

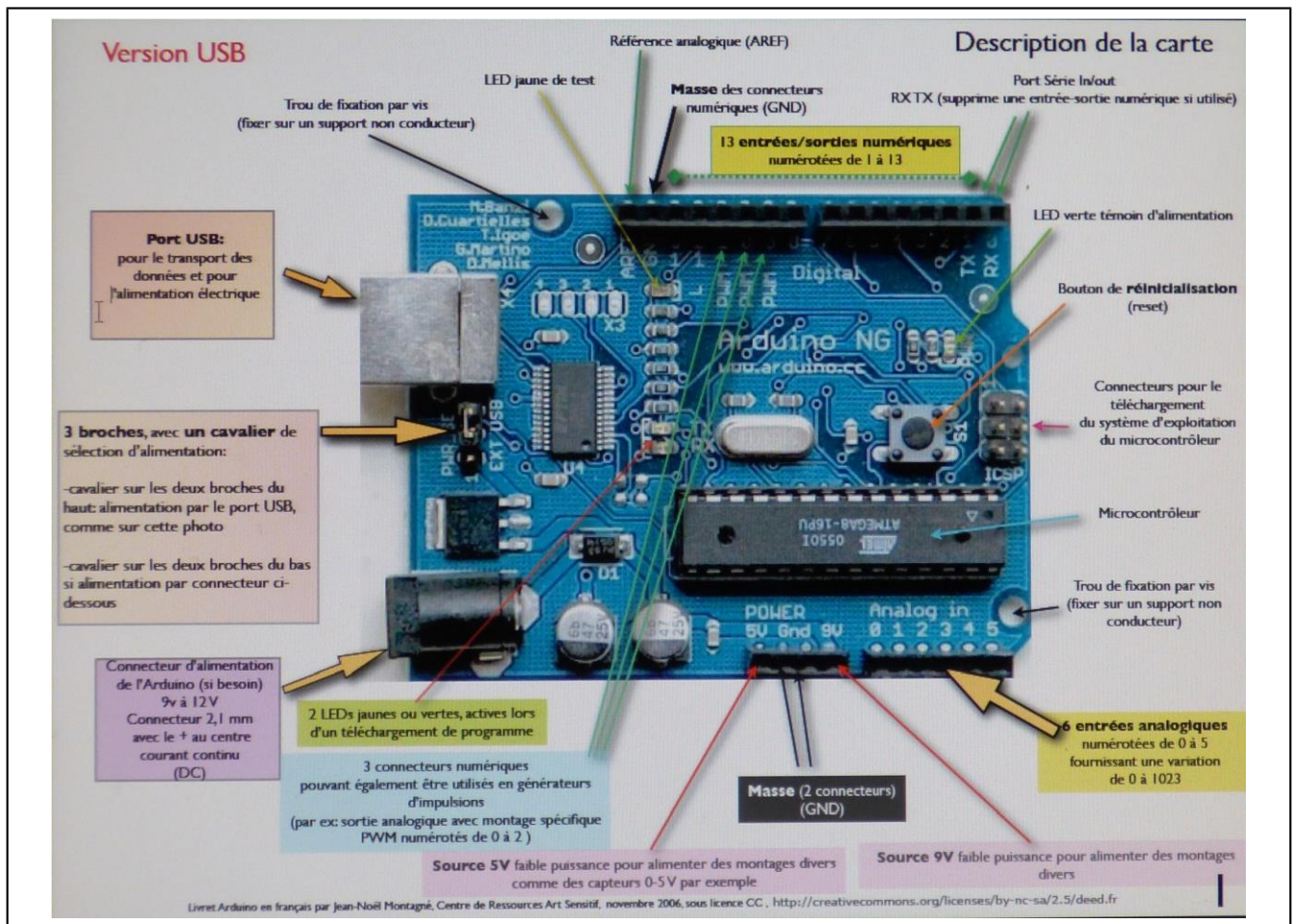
## Fiche pédagogique :

Objectif de la séquence	Dossier technique et dossier ressource	Equipement et matériel nécessaire :
<ul style="list-style-type: none"> <li>Mettre en œuvre le système et tester son fonctionnement.</li> <li>Produire un algorithme simple de fonctionnement.</li> <li>Analyser le branchement de Leds sur le système</li> <li>Utiliser une entrée analogique</li> <li>Interpréter un programme</li> </ul>	<ul style="list-style-type: none"> <li>□ Cours sur Algorigramme.</li> <li>□</li> <li>□ Doc ressource sur logiciel Arduino</li> <li>□ Documentation sur les capteurs</li> <li>□ Livre NSI</li> </ul>	<ul style="list-style-type: none"> <li>Kit Arduino.</li> <li>Logiciel Arduino et Tinkercad</li> <li>Capteur photorésistance</li> <li>LED</li> <li>Câble USB</li> </ul>
Compétences visées	Pré-requis	Positionnement et volumes horaires
Architectures matérielles et systèmes d'exploitation : Périphériques d'entrée et de sortie Interface Homme-Machine.	Représentation des données : <ul style="list-style-type: none"> <li>- Ecriture d'un entier positif dans un base</li> <li>- Valeurs booléennes</li> </ul> Langages et programmation <ul style="list-style-type: none"> <li>- Construction élémentaire</li> </ul>	Classe de première. Premier trimestre.  4h max.

Notes : - Tinkercad peut être utiliser en amont de chaque branchement. Intérêts : éviter les erreurs de branchement et les LED grillées, comparaison modèle physique / modèle numérique, activités débranchées.

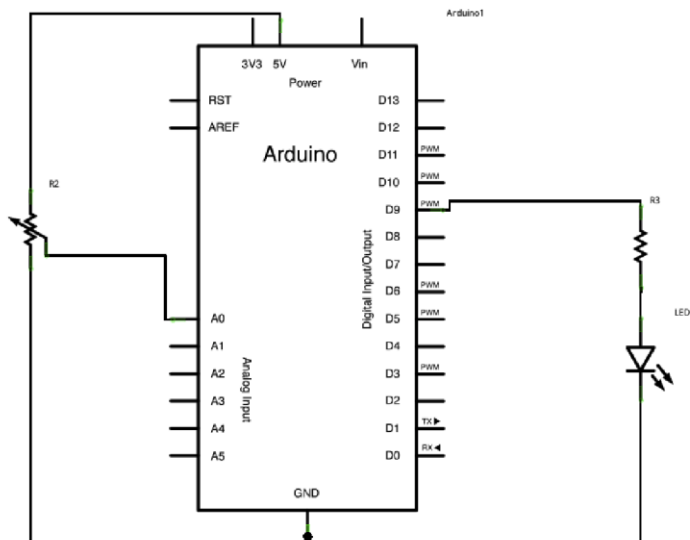
- Le langage Arduino peut être remplacé par du Python (révision de seconde)

- Il y a 60 items au programme et environ 60 semaines de cours sur deux ans ce qui implique une semaine par item en moyenne. Il faut en moyenne consacrer une semaine à l'item périphériques d'entrée et de sortie / Interface Homme Machine soit 2h de cours et de 2h de TP.



## Travail à effectuer en laboratoire :

On souhaite mettre en œuvre le système Arduino pour modifier l'intensité lumineuse d'une LED à partir d'un potentiomètre suivant le schéma ci-dessous :

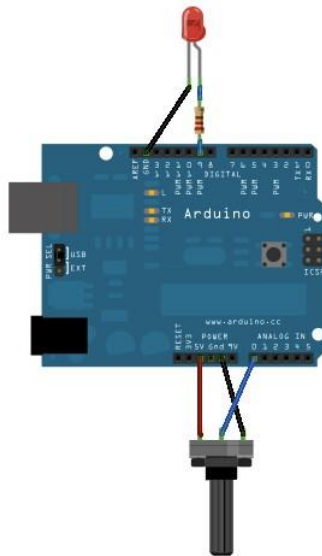


### Activité 1 : Branchement des composants sur la carte Arduino:

- ☐ Connecter le curseur du potentiomètre (broche centrale) sur la pin analogique A0 puis les deux broches restante l'une sur le Vcc (+5V) et l'autre à la masse (GND).

- ❑ Placer la patte courte de la LED (Cathode) sur la masse (GND) puis relier l'autre patte (Anode) à la pin D9 par l'intermédiaire d'une résistance de 220 Ohms pour limiter la courant dans la LED.

- On obtient le circuit suivant :



- Relier la carte Arduino à l'ordinateur avec le câble USB

### **Activité 2 : Premier programme « Test de clignotement d'une LED » :**

On va dans un premier temps faire clignoter une LED sans utiliser le potentiomètre. Ce « test » est la base de tout système programmé, il permet de valider le hard « carte arduino », le soft « logiciel de développement » et la transmission du programme dans la carte.

- ❑ **Lancer le logiciel Arduino (cf. Icône sur le bureau)** puis saisir le « mini-programme » ci-dessous avec l'éditeur de texte.
- ❑ **Compiler** le programme avec le bouton ☒ pour vérifier s'il n'y a pas d'erreur de saisie puis observer en bas de page le nombre d'octets que contient votre programme.
- ❑ **Transférer** les octets de votre programme sur la carte Arduino (Cliquer sur ➔ )

```
/*
Objet : Clignotement de la LED soudée sur la carte Arduino
*/

// Définition du numéro de la patte (ou pin)
// sur laquelle est connectée la LED montée sur la carte (pin13)
const int LED_test = 13;

// On active notre LED en sortie pour la commander
void setup()
{
    pinMode(LED_test, OUTPUT);
}

// On allume et on éteint alternativement la LED sur la patte 13
void loop()
```

```

{
  digitalWrite(LED_test, HIGH);
  delay(500);
  digitalWrite(LED_test, LOW);
  delay(1000);

}

```

### **Activité 3 : Performance du test:**

On vous demande de remettre en œuvre plusieurs fois le programme afin d'en maîtriser chaque ligne.

\*\*\*\*\*Faire valider par le professeur\*\*\*\*\*

- **Présenter le programme sous la forme d'un algorithme.**
- **Évaluer** la période de clignotement de la LED en liaison avec la fonction `delay( ??);` en déduire l'unité de temps du chiffre entre parenthèses.
- **Modifier** les valeurs (en divisant par 10 par ex) puis relancer le test pour voir l'effet sur l'éclairement de la LED.
- **Modifier** le programme pour mettre en œuvre la LED placée à l'extérieur de la carte comme sur le schéma d'implantation vu page précédente.
- **Présentez** vos résultats dans votre compte-rendu.

### **Activité 4 : Evolution du produit:**

On vous demande de mettre en œuvre un potentiomètre comme un capteur capable de faire varier l'intensité de l'éclairement de la LED.

- **Le capteur** (potentiomètre) est branché entre le 0V et le +5V, son curseur délivre donc une tension (grandeur analogique) comprise entre 0 et 5V qu'il va falloir convertir en grandeur numérique pour être compris par la logique du microprocesseur de la carte Arduino.
- **Le convertisseur** (Analogique/Numérique) du microprocesseur va convertir la valeur (0-5V) du capteur « `analogCapteur1` » en une variable numérique « `sensorValue` » de résolution (0-1023) représentant l'image du capteur.

```

// lire la valeur analogique du capteur et la convertir en valeur numérique:
sensorValue = analogRead ( analogCapteur1 );

```

- **La sortie digitalWrite** du microprocesseur place un état 0 ou 1 dont la durée peut être modulée par la fonction `delay( ??)`

- **Le programme devient :**

```

/*

```

Objet : Clignotement de notre LED

```
*/  
  
// Définition du numéro de la patte (ou pin)  
  
// sur laquelle est connectée la  
LED const int LED_test = . ??..;  
const int AnalogCapteur = 0; int  
SensorValue = 0;  
  
// On active led en sortie  
  
void setup()  
{ pinMode(LED_test, OUTPUT);  
}  
  
/* On allume et on éteint alternativement la LED  
en fonction de la valeur numérique du capteur (sensorValue)  
*/  
  
void loop()  
{  
SensorValue = analogRead(AnalogCapteur);  
digitalWrite(LED_test, HIGH);  
delay(SensorValue / 50);  
digitalWrite(LED_test, LOW);  
delay((1023 - SensorValue) / 50);  
}
```

\*\*\*\*\*Faire valider par le professeur\*\*\*\*\*

- **Expliquer précisément le rôle** de chacune des lignes de la boucle **loop( )**.
- **Modifier la division** par 50. Expliquer son rôle.
- **Présentez** vos résultats dans votre compte-rendu.

#### **Activité 5 : Autre mode de commande :**

- **Modifier** votre programme pour obtenir celui-ci. ☐ **Tester** le.

```
/*
```

Objet : Clignotement de la LED

```
*/  
  
// Définition du numéro de la patte (ou pin)  
// sur laquelle est connectée la  
LED const int LED = ?? ; const int  
AnalogCapteur = 0; int SensorValue  
= 0;  
  
// On active la led en sortie  
void setup()  
{ pinMode(LED, OUTPUT);  
}  
  
/* On allume la LED en fonction de la valeur SensorValue  
*/  
  
void loop()  
{  
SensorValue = analogRead(AnalogCapteur);  
analogWrite(LED, SensorValue / 4);  
}
```

\*\*\*\*\*Faire valider par le professeur\*\*\*\*\*

- **Rechercher** quel est le principe de fonctionnement de la fonction **analogWrite()** □  
**Conclure** sur la différence de commande de la led entre le programme 2 et celui-ci.
- **Présentez** vos résultats dans votre compte-rendu.