Fiche tutoriel 8
Introduction programmation arduino
Exemple : allumer la LED interne et et une LED externe

Comme nous l'avons vu dans la fiche pratique 1 : les diodes TX et RX clignotent quand l'arduino reçoit ou envoie des informations. Il existe également une diode « **L »** qui clignote si on appuie sur le bouton reset, sinon elle ne fait rien tant qu'on ne lui a pas dit.

Nous allons allumer le LED « **L** » grâce à notre programme. La diode s'allumera quand elle sera parcourue par un courant.

Programmation de la LED L

La force de l'Arduino est d'envoyer du courant, ou non, par les connexions numérotées de 0 à 13. La diode « **L** » s'allume quand on dit à l'Arduino d'envoyer du courant dans la connexion 13
Vous verrez que ce petit programme peut servir dans certains cas pour nous aider à débugger.

- Ouvrer le programme Arduino
- Configurer votre carte et le port (cf fiche 1)

Tapez le programme suivant

```
1  void setup()
2  {
3    pinMode(13,OUTPUT);
4    digitalWrite(13,HIGH);
5  }
6
7  void loop()
8  {
9
10 }
```

Pour que cela fonctionne n'oubliez pas de téléverser le programme en appuyant sur la touche

Les étapes du programme sont assez simples :

1. On dit à l'Arduino que nous voulons que la connexion 13 puisse envoyer du courant et pas en recevoir (instruction OUTPUT).
2. On dit à l'Arduino d'envoyer du courant dans la connexion 13(instruction HIGH).

Ici apparait de nouveau mots-clés : **pinMode(), digitalWrite(), OUTPUT, HIGH**.

pinMode (broche, mode) : Configure la broche spécifiée pour qu'elle se comporte comme une entrée ou une sortie avec : ***broche*** : numéro de la pin ***mode* :** INPUT ou OUTPUT

OUTPUT : Mot clé qui configure la pin spécifieé comme une sortie. C'est-à-dire que l'Arduino enverra du courant vers la pin spécifiée.

INPUT : Mot clé qui configure la pin spécifiée comme une entrée. C'est-à-dire que l'Arduino recevra du courant vers la pin spécifiée.

digitalWrite() : Envoie ou écris la valeur digitale du composant situé sur la pin entre parenthèse avec HIGH pour alimenter en électricité et LOW pour couper l'électricité.

Nous ne verrons pas en détails toutes les fonctions existantes pour Arduino. Si vous le souhaitez reporter-vous sur le lien suivant qui décrit toutes les fonctions :

https://www.arduino.cc/reference/en/

Les mots clés en bleu dans le logiciel, sont spéciaux. Ils s'écrivent en majuscules. Dans notre cas, le mot clé **HIGH** signifie "valeur haute". Il existe son contraire : **LOW** qui signifie "valeur basse". On pourrait aussi remplacer et dire que HIGH vaut 1 et LOW vaut 0, ou encore que HIGH veut dire que la connexion est à +5V et LOW, qu'elle est à est à 0V.
Le mot **OUTPUT** signifie que c'est une sortie, l'Arduino enverra donc du 5V a la pin. Son contraire INPUT signifie que la pin est une entrée et l'Arduino attendra un courant d'entrée (attention l'Arduino ne pourra recevoir que du 5V cc).

Ce petit programme allume donc la LED « **L** » mais ne l'éteint pas. Si l'on veut éteindre la LED il faudrait rajouter dans le programme l'instruction suivante :

digitalWrite(13,LOW) ;

Cependant la LED resterai éteinte. Le programme tournant en boucle. Nous allons donc modifier notre programme pour faire « blinker « notre LED.

Programme blink de la LED L

Notre programme doit donc comporter 4 points :

- Indiquez à l'arduino que la pin 13 doit envoyer du courant
- Envoyer du courant dans le pin 13 (on allume)
- Arrêtez le courant dans la pin 13 (on éteint)
- Recommencez au point 2 jusqu'à l'infini.

Cependant jusqu'à maintenant, nos instructions ont été mises dans les accolades de "setup". Nous dirons qu'elles sont dans le setup. Ces instructions ne sont lues et activées qu'**une seule fois.**
Les instructions mises dans les accolades de "loop" seront lues et activées **à l'infini.**
Le programme sera donc :

```
void setup() {
  // put your setup code here, to run once:
pinMode(13,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
digitalWrite(13,HIGH);
digitalWrite(13,LOW);
}
```

```
int Led=13;         // on déclare une variable diode=13

void setup() {
  // put your setup code here, to run once:
pinMode(Led,OUTPUT); // defini la pin 13 de la diode en sortie
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(Led,HIGH);} // on allume la diode
  digitalWrite(Led,LOW); // on éteint la diode
}
```

Les deux programmes sont identiques on a seulement déclaré la LED dans une variable Diode

Cependant même si le programme est bon la diode reste allumée.
En fait le microprocesseur de l'Arduino est cadencé à 16 MHz, soit 16 000 000 actions machine par seconde. Dans le cas de notre programme la LED clignote donc 77 000 fois par seconde, ce qui est trop rapide pour la perception humaine. Il faut donc ralentir l'Arduino. Nous allons donc utiliser un nouveau mot clé : **delay()**
Qui indique à l'Arduino d'attendre du temps situé dans la parenthèse (exprimé en ms).
Ainsi delay (500) met en pause le programme 500 ms.

Le programme devient donc

```
1  void setup() {
2    // put your setup code here, to run once:
3  pinMode(13,OUTPUT);
4  }
5
6  void loop() {
7    // put your main code here, to run repeatedly:
8  digitalWrite(13,HIGH);
9  delay(500);
10 digitalWrite(13,LOW);
11 delay(500);
12 }
```

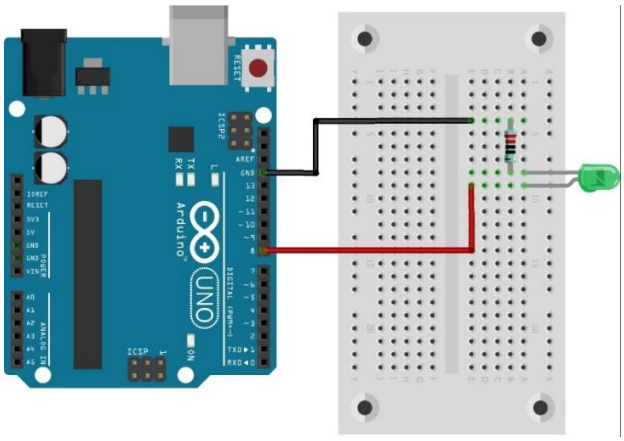
Tout fonctionne. Cependant en fonction de la complexité de vos programmes dans le futur, il est nécessaire de commentez votre programme pour une meilleure visibilité et lecture. Cela peut être fait avec **//**. Toutes les instructions après le // sont considéré comme des commentaires.
Notre programme deviendrez donc :

```
1  void setup() {
2    // put your setup code here, to run once:
3  pinMode(13,OUTPUT); // defini la pin 13 en sortie
4  }
5
6  void loop() {
7    // put your main code here, to run repeatedly:
8  digitalWrite(13,HIGH); // allume la diode 13 HIGH courant
9  delay(500);// attendre 500 ms
10 digitalWrite(13,LOW); // éteint la diode 13 LOW courant
11 delay(500); // attend 500 ms
12 }
```

Il existe dans le logiciel Arduino de nombreux exemples de programmation allant du plus simple au plus complexe. Pour cela vous pouvez ouvrir un exemple en cliquant sur :
Fichier->Exemples
Chaque programme vous montre un exemple de code et contient de plus un lien sur le montage correspondant.

Vous pouvez Réutiliser les codes ci-dessus pour pouvoir allumer uns LED externe branchée sur la pin 13 ou autre (il suffira de remplacer dans le code le chiffre 13 par le chiffre de la pin que vous avez utilisé). Le circuit de montage est assez simple. Il faut juste tenir compte de la polarité de la LED (borne **+ anode** qui est la plus longe et légèrement courbée et la borne **- cathode** la plus courte)

Pour cela nous allons donc brancher la LED sur la pin de votre choix par sa borne plus et relié le négatif au GND de l'Arduino. Nous allons cependant rajouter une **résistance** de 220 Ω. Pour éviter de griller la LED et utiliser une BreadBoard. Le montage est le suivant :



Vous pouvez maintenant tenter de faire l'exercice de la fiche travaux pratique 1.