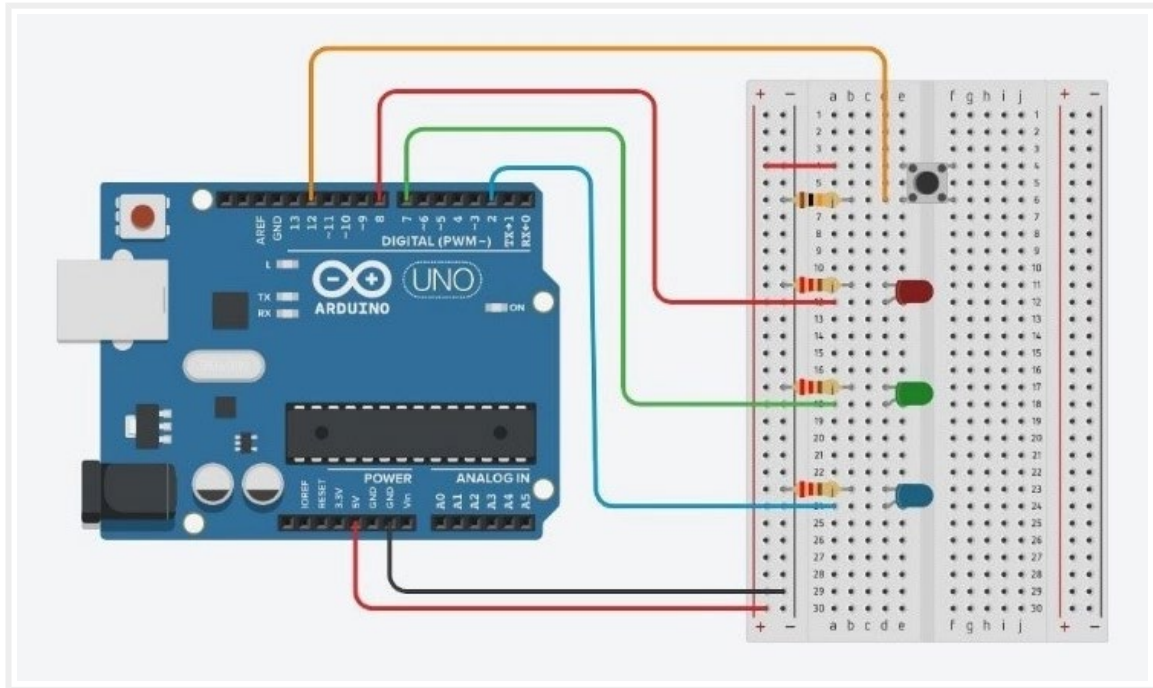


Bouton-poussoir – Alternance

(Allumer en alternance ou éteindre 3 DELs
avec un bouton-poussoir)



Liste des composants :

- . 1 DEL rouge
- . 1 DEL verte
- . 1 DEL bleue
- . 3 résistances de 220 Ω (résistances des DELs)
- . 1 bouton poussoir
- . 1 résistance de 10 k Ω (résistance du circuit du bouton poussoir)
- . 1 plaque d'essai
- . Fils de connexion

Objectif

Dans cette activité, l'allumage en alternance des DELs est géré par le bouton poussoir. Un premier appui sur le bouton allume la diode rouge, un deuxième appui allume la diode verte, un troisième appui allume la diode bleue et ainsi de suite...

Un appui prolongé sur le bouton éteint la DEL allumée.

Comme pour l'activité **Interrupteur** , c'est à l'aide des variables permettant de stocker les valeurs (actuelle et précédente) de l'état logique de la broche du bouton poussoir, mais aussi d'une variable pour compter le nombre d'appui sur le bouton et de variables pour mesurer la durée d'appui, que l'Arduino pourra allumer ou éteindre les DELs.

Le code pourra être modifié pour voir l'influence des variables (durée d'appui pour éteindre les DELS).

. Le programme

Bouton_poussoir_Alternance

```
// Déclaration des constantes et variables
```

```
const int PinLEDR = 8;
const int PinLEDV = 7;
const int PinLEDB = 2;
const int PinButton = 12;
```

```
int ValButton=0;
int OldValButton=0;
int ComptBtn=0;
unsigned long StartTime = 0;
unsigned long DeltaTime = 0;
```

```
// Initialisation des entrées et sorties
```

```
void setup() {
  pinMode (PinLEDR, OUTPUT);
  pinMode (PinLEDV, OUTPUT);
  pinMode (PinLEDB, OUTPUT);
  pinMode (PinButton, INPUT);
}
```

```

// Fonction principale en boucle

void loop() {
    ValButton = digitalRead(PinButton);
    delay(10);
    if ((ValButton == HIGH) && (OldValButton == LOW)) {
        StartTime = millis();
        ComptBtn = ComptBtn +1;
        if (ComptBtn == 4) {
            ComptBtn = 1;
        }
    }

    if ((ValButton == HIGH) && (OldValButton == HIGH)) {
        DeltaTime = millis() - StartTime;
        if (DeltaTime > 500) {
            ComptBtn = 0;
        }
    }

    OldValButton = ValButton;

    switch (ComptBtn) {
        case 1 :
            digitalWrite(PinLEDR, HIGH);
            digitalWrite(PinLEDV, LOW);
            digitalWrite(PinLEDB, LOW);
            break;

        case 2 :
            digitalWrite(PinLEDR, LOW);
            digitalWrite(PinLEDV, HIGH);
            digitalWrite(PinLEDB, LOW);
            break;

        case 3 :
            digitalWrite(PinLEDR, LOW);
            digitalWrite(PinLEDV, LOW);
            digitalWrite(PinLEDB, HIGH);
            break;

        case 0 :
            digitalWrite(PinLEDR, LOW);
            digitalWrite(PinLEDV, LOW);
            digitalWrite(PinLEDB, LOW);
            break;
    }
}

```

Déroulement du programme :

Déclaration des constantes et variables

- N° de la broche correspondant à la DEL rouge: **const int PinLEDR = 8**
- N° de la broche correspondant à la DEL verte: **const int PinLEDV = 7**
- N° de la broche correspondant à la DEL bleue: **const int PinLEDB = 2**
- N° de la broche correspondant au bouton poussoir: **const int PinButton = 12**
- Variable pour stocker la valeur de la broche du bouton poussoir: **int ValButton = 0**
- Variable pour stocker l'ancienne valeur de la broche du bouton poussoir: **int OldValButton = 0**
- Variable pour compter le nombre de fois que le bouton poussoir est appuyé: **int ComptBtn=0**
- Variables pour mesurer le temps d'appui sur le bouton poussoir: **unsigned long StartTime = 0**
unsigned long DeltaTime = 0

void setup()

initialisation des entrées et sorties

- Les broches des DELs sont initialisées comme des sorties digitales. Des données seront donc envoyées depuis le microcontrôleur vers ces broches :

Exemple: **pinMode (PinLEDR, OUTPUT)**

- La broche du bouton poussoir est initialisée comme une entrée digitale. Des données seront donc envoyées depuis cette broche vers le microcontrôleur :

pinMode (PinButton, INPUT)

void loop()

Fonction principale en boucle

Lecture de l'état logique de la broche du bouton poussoir:

ValButton = digitalRead(PinButton)

Attente (10 ms) pour éviter les perturbations dues au phénomène de rebond du bouton poussoir:

delay(10)

bouton poussoir appuyé (broche à +5V):
ValButton = HIGH
et avant relâché (broche à 0V):
OldValButton = LOW

?

Oui

- Incrémement du compteur d'appui sur le bouton poussoir:
ComptBtn = ComptBtn +1
(Si le compteur = 4 alors compteur = 1)

- Mise à jour de la variable pour stocker le debut d'appui sur le bouton poussoir:
StartTime = millis()

bouton poussoir maintenu appuyé
(broche à +5V):
ValButton = HIGH
et **OldValButton = HIGH**

?

Oui

- Mesure du temps d'appui sur le bouton poussoir:
DeltaTime = millis() - StartTime

- Si DeltaTime > 500 ms, le compteur d'appui sur le bouton poussoir est réinitialisé afin d'éteindre toutes les DELs:
ComptBtn = 0

Stockage de l'état logique de la broche du bouton poussoir lu:
OldValButton = ValButton

En fonction de la valeur du compteur et du temps d'appui sur le bouton poussoir,
une DEL est allumée ou toutes les DELs sont éteintes:

switch (ComptBtn)

ComptBtn = 0

Eteindre toutes les DELs
digitalWrite(PinLedR, LOW)
digitalWrite(PinLedV, LOW)
digitalWrite(PinLedB, LOW)

ComptBtn = 1

Allumer la DEL rouge et éteindre les autres DELs
digitalWrite(PinLedR, HIGH)
digitalWrite(PinLedV, LOW)
digitalWrite(PinLedB, LOW)

ComptBtn = 2

Allumer la DEL verte et éteindre les autres DELs
digitalWrite(PinLedR, LOW)
digitalWrite(PinLedV, HIGH)
digitalWrite(PinLedB, LOW)

ComptBtn = 3

Allumer la DEL bleue et éteindre les autres DELs
digitalWrite(PinLedR, LOW)
digitalWrite(PinLedV, LOW)
digitalWrite(PinLedB, HIGH)