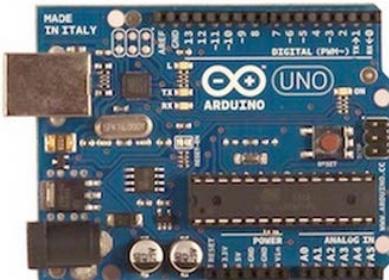


# Introduction à la programmation Arduino

## Bandeau lumineux de K2000 !

### 1. Introduction



La carte Arduino Uno est une carte à microcontrôleur basée sur l'ATmega328.

Elle dispose :

- ✚ de 14 broches numériques d'entrées/sorties (dont 6 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée)),
- ✚ de 6 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques),
- ✚ d'un quartz 16Mhz,
- ✚ d'une connexion USB,
- ✚ d'un connecteur d'alimentation jack,
- ✚ d'un connecteur ICSP (programmation "in-circuit"),
- ✚ et d'un bouton de réinitialisation (reset).

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur; Pour pouvoir l'utiliser et se lancer, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB).

La programmation de la carte s'effectue avec le logiciel « **Arduino 1.0.5** ».

### 2. Matériel disponible.

Vous disposez de :

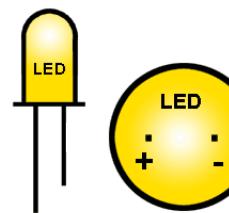
- ✚ Une carte arduino uno
- ✚ 5 leds
- ✚ 5 resistances
- ✚ Des fils
- ✚ Une platine Labdec
- ✚ Un potentiomètre

### 3. Montage électronique

Q1. En effectuant une recherche internet (par exemple sur ce site : <http://www.interfacez.fr/conseils/nomenclature3.htm#diodes>), compléter le schéma suivant en indiquant la polarité et le nom des deux entrées d'une led.

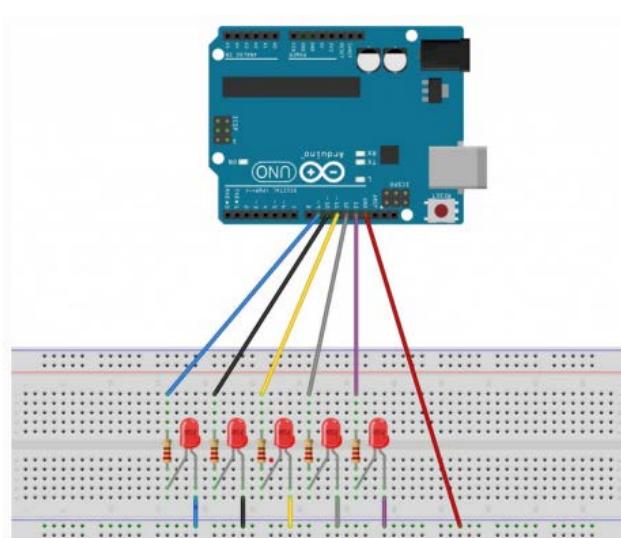


Représentation normalisée

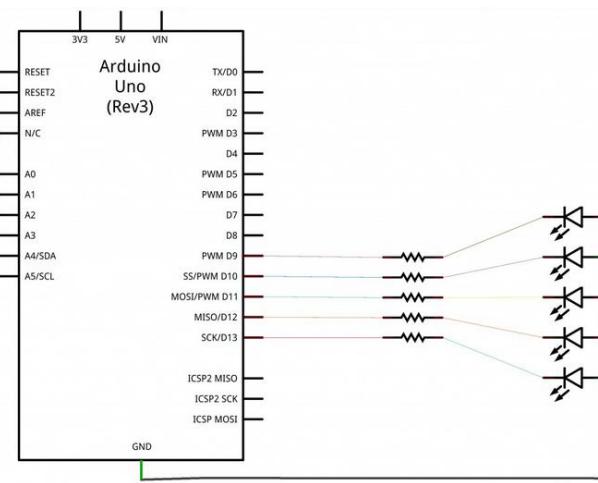


Représentation schématique

Q2. Réaliser le branchement à partir du schéma fourni: Vous devez monter en série les leds et les résistances.



sens..



Attention les leds sont polarisées !! respecter le

Remarque les résistances ne sont pas polarisées.

Q3. En vous aidant du site <http://www.apprendre-en-ligne.net/crypto/passesecret/ohm.html>, donner la valeur des résistances utilisées dans le montage.

#### 4. Initiation à la programmation Arduino

Q4. Démarrer le logiciel Arduino avec le lien présent sur le bureau du PC.



Nous allons dans un premier temps programmer le clignotement d'une led (celle connectée sur la pin 13 de la carte Arduino).

La première opération consiste à déclarer les variables que nous utiliserons dans notre programme. La variable led1 sera associé à la pin 13.

Q5. Ecrire le code suivant dans l'interface du logiciel

```

// Variables
int led1 = 13;

// Initialisation
void setup() {
// initialise la pin13 comme une sortie
pinMode(led1, OUTPUT);
}

// Création d'une boucle
void loop() {
  digitalWrite(led1, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(150); // wait for a second
  digitalWrite(led1, LOW); // turn the LED off by making the voltage LOW
  delay(150); // wait for a second
}

```

Quelques explications :

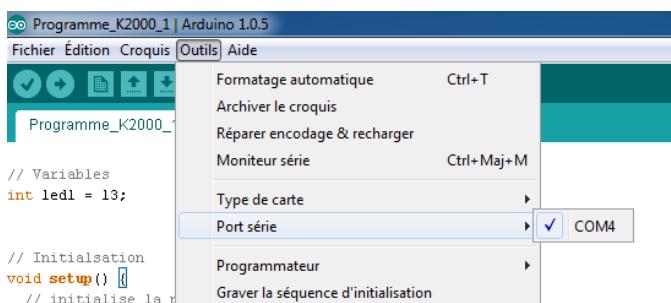
Lorsque vous avez écrit ce code dans le logiciel, on voit que le texte n'a pas la même couleur en fonction des lignes ou des mots.

On remarque que le texte précédé d'un // signifie un commentaire. Il est là pour présenter ce qui va être fait par la suite. Ce texte est facultatif.

En **orange**, apparaissent les mots-clés reconnus par le langage Arduino comme des **fonctions** existantes. Lorsqu'on sélectionne un mot coloré en orange et qu'on effectue un clic avec le bouton droit de la souris, l'on a la possibilité de choisir « Find in reference » : cette commande ouvre directement la documentation de la fonction sélectionnée.

En **bleu**, apparaissent les mots-clés reconnus par le langage Arduino comme des **constantes**.

Q6. Connecter la carte Arduino au PC et vérifier que le port est bien paramétré dans le logiciel... Pour cela cliquer sur « Outil » et « Port série ». Le port doit être validé comme sur l'image ci-dessous.



Q7. Vérifier aussi que le type de carte soit correctement paramétré (carte Uno)

Q8. Téléverser le programme dans la carte et observer le résultat





## 5. Evolutions du programme.

### Clignotement série :

Nous souhaitons faire évoluer le programme de façon à ce que les leds s'allument les unes après les autres avec un délai de 150 ms.

Q9. Modifier le programme pour répondre au besoin précédent.

### Clignotement avec variable de temps :

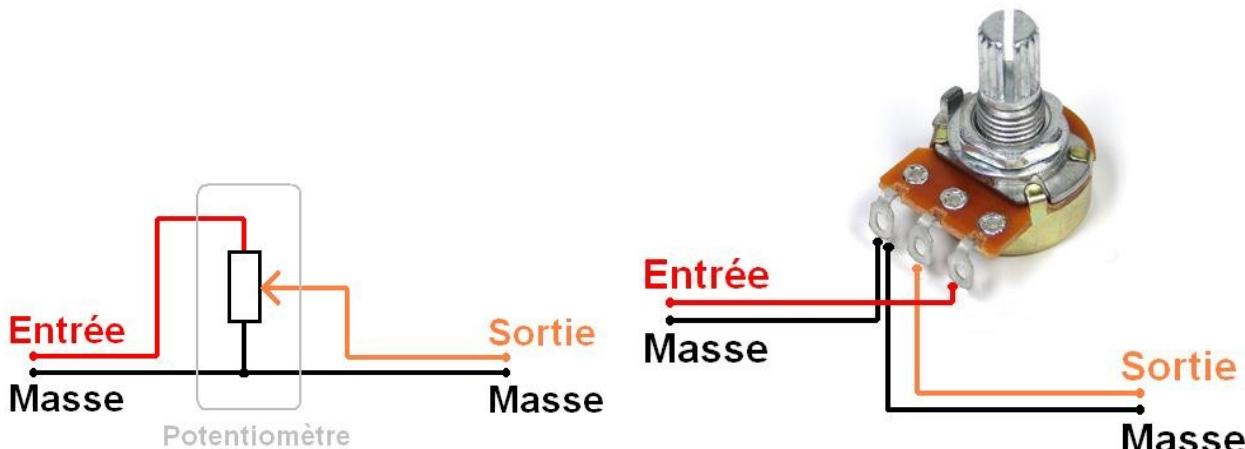
Ajouter une variable permettant de faire varier le temps de clignotement des leds. Nous noterons cette variable « time »

Q10. Modifier le programme pour répondre au besoin précédent et faire varier le temps de clignotement.

### Utilisation d'un potentiomètre pour faire varier la vitesse de clignotement :

Un potentiomètre permet de faire varier la résistance en fonction de sa position angulaire.

Le schéma équivalent est le suivant :



Q11. Modifier le montage en implantant le potentiomètre de façon à ce que son entrée soit alimentée en +5V (Une sortie de ce type est disponible sur la carte) et que sa sortie soit reliée au pin2 de la carte Arduino.

Ce montage permet de lire une tension variable entre 0 et +5V sur le pin2 de la carte.

Sachant que

la fonction suivante permet de rentrer la valeur lue sur le pin 2 dans la variable « time »

```
int potPin = 2;
time = analogRead(potPin); // read the value from the sensor
```

Q12. Modifier le programme afin de faire varier la durée de clignotement avec le potentiomètre.

Q13. Réaliser un compte rendu de votre travail