




LE BLINK

Voici le premier programme que nous allons expérimenter ensemble ! Cette suite d'instructions va faire clignoter une LED branchée sur la broche 13 de l'Arduino toutes les secondes.

Codecast

Bienvenue dans ce premier Codecast appelé « Blink » ou clignotement en français. Pour lancer le Codecast, appuyez sur le bouton lecture  et laissez vous guider ! N'oubliez pas que vous pouvez appuyer sur le bouton pause  à tout moment pour arrêter le Codecast et revenir en arrière avec la barre de temps bleue  Attention, c'est à vous !

Si vous rencontrez des difficultés d'affichage du Codecast, cliquez [ici](#) pour l'afficher en plein écran ! Vous pouvez télécharger la transcription du Codecast [ici](#) !

Arduino ou simulateur

Lorsque vous utilisez le logiciel Arduino, le programme Blink peut être trouvé en cliquant sur *Fichier*→*Exemples*→*01.Basics*→*Blink*. Pour le simulateur, vous pouvez copier-coller le code présent ci-dessous :

```
/*  
  
Clignotement  
  
Allume la LED 13 pendant 1 seconde,  
  
puis l'éteint pendant 1 seconde.  
  
*/  
  
// On stocke la valeur 13 dans la variable "led", pour indiquer que la led se trouve sur la broche 13  
// La variable "led" est un nombre entier (integer in english)  
int led = 13;  
  
// le code dans cette fonction est exécuté une fois au début  
void setup() {  
  
// indique que la broche de la LED 13 est une sortie :  
pinMode(led, OUTPUT);  
}
```

```
// le code dans cette fonction est exécuté en boucle
void loop() {
  digitalWrite(led, HIGH); // allumer la LED (tension 5V sur la broche)
  delay(1000);             // attendre 1000ms = 1s
  digitalWrite(led, LOW);  // éteindre la LED (tension 0V sur la broche)
  delay(1000);             // attendre à nouveau 1 seconde
}
```

Montage

Plusieurs personnes sur le forum nous ont indiqué qu'elles aimeraient avoir le montage à réaliser sur cette page. Le voici :



N'oubliez pas que les LEDs ont un sens donc si votre Arduino contient le programme `Blink` et que la LED qui est branchée comme sur le schéma ci-

dessus **ne s'allume pas**, alors vérifiez que la patte - (**moins**) est bien connectée au **GND** (ground = masse en anglais) de votre Arduino. Moyen mnémotechnique, la patte la **moins** longue est la patte - (**moins**). L'autre patte + (**plus**) devra être connectée à la broche 13 qui reçoit le courant :

Instructions

À la fin de chaque programme, nous détaillerons les nouvelles briques logicielles utilisées. Comme c'est notre premier programme, nous avons beaucoup de choses à voir (n'hésitez pas à cliquer sur les liens ci-dessous afin d'arriver sur [la référence Arduino](#)).

Dans ce programme, nous avons :

- **Des [commentaires](#)** : qui sont des lignes de texte incluses dans le programme et qui ont pour but de vous aider à comprendre (ou à vous rappeler) comment votre programme fonctionne ou d'en informer les autres. Ces lignes ne sont pas envoyées à Arduino. Il y a deux façons de créer des lignes de [commentaires](#) :

```
/*  
  
Voici des  
  
commentaires sur  
  
plusieurs ligne  
  
*/  
  
// Ceci est également un commentaire
```

- Des instructions :
- **Déclaration d'une [variable](#)** : on vient avec cette ligne stocker la valeur à droite du signe égal dans la [variable](#) à gauche du signe égal.

```
int led = 13;
```

Dans notre cas, cela signifie que la **variable** appelée `led` qui sera un nombre (puisque elle est précédée du mot clé `int` pour integer = nombre entier en anglais) viendra prendre la valeur 13.

- **Les blocs d'instructions** : **setup** (mise en place) regroupe toutes les instructions qui seront exécutées au démarrage du programme. La fonction **setup** n'est exécutée qu'une seule fois, après chaque mise sous tension ou reset (réinitialisation) de la carte Arduino. **loop** (boucle en anglais) contient les instructions que l'on souhaite voir exécuter encore et encore tant que l'Arduino est branché.



Les 2 blocs d'instructions **setup** et **loop** sont obligatoires dans tout programme Arduino, même si ces blocs sont vides (ie, même si ces blocs ne contiennent aucune instruction).

```
void setup() { }  
void loop() { }
```

- Les **fonctions** : sont des instructions qui permettent d'exécuter une ou plusieurs actions. Les fonctions sont définies avec :
 - **Un nom** : le nom de la fonction.
 - **Une ou des entrées** : ce sont des variables passées à la fonction appelées **paramètres** ou **arguments**. Ces arguments sont placés entre parenthèses.
 - **Une sortie** : le résultat de la fonction qui peut être stocké dans une variable.

Prenons l'exemple de la fonction suivante :

```
digitalWrite(led, HIGH);
```

Dans ce cas, le nom de la fonction est `digitalWrite`. Nous passons deux paramètres à la fonction : `led` et `HIGH`. La fonction `digitalWrite` n'a pas de sortie. Avec cette fonction, nous allumons la broche située sur la broche passée avec le premier paramètre (qui peut être un nombre ou une

variable). Lorsque le second argument est placé à **HIGH**, on vient allumer la LED. Tandis qu'on éteindra la LED si le second argument passé est **LOW**.

Les autres fonctions présentes dans le programme *Blink* sont :

- **pinMode** configure la broche spécifiée dans le premier paramètre pour qu'elle se comporte soit en entrée (**INPUT**), soit en sortie (**OUTPUT**) passée avec le second paramètre :

```
pinMode(led, OUTPUT);
```

- **delay** fait une pause dans l'exécution du programme pour la durée (en millisecondes) passée en paramètre :

```
delay(1000);
```

Voilà, c'est tout pour le code de cette semaine, la suite c'est les quiz et le premier TP.

Références

- [Référence mini du langage Arduino](#) par Xavier Hinault
- [Référence officielle du langage Arduino \(anglais\)](#) par l'équipe d'Arduino

Licence

Nous profitons de cette section pour remercier Xavier Hinault qui met à disposition sur son site www.mon-club-elec.fr une documentation détaillée et libre sur laquelle s'appuie ce texte.