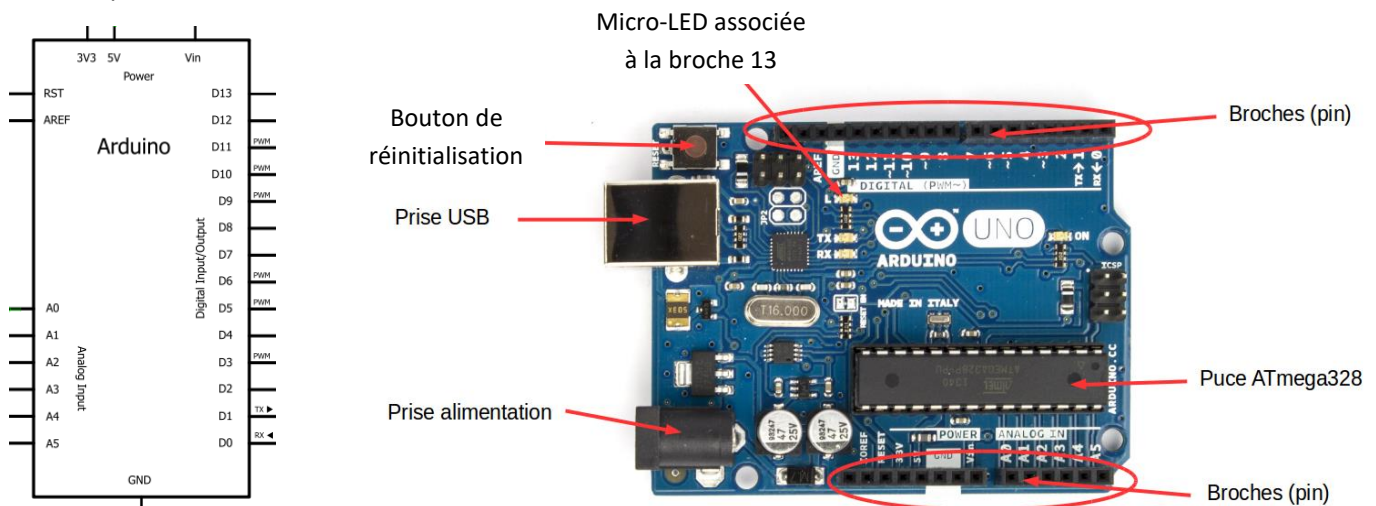


« Le succès est un mauvais professeur. Il pousse les gens intelligents à croire qu'ils sont infailibles. » Bill Gates

Arduino est une marque qui couvre des cartes matériellement libres sur lesquelles se trouve un microcontrôleur. L'arduino Uno utilise une architecture Atmel AVR nommée Atmega328p.

Ci-dessous une photo d'une carte Arduino Uno avec le schéma électronique qui sera utilisé pour la représenter.



Nous ne programmerons pas directement le microcontrôleur. Nous allons utiliser le protocole Firmata pour assurer la communication entre un ordinateur et un microcontrôleur.

Pour installer la bibliothèque *pyfirmata* qui implémente le protocole en Python, il suffit d'ouvrir dans Spyder via le menu « Outils » une invite de commande puis de taper la commande suivante :

```
pip install pyfirmata
```

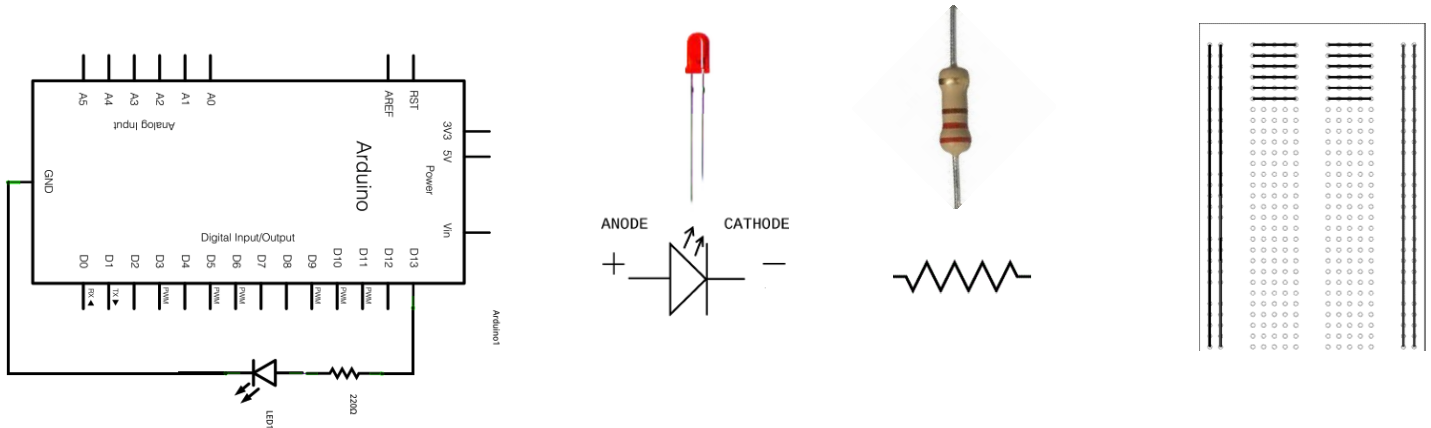
- 1- Le premier programme consiste à allumer pendant 5 secondes puis à éteindre la micro-LED intégrée associée à la broche 13 :

```
import pyfirmata
import time
port = 'COM3' #windows, le numéro dans le gestionnaire de périphériques
#port = '/dev/ttyACM3' #linux, le numéro avec la commande: ls -l /dev/ttyACM*
HIGH = 1
LOW = 0
board = pyfirmata.Arduino(port) # initialise la communication avec la carte
LED_pin = board.get_pin('d:13:o') # initialise la broche (d => digital, 13 => N° broche, o => output)
LED_pin.write(HIGH) # envoie l'ordre de mettre la broche à l'état haut
time.sleep(5) # pause de cinq secondes
LED_pin.write(LOW) # envoie l'ordre de mettre la broche à l'état bas
board.exit() # clôture la communication avec la carte
```

Reliez la carte et l'ordinateur à l'aide du cordon USB puis exécutez le code Python via une console IPython.

- 2- A l'aide d'une boucle, modifiez le programme pour que la LED clignote dix fois de suite avec une période d'une seconde.
- 3- A l'aide d'une variable booléenne, conservez le même comportement mais de sorte que vous n'ayez qu'un seul appel à la fonction « LED\_pin.write » dans votre boucle.

- 4- Réalisez un montage en plaçant une LED sur un banc d'essai relié à la carte Arduino. Ci-dessous le schéma électronique du montage, l'illustration montrant la correspondance entre la LED et sa représentation schématique, l'illustration de la résistance de 220ohm et un schéma révélant les bandes conductrices sous-jacentes au banc d'essai. Vérifiez deux fois le montage puis testez le circuit avec votre dernier programme.



- 5- Afin de faire clignoter indéfiniment la LED, il faudrait remplacer la boucle « pour » dans une boucle « tant que » avec une condition toujours vraie. Cette technique obligerait l'utilisateur à interrompre le programme pour mettre fin au clignotement. Le problème est que, dans ce cas, la clôture du programme ne pourrait se faire correctement. Une manière de pallier cette difficulté consiste à utiliser le mécanisme d'exception. Ce mécanisme permet d'intercepter une erreur d'exécution avant qu'elle interrompe le programme et de trouver une solution. L'arrêt brutal du programme est considéré comme une erreur. Le code ci-dessous illustre l'écoute de toute erreur d'exécution entre les mots clés « try » et « except ». Testez-le.

```
... #importation des bibliothèques et initialisation des variables
board = pyfirmata.Arduino(port)
LED_pin = board.get_pin('d:13:o')
while True:
    try: #début de l'écoute d'erreurs
        LED_pin.write(HIGH)
        time.sleep(0.5)
        LED_pin.write(LOW)
        time.sleep(0.5)
    except: # fin de l'écoute d'erreurs et proposition d'une solution
        break # mets fin à la boucle et par conséquent passer à la ligne suivante
board.exit()
```

- 6- Faites le montage suivant avec 6 LED et programmez le déroulement des différents motifs ci-dessous :

