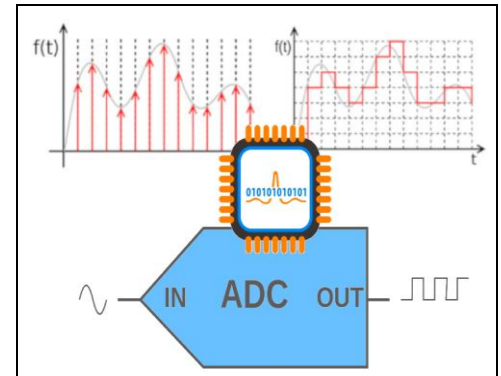


Noms : \_\_\_\_\_  
 Prénoms : \_\_\_\_\_  
 Classe : \_\_\_\_\_  
 Date : \_\_\_\_\_

**Note : /20**



## Objectif du TP

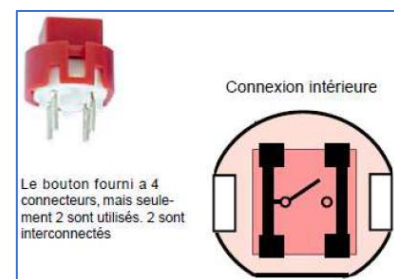
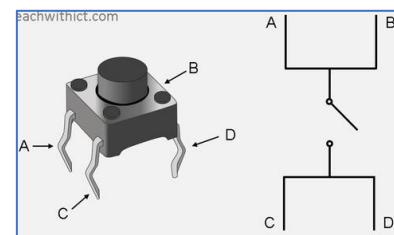
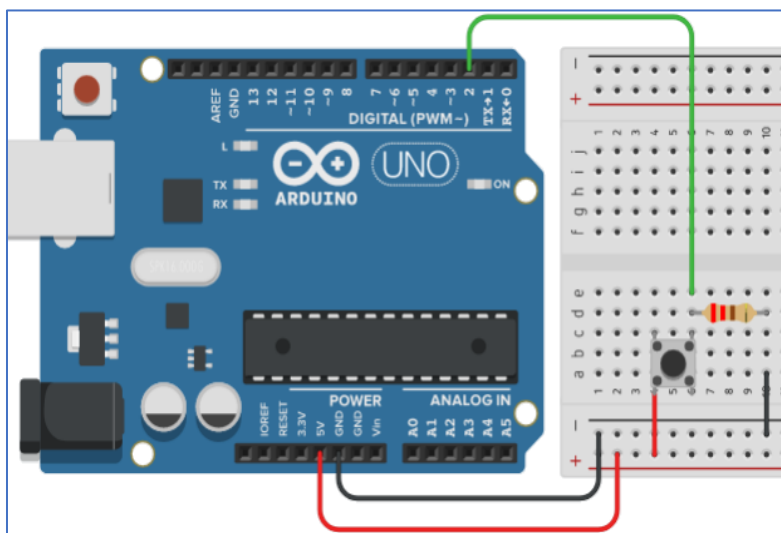
- Découvrir les bases de la programmation sur Arduino
- Découvrir la conversion analogique numérique
- Programmer un seuil de déclenchement

## Barème

Attitude, quantité de travail, qualité du compte-rendu	/2
Bouton poussoir (Q1)	/1
Potentiomètre (Q2, 3, 4 et 5)	/7
Seuils (Q6)	/3.5
Déclenchement par contact (Q7, Q8 et Q9)	/6.5

## 1. Acquisition de l'état d'un bouton poussoir


Réaliser le montage composé d'une carte Arduino, d'un bouton poussoir et d'une résistance 1kΩ.



Saisir le programme suivant :

```
1 void setup()
2 {
3   pinMode(2, INPUT);
4   Serial.begin(9600);
5
6 }
7
8 void loop()
9 {
10  Serial.println(digitalRead(2));
11  delay(10); // Delay a little bit
12 }
```

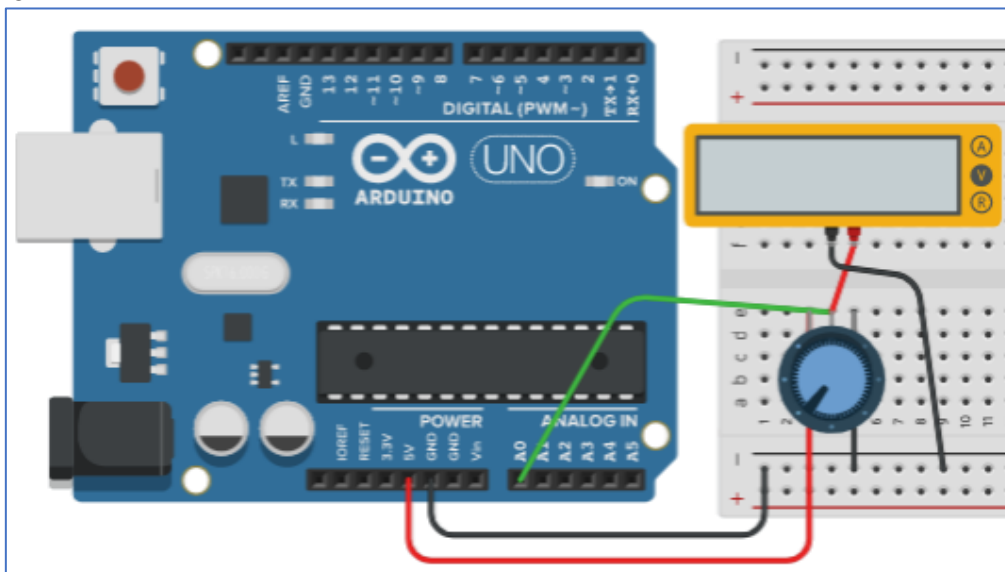
Tester le programme, soit :

- En utilisant l'IDE Arduino en double cliquant sur : , puis en téléversant le programme sur la carte
- En lançant la simulation sur le site Tinkercad : <https://www.tinkercad.com/>

**Q1** : Ouvrez le moniteur série, indiquez ce que vous y lisez et comment vous l'interprétez :

## 2. Acquisition de l'état d'un potentiomètre

Réaliser le montage suivant composé d'une carte Arduino, d'un potentiomètre 1k $\Omega$  et d'un voltmètre.



Saisissez le programme suivant :

**Q2** : Ouvrez le moniteur série et indiquez la valeur minimale et la valeur maximale que vous lisez lorsque vous tournez le potentiomètre

**Q3** : Indiquez la valeur minimale et la valeur maximale que vous lisez sur le voltmètre.

**Q4** : Sur un tableur (Excel, Calc, ou Google Sheets), pour différentes positions du potentiomètre, notez les valeurs lues sur la console et sur le Voltmètre. Vous ferez une dizaine de relevés. Tracez la courbe affichant en abscisse le mot lu sur la console et en ordonnée la tension. Que constatez-vous ?

```

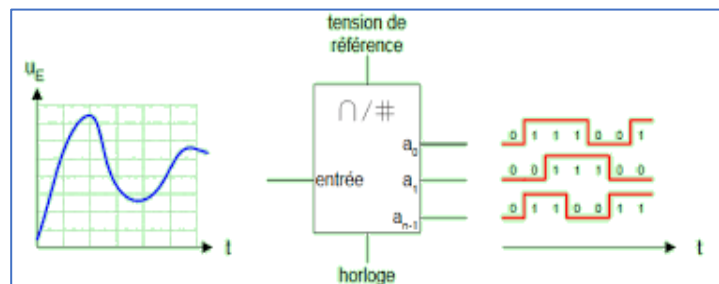
1 void setup()
2 {
3   pinMode(A0, INPUT);
4   Serial.begin(9600);
5 }
6
7
8 void loop()
9 {
10  Serial.println(analogRead(A0));
11  delay(10); // Delay a little bit
12 }

```

**Q5** : Sachant que la carte Arduino code l'entrée analogique, allant de 0 à 5V, sur 10 bits ( $2^{10} = 1024$ ), calculer la plus petite variation de tension générant un changement de mot.

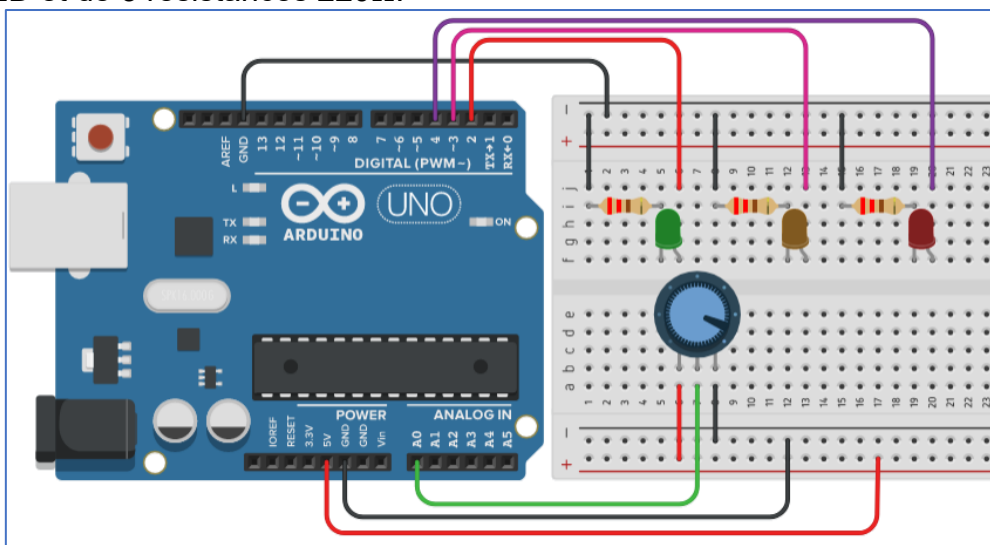
Cette valeur s'appelle le **quantum** du convertisseur analogique numérique.

$$q = \frac{\Delta V}{2^n}$$



### 3. Notion de seuils

Réaliser le montage suivant composé d'une carte Arduino, d'un potentiomètre 1k $\Omega$ , de 3 LED et de 3 résistances 220 $\Omega$ .



**Q6** : Saisissez le programme suivant sur Arduino ou sur Tinkercad, testez-le puis analysez-le pour identifier les numéros des lignes correspondant à chaque élément du tableau :

```

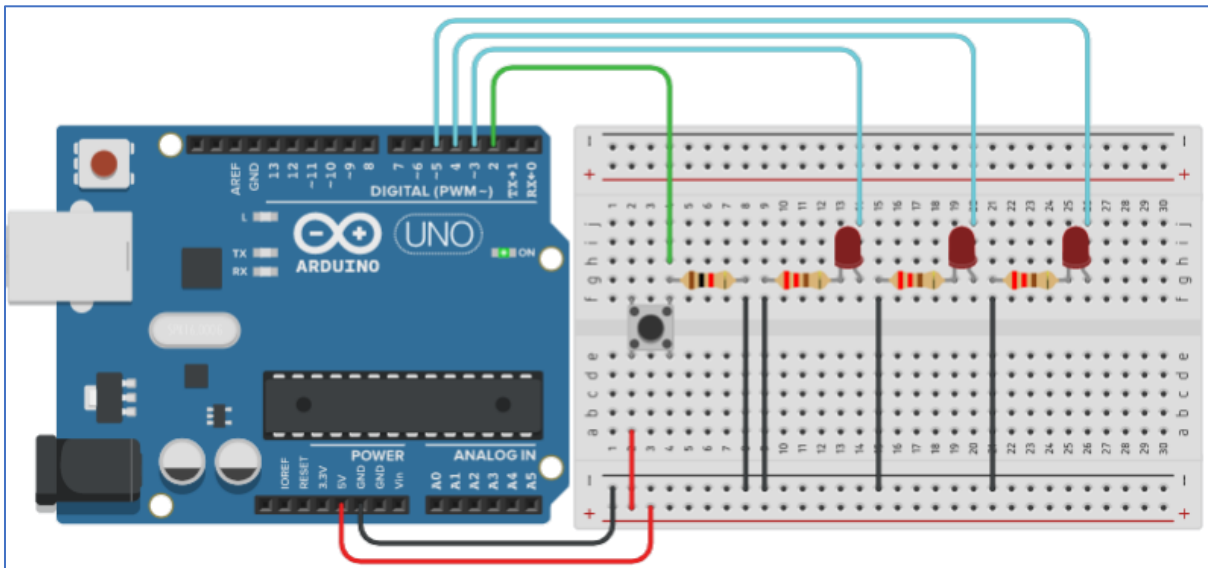
1 void setup()
2 {
3   pinMode(A0, INPUT);
4   Serial.begin(9600);
5
6   pinMode(2, OUTPUT);
7   pinMode(3, OUTPUT);
8   pinMode(4, OUTPUT);
9 }
10
11 void loop()
12 {
13   if (analogRead(A0) < 300) {
14     Serial.println(analogRead(A0));
15     digitalWrite(2, HIGH);
16     digitalWrite(3, LOW);
17     digitalWrite(4, LOW);
18     delay(1000); // Wait for 1000 millisecond(s)
19   } else {
20     if (analogRead(A0) >= 300 && analogRead(A0) < 600) {
21       Serial.println(analogRead(A0));
22       digitalWrite(3, HIGH);
23       digitalWrite(2, LOW);
24       digitalWrite(4, LOW);
25       delay(1000); // Wait for 1000 millisecond(s)
26     } else {
27       Serial.println(analogRead(A0));
28       digitalWrite(4, HIGH);
29       digitalWrite(2, LOW);
30       digitalWrite(3, LOW);
31       delay(1000); // Wait for 1000 millisecond(s)
32     }
33   }
34 }

```

N° de lignes	Éléments
	Initialisation de la vitesse de communication entre la carte et le moniteur série
	Initialisation des broches comme des entrée ou des sorties
	Attendre 1 seconde
	Allumer la diode orange et éteindre les deux autres
	Allumer la diode rouge et éteindre les deux autres
	Ecrire la valeur lue sur la broche analogique, sur le moniteur série
	Actions réalisées si la valeur sur la broche analogique est supérieure ou égale à 300 tout en étant inférieure à 600.

## 4. Déclenchement par contact

Réaliser le montage suivant composé d'une carte Arduino, d'un bouton poussoir, d'une résistance de 1k $\Omega$ , de 3 LED et de 3 résistances 220 $\Omega$ .



**Q7** : Saisissez le programme suivant sur Arduino ou sur Tinkercad, testez-le puis analysez-le pour identifier les numéros des lignes correspondant à chaque élément du tableau :

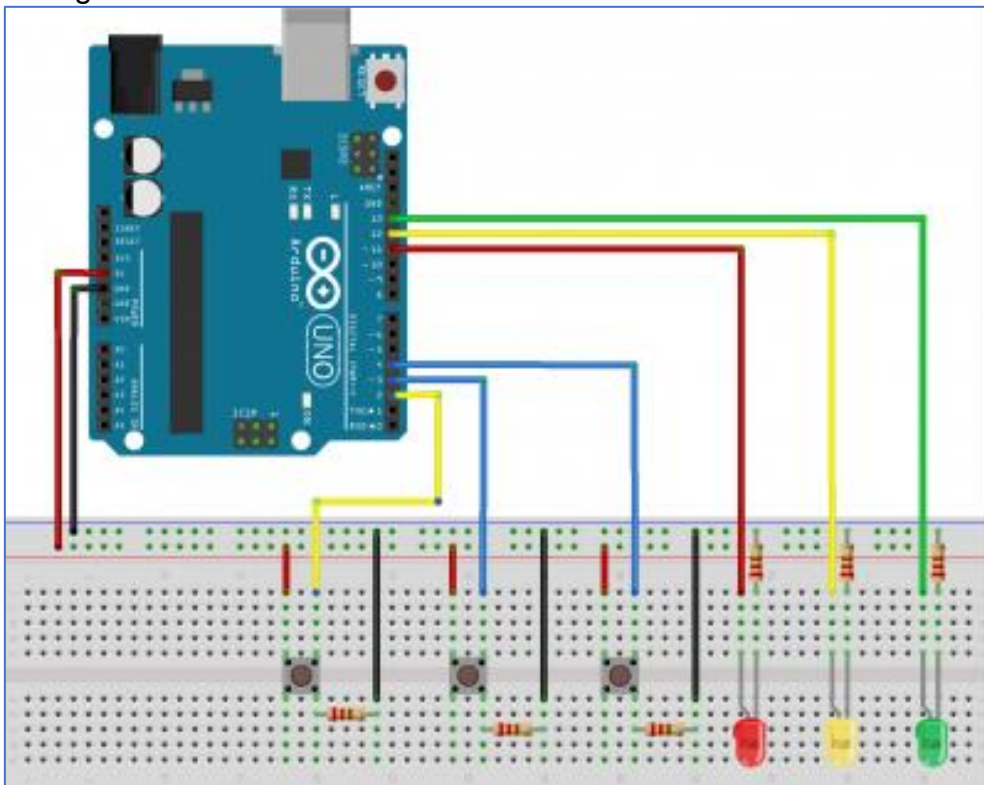
```

1 void setup()
2 {
3   pinMode(2, INPUT);
4   pinMode(4, OUTPUT);
5   pinMode(3, OUTPUT);
6   pinMode(5, OUTPUT);
7 }
8
9 void loop()
10 {
11   if (digitalRead(2) == 1) {
12     digitalWrite(4, HIGH);
13     delay(500); // Wait for 500 millisecond(s)
14     digitalWrite(4, LOW);
15     delay(500); // Wait for 500 millisecond(s)
16     digitalWrite(3, HIGH);
17     digitalWrite(5, HIGH);
18     delay(500); // Wait for 500 millisecond(s)
19     digitalWrite(3, LOW);
20     digitalWrite(5, LOW);
21     delay(500); // Wait for 500 millisecond(s)
22   } else {
23     digitalWrite(3, LOW);
24     digitalWrite(4, LOW);
25     digitalWrite(5, LOW);
26   }
27 }

```

N° de lignes	Eléments
	Initialiser les broches comme des entrée ou des sorties
	Attendre 0.5 seconde
	Tester si la broche 2 est à l'état haut
	Eteindre toutes les diodes
	Faire clignoter (=allumer puis éteindre) la diode branchée sur la broche 4
	Faire clignoter (=allumer puis éteindre) les diodes branchées sur les broches 3 et 5

**Q8** : Modifiez le programme pour changer le type de clignotement des diodes. Vous pouvez changer la couleur des diodes et/ou le nombre de diodes



**Q9** : Modifiez le programme en ajoutant, au moins 1 bouton poussoir. Décrivez par un texte le fonctionnement de votre montage.