

Utilisation de la fonction milli pour réaliser un Delay NON BLOQUANT

Objectif : Réaliser un code permettant le fonctionnement suivant : La Led rouge clignote sur le rythme suivant (30 secondes allumée puis 30 secondes éteinte)

Pendant les 30 sec ou la rouge est allumée la verte clignote 5 fois (donc 6 sec allumée puis 6 sec éteinte).

On pourra s'inspirer des codes présentés sur ce site :

<https://www.carnetdumaker.net/articles/faire-plusieurs-choses-la-fois-avec-une-carte-arduino/>

Correction :

```
// Déclare les broches sur lesquelles sont câblées les LEDs
const int BROCHE_LED_1 = 13; // La rouge
const int BROCHE_LED_3 = 11; //la verte

// Nombre de millisecondes entre deux changements d'état des LED
const unsigned long BLINK_INTERVAL_1 = 30000;
const unsigned long BLINK_INTERVAL_3 = 6000;

// Précédente valeur de millis() pour la LED 1, 2 et 3
unsigned long previousMillisLed1 = 0;
unsigned long previousMillisLed3 = 0;

// Précédent état de la LED 1, 2 et 3
byte etatBrocheLed1 = LOW;
byte etatBrocheLed3 = LOW;

// Fonction setup(), appelée au démarrage de la carte Arduino
void setup() {

    // Configure les broches des LEDs en sortie
    pinMode(BROCHE_LED_1, OUTPUT);
    pinMode(BROCHE_LED_3, OUTPUT);

    // Configure l'état initial des LEDs
    digitalWrite(BROCHE_LED_1, etatBrocheLed1);
    digitalWrite(BROCHE_LED_3, etatBrocheLed3);
}
```

```
void loop() {

    // Récupère la valeur actuelle de millis()
    unsigned long currentMillis = millis();

    // Si BLINK_INTERVAL_1 ou plus millisecondes se sont écoulés
    if(currentMillis - previousMillisLed1 >= BLINK_INTERVAL_1) {

        // Garde en mémoire la valeur actuelle de millis()
        previousMillisLed1 = currentMillis;

        // Inverse l'état de la LED 1
        etatBrocheLed1 = !etatBrocheLed1;
        digitalWrite(BROCHE_LED_1, etatBrocheLed1);
    }

    // Si BLINK_INTERVAL_3 ou plus millisecondes se sont écoulés
    if ((etatBrocheLed1==HIGH) && (currentMillis - previousMillisLed3 >= BLINK_INTERVAL_3)) {

        // Garde en mémoire la valeur actuelle de millis()
        previousMillisLed3 = currentMillis;

        // Inverse l'état de la LED 3
        etatBrocheLed3 = !etatBrocheLed3;
        digitalWrite(BROCHE_LED_3, etatBrocheLed3);
    }
}
```