

# Prenons notre temps avec Arduino

## Matériel nécessaire

- 1 diode (récepteur) infra-rouge 3 broches 38 kHz .
- 1 petite télécommande IR standard à 21 boutons.
- 1 carte Arduino Uno
- 3 fils de connexion



## Approche simple de la gestion du temps :

Ouvrir le programme « telecommande-del13.ino»

Modifier le programme pour que lorsqu'on appuie sur le bouton 1 de la télécommande, la DEL s'allume pendant 10s puis s'éteigne.

Aide : utiliser la fonction delay(durée) //la durée est en ms

Réponse :

```
void loop() {
  if (irrecv.getResults()) { //reception du signal infra rouge
    results.decode();        //decodage du signal infrarouge
    if (results.value==0xFD08F7){ // si on appuie sur la touche 1
      Serial.println(results.value, HEX);
      allume_del();
      delay(10000);
      eteint_del()
    }
    if (results.value==0xFD8877){ // si on appuie sur la touche 2
      eteint_del()
    }
  }
  irrecv.enableIRIn(); //initialise le recepteur
}

void allume_del(){
  digitalWrite(ledPin, HIGH); // On allume la LED
}

void eteint_del(){
  digitalWrite(ledPin, LOW); // On eteint la LED
}
```

Faire fonctionner le programme. (telecommande-del13-delay.ino)

Test : on allume la DEL avec la touche « 1 » de la télécommande.

Que se passe-t-il si on appuie sur la touche « 2 » de la télécommande au bout de 3 ou 4 secondes ?

Réponse : le microcontrôleur ne répond pas à la commande. Il faut attendre 10 s pour qu'il accepte la requête

En conclusion, la fonction delay() est une fonction bloquante.

Ce type de programmation n'est pas pratique si de nombreuses tâches nécessitent une lecture fréquente des broches en entrée.

Il faut pouvoir continuer à surveiller les autres entrées tout en exécutant la tâche.

## Une gestion du temps plus souple :

On utilise la fonction `millis()` qui renvoie une date qui correspond au nombre de millisecondes qui se sont écoulées depuis le lancement du programme ou depuis le dernier reset. Cette fonction n'est pas bloquante.

Pour connaître la durée en temps réel de l'allumage de la DEL, il suffit dans un premier temps, de mémoriser la date d'allumage de la DEL tel que `date_allumage=millis()` au moment où elle s'allume. Dans `Loop()`, on peut surveiller la durée d'allumage de la led en utilisant la fonction : `duree= millis()- date_allumage`. Il suffit de faire quelques tests sur la variable `duree` pour commander la DEL : Si `duree < 10000` alors allumer la DEL sinon l'éteindre.

Modifier le code précédent comme suit:

```
long duree, date_allumage ;
boolean b_del = false;

void loop() {
    if (irrecv.getResults()) { //reception du signal infra rouge
        results.decode();      //decodage du signal infrarouge
        if (results.value==0xFD08F7) { //si on appuie sur la touche 2
            if (b_del==false) {
                date_allumage = millis(); //initialisation de la date d'allumage
                b_del=true ;
            }
        }
        if (results.value==0xFD8877) { //si on appuie sur la touche 2
            eteint_del();
        }
        Delay(50); //évite si l'appui est trop long, l'affichage de #FFFFFF
        irrecv.enableIRIn(); //reinitialise le recepteur infra rouge
    }
    if (b_del==True) {
        allume_dell();
        duree=millis()-date_allumage ;
        if (duree > 10000) eteint_del ;
    }
}

void allume_del(){
    digitalWrite(ledPin, HIGH); // On allume la LED
}

void eteint_del(){
    digitalWrite(ledPin, LOW); // On eteint la LED
    duree=0; //reinitialise la duree à 0
    b_del=false;
}
```

Programme : [telecommande-del13-millis.ino](#)

**Test :** on allume la DEL avec la touche « 1 » de la télécommande.

Que se passe-t-il maintenant si on appuie sur la touche « 2 » de la télécommande au bout de 3 ou 4 secondes ?

Conclure :

L'utilisation de la fonction `millis()` et l'ajout de quelques variables dans le programme permet de mieux gérer l'état des entrées en temps réel.

Remarque : on pourrait de façon analogue utiliser la fonction `micros()` qui renvoie la date en microsecondes.