

Objectifs

L'idée première est de prendre en main les cartes Arduino en réalisant rapidement des montages sur plaque de prototypage rapide (Breadboard) sans pour autant tout comprendre aux programmes proposés....juste l'essentiel !

Les activités sont graduelles en complexité sans toutefois être bien compliquées, tous les programmes sont fournis. Elles vous donneront des bases saines pour envisager des projets personnels ou dans le cadre des projets de votre scolarité.

Pour cette prise en main nous utiliserons les cartes Arduino Mega2560 même si les illustrations seront faites avec une carte Arduino Uno (plus petite que la Mega2560) du simulateur en ligne [TinkerCad](#) (nécessite la création d'un compte gratuit pour l'utiliser)

Présentation de la carte Uno et la carte Méga2560

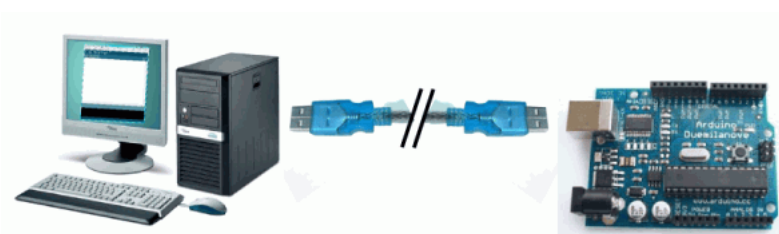
Cf. document en annexe A.

Premier programme

Afin de vérifier le bon fonctionnement, on commence en général par téléverser le programme exemple « Blink.ino ».

➤ **Cablage à réaliser :**

Pour le premier test il suffit de connecter la carte Arduino à l'ordinateur par l'intermédiaire d'un câble USB A / USB B

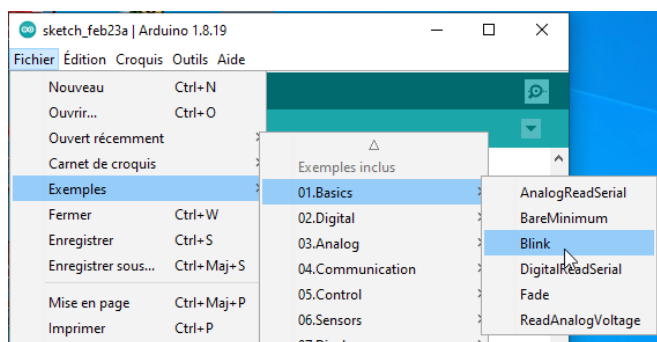


➤ **Programme à téléverser**

- ◆ Ouvrir l'IDE (« Integrated Development Environment ». Cette dénomination anglaise peut être traduite par « environnement de développement intégré ») Arduino.



- ◆ Dans les exemples, chercher et ouvrir l'exemple Blink.



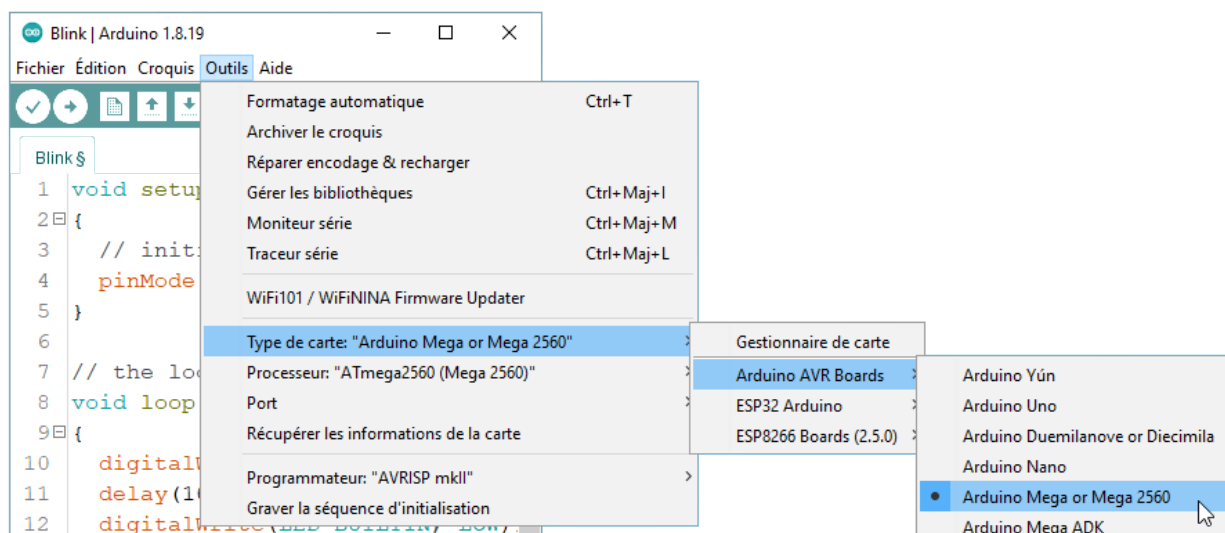
Programme « Blink »

```

Blink $
1 void setup()
2 {
3   // initialize digital pin LED_BUILTIN as an output.
4   pinMode(LED_BUILTIN, OUTPUT);
5 }
6
7 // the loop function runs over and over again forever
8 void loop()
9 {
10  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
11  delay(1000); // wait for a second
12  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
13  delay(1000); // wait for a second
14 }
  
```

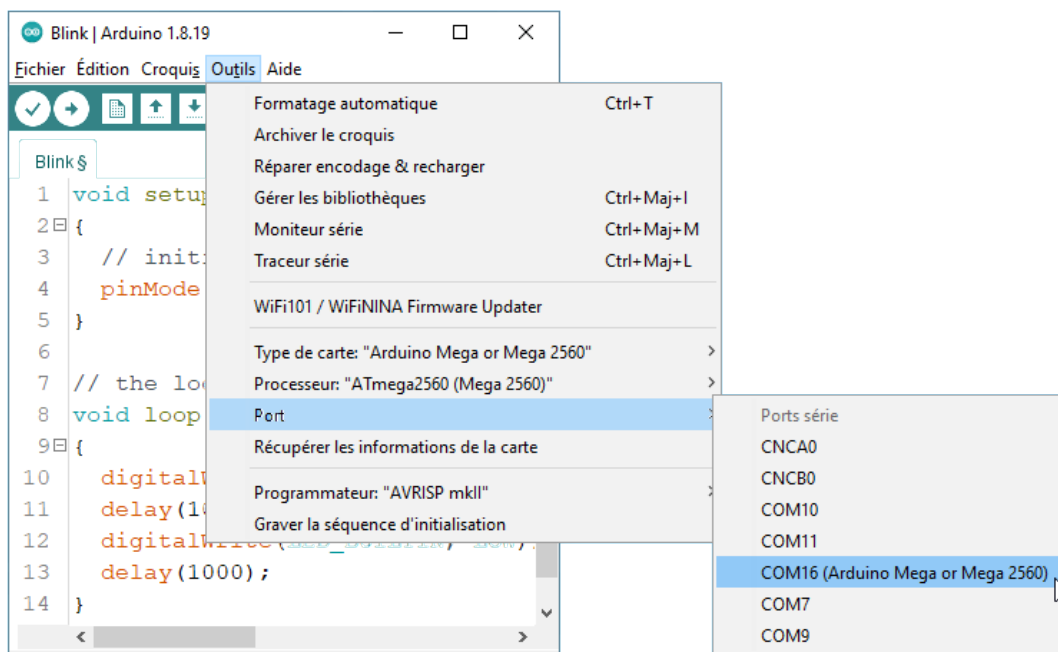
Ce programme fait clignoter la led « LED_BUILDING » présente sur la carte à 0,5Hz, soit 1 seconde allumée (1000 milli secondes) et 1 seconde éteinte.

- ◆ Paramétrer la compilation cible :



Sélectionner à partir du menu de l'IDE Arduino la carte cible (ici Arduino Mega 2560).

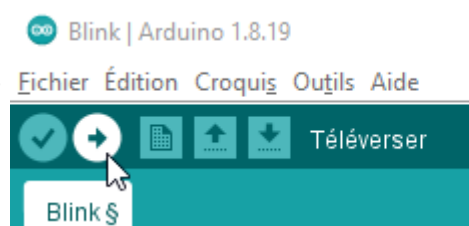
◆ Paramétrer la communication :



Sélectionner à partir du menu de l'IDE Arduino le port de communication avec la carte (sur l'exemple c'est le port COM16, certainement différent pour vous). *La carte doit bien évidemment être connectée avant pour qu'elle soit vue par l'IDE Arduino !*

◆ Compiler et téléverser le programme :

La flèche blanche permet de compiler et téléverser le programme.
La coche juste à gauche permet de vérifier les erreurs de codes éventuelles.



Téléverser le programme dans la carte en cliquant sur la flèche blanche.

➤ **Fonctionnement à observer**

Lors du téléversement du programme on peut observer le clignotement aléatoire des leds Rx/Tx de la carte signifiant le transfert des données.

Observer le clignotement de la LED_BUILDIN (led 13) (cf. Annexe A pour la repérer sur la carte).



- ◆ Vous pouvez changer la valeur delay() le cas échéant pour observer les changements. Un nouveau téléversement doit être fait pour que les modifications prennent effet.

Deuxième programme – Faire varier l'intensité lumineuse d'une LED

Pour faire varier l'intensité lumineuse d'une LED, on utilise la commande PWM ce qui revient à allumer et éteindre la LED très rapidement. La perception ressentie est une LED dont la luminosité varie.

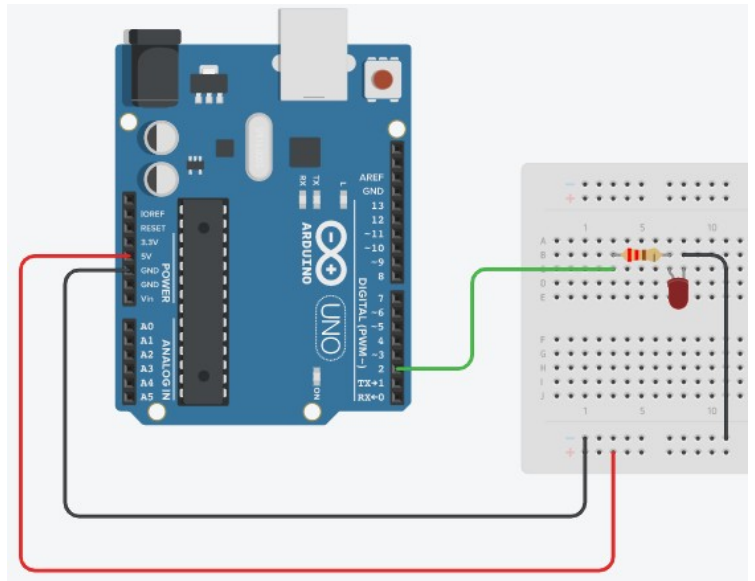
Le câblage à effectuer est le suivant :

La résistance est de 220 Ω

Récupération du code :

- Cliquer sur l'image pour accéder à la simulation TinkerCad et récupérer le code si besoin.
- Ou ouvrir le fichier PWM_FaireVarierLed.txt pour le code à télécharger.

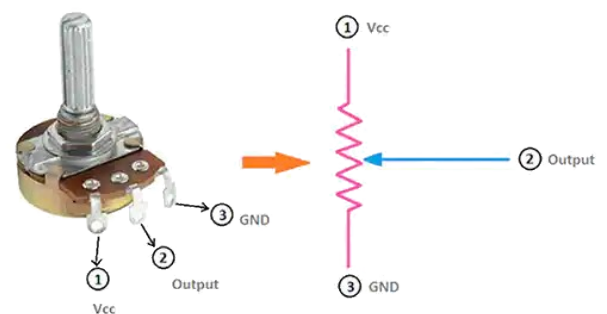
◆ **Téléverser** et **observer** le fonctionnement



Troisième programme – Acquérir une information analogique

Pour ce premier exemple, nous utiliserons un potentiomètre (c'est une résistance à 3 connecteurs dont la résistance varie entre deux des connecteurs).

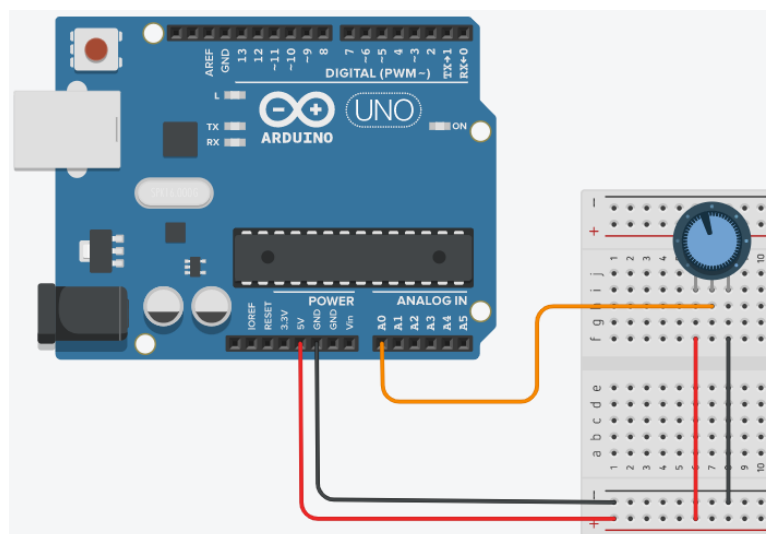
On se servira dans un premier temps du terminal série pour observer la valeur convertie par le CAN (convertisseur analogique numérique de la carte).



Le câblage à effectuer est le suivant :

Récupération du code :

- Cliquer sur l'image pour accéder à la simulation TinkerCad et récupérer le code si besoin.
 - Ou ouvrir le fichier CAN-potentiometre.txt pour le code à télécharger.
- ◆ **Téléverser** et **observer** le fonctionnement en ouvrant le terminal série (menu « Outils > Moniteur Série » ou CTRL+MAJ + M)



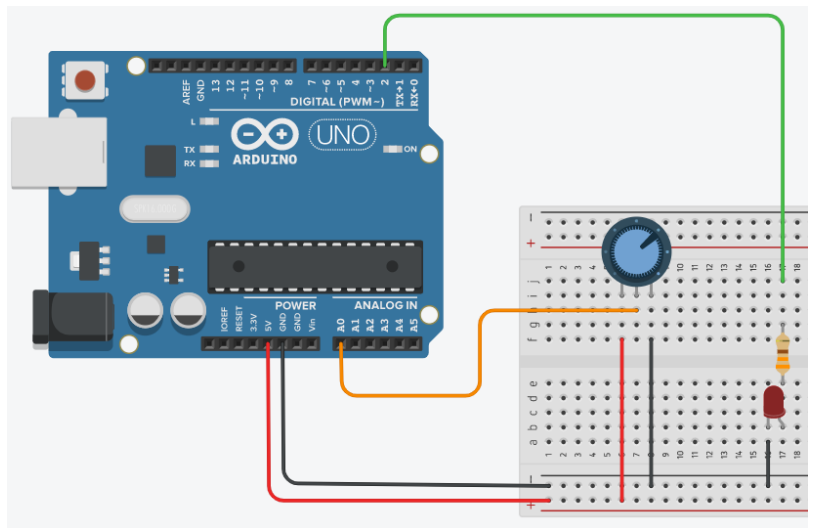
Quatrième programme – Faire varier luminosité d'un led à partir d'un potentiomètre

Il s'agit ici de faire varier l'intensité lumineuse d'une led en agissant sur un potentiomètre. C'est en quelque sorte une concaténation des programmes 2 et 3.

Le câblage à effectuer est le suivant :

Récupération du code :

- Cliquer sur l'image pour accéder à la simulation TinkerCad et récupérer le code si besoin.
- Ou ouvrir le fichier `VarietLedAvecPotentiometre.txt` pour le code à téléverser.
- ◆ **Téléverser** et **observer** le fonctionnement



Cinquième programme – activer éclairage en fonction de la luminosité

Il s'agit ici de faire varier l'intensité lumineuse d'une led en fonction de la quantité de lumière (l'éclairement). Cet éclairement est capté par une LDR (Light Dependent Resistor, photorésistance en français). C'est un composant dont la résistance varie en fonction de la luminosité (la résistance diminue avec l'obscurité).

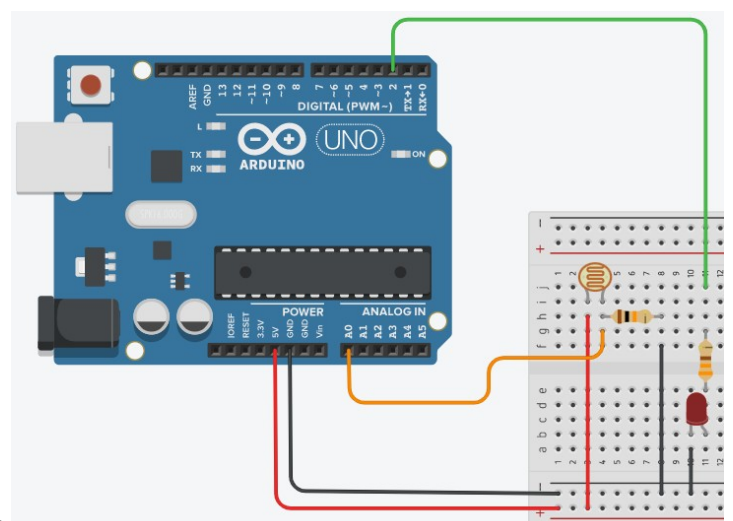
Contrairement à un potentiomètre qui possède 3 connecteurs, la LDR est un dipole (2 connecteurs). Elle doit par conséquent être insérée dans un montage électrique. Nous l'utiliserons dans un « pont diviseur de tension ».

Le câblage à effectuer est le suivant :

La résistance est en série avec la LDR dans le pont diviseur de tension est de 10 k Ω

Récupération du code :

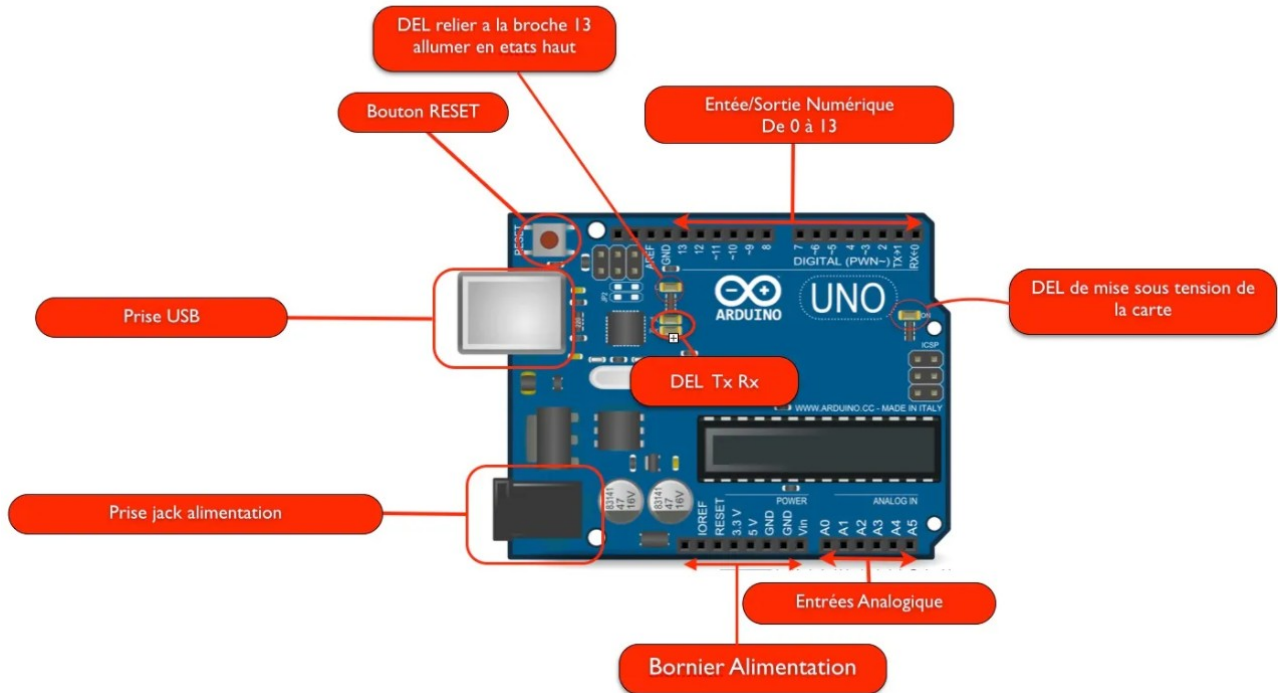
- Cliquer sur l'image pour accéder à la simulation TinkerCad et récupérer le code si besoin.
- Ou ouvrir le fichier `VarietLedAvecLDR_eleve.txt` pour le code à téléverser.



- ◆ **Téléverser** et **observer** le fonctionnement. Vous devriez constater que le fonctionnement est inverse à celui attendu. Normalement la led éclaire plus intensément lorsque la LDR ne capte plus de lumière. **Modifier** le code en conséquence pour obtenir le fonctionnement attendu.
- ◆ À partir du terminal série, **observer** les valeurs minimale et maximale contenues dans la carte Arduino après conversion analogique numérique (variable « valN_ldr »). Vous constaterez qu'elles ne varient pas de 0 à 1023. **Modifier** le code pour obtenir de meilleures performances de la led en fonction de l'éclairement (c'est-à-dire LED éteinte quand éclairée au smartphone et led pleinement allumée lorsque le doigt est posé sur la LDR pour simuler l'obscurité complète).

ANNEXE A – PRESENTATION CARTES UNO ET MEGA 2560

➤ Carte Uno



➤ Carte Méga 2560

