

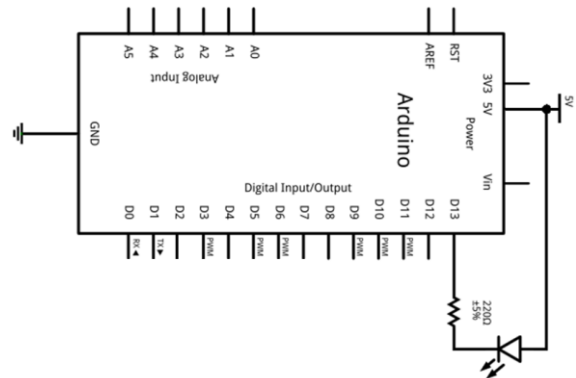
« Là où une calculatrice comme l'ENIAC aujourd'hui est équipée de 18.000 tubes à vide et pèse 30 tonnes, les ordinateurs à l'avenir peuvent avoir seulement 1000 tubes à vide et peut-être peser seulement 1 ½ tonne. »  
Andrew Hamilton, "Brains that Click", Popular Mechanics 91 (3), March 1949, (pp. 162 et seq.) at p. 258.

1. Lors de la dernière séance, nous avons utilisé la procédure « `time.sleep()` » pour introduire une pause dans le déroulement du programme afin de laisser une LED allumée ou éteinte pendant un certain laps de temps. Maintenant, si nous souhaitons par exemple exécuter des instructions pendant un laps de temps l'utilisation de « `time.sleep()` » n'est plus adapté. En effet, Nous ne pourrions continuellement afficher « toto » pendant 1 milliseconde.

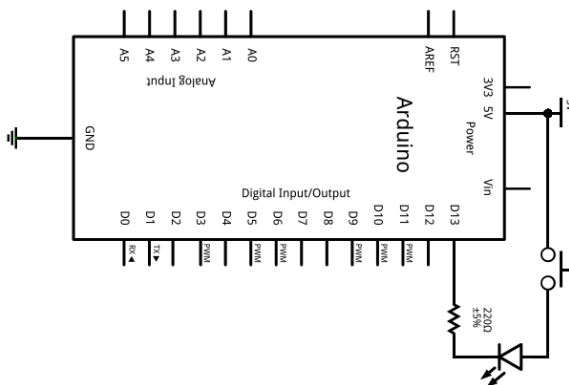
Une solution consiste à mesurer le temps passé avec la fonction « `time.clock()` », c'est-à-dire faire la différence entre le temps mesurer à l'instant « t1 » et celui mesurer à l'instant « t2 ». De sorte qu'un programme qui afficherait continuellement afficher « toto » pendant 1ms pourrait être comme ci-contre.

Réalisez un montage avec une LED comme l'illustre le schéma ci-contre et écrivez un programme qui fasse clignoter la LED avec une période d'une seconde sans utiliser « `time.sleep()` ».

```
import time
t1 = time.clock()
t2 = t1
while t2 - t1 < 0.001:
    print ("toto")
    t2 = time.clock()
```

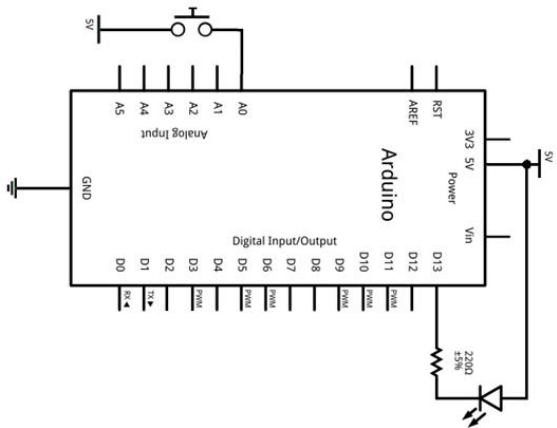


2. Réalisez un programme qui allume indéfiniment la LED et modifiez le montage en rajoutant un bouton poussoir normalement ouvert illustré ci-dessous avec son schéma électronique. Si le bouton est relâché alors le courant ne passe pas, le circuit est ouvert. Si le bouton est Appuyé alors le courant passe, le circuit est fermé :



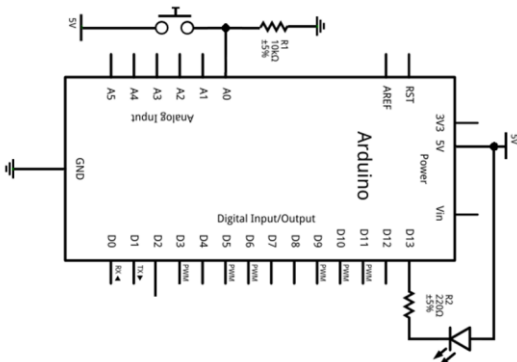
3. Maintenant nous souhaitons gérer via un programme l'allumage de la LED à l'aide d'un bouton poussoir. Ci-dessous une solution de montage est proposée : le bouton poussoir une fois appuyé applique une tension à une broche que l'on pourra lire depuis le programme. Afin de comprendre le problème de ce montage, nous avons utilisé une broche « analogique ».

Ecrivez un programme qui allume la LED pendant 5 seconde et qui capture pendant ce laps de temps la mesure effectuée sur la broche A0 pendant que vous appuyez et relâchez le bouton poussoir. Ci-dessous le schéma du montage et une partie du programme à compléter, la partie en gras sert à afficher sous forme graphique les données capturés :



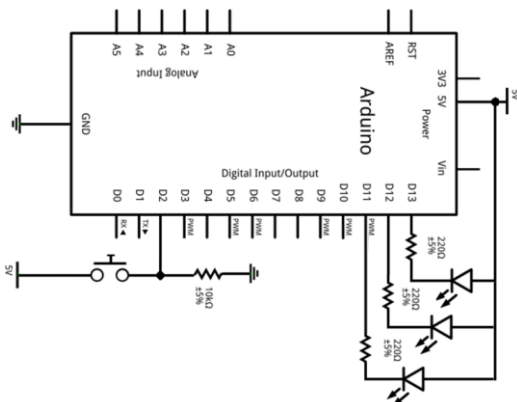
```
import pyfirmata
import time
import matplotlib.pyplot as graphic
#importe la bibliothèque "pyplot" et la renome " graphic"
port = 'COM3'
data = []
board = pyfirmata.Arduino(port)
...
switch_pin = board.get_pin('a:0:i')
iterator = pyfirmata.util.Iterator(board) # initialise le début
iterator.start() # de la journalisation
switch_pin.enable_reporting() # autorise la journalisation
...
value = switch_pin.read()
...
board.exit()
graphic.plot(data)
graphic.show()
```

4. Afin de tirer rapidement le potentiel à zéro et ne plus dépendre des contraintes électromagnétiques environnementales, il faut placer une résistance R1 de 10kΩ comme indiqué sur le schéma ci-dessous avec une photo d'une résistance de 10kΩ. De sorte que la broche est toujours reliée à la masse et que si le bouton est appuyé le courant aille vers la moindre résistance c'est-à-dire vers la broche et non la masse ce qui provoquerait un court-circuit :



Refaites l'essai précédent. Ensuite, placer le bouton poussoir sur une broche digital et faites-en sorte d'allumer la LED lorsque le bouton est enfoncé.

5. Nous souhaitons maintenant réaliser un interrupteur bistable, c'est-à-dire qu'un appui/relâché ferme le circuit puis un second appui/relâché ouvre de nouveau :



- Allumez pendant 500ms une LED verte lors du premier front montant (transition état bas/état haut).
- Allumez pendant 500ms une LED jaune lors du premier front descendant.
- Allumez une LED rouge au premier appui/relâché puis l'éteindre second.
- Dessiner le chronogramme décrivant l'état des LED et du bouton poussoir au cours du temps.

6. Utilisez ce principe pour déclencher et arrêter instantanément une des séquences temporelles de 5 LED présentées dans le sujet de la première séance de projet d'application.