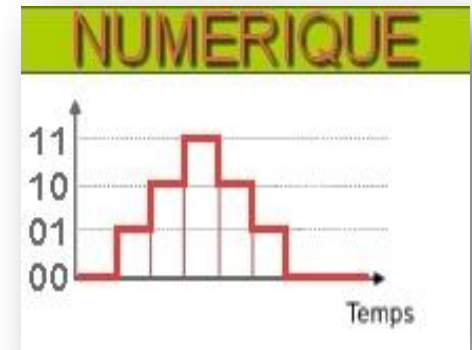
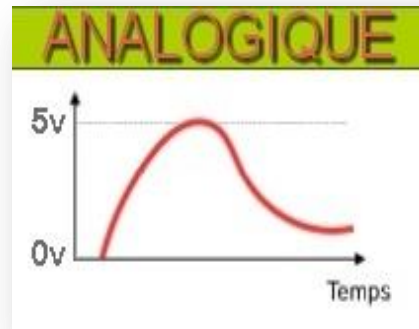




# Compléter le document réponse DR4

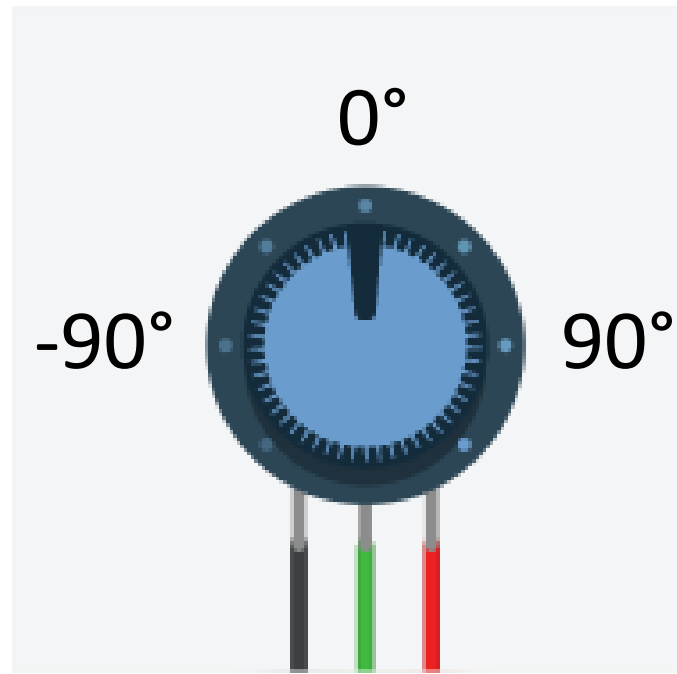


Exécuter le  
programme





Les potentiomètres peuvent être utilisés comme capteur angulaire (rapporteur). Pour cela il faut modifier le programme pour afficher un angle.





# Compléter le document réponse DR5

Modifier le  
programme  
pour afficher  
un angle.

