 LYCÉE LOUIS LACHENAL	Sciences et Technologies de l'Industrie et du Développement Durable		T ^{le} STI2D
	Raspberry – Partie 4		
	Utilisation du PWM	crolland	Spé SIN

Objectif :

- * Créer le programme pour allumer et éteindre progressivement une LED à l'aide d'un Raspberry Pi
- * Réaliser le câblage
- * Valider le fonctionnement

Prérequis

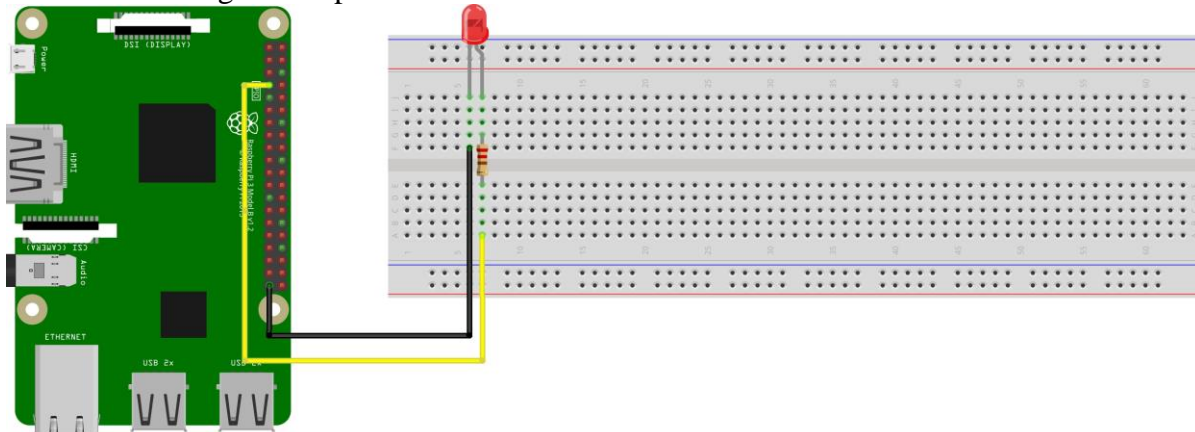
Il vous faut :

- * - Un Raspberry Pi,
- * - Une LED rouge,
- * - Une résistance 220Ω,
- * - Des cables,
- * - Une platine d'essai (BreadBoard)(de préférence).

Une fois tous les éléments réunis, on peut passer au branchement.

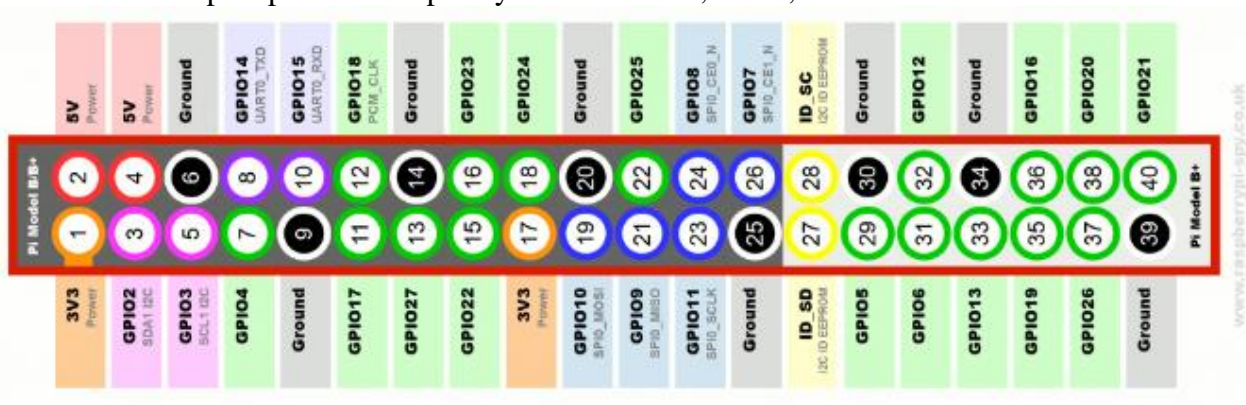
Branchement d'une LED au Raspberry

Réaliser le câblage correspondant au schéma ci-dessous.



fritzing

Numérotation des pins pour les Raspberry Pi1 Model B+, Pi 2B, PiZero et Pi 3B



Identifier à partir du câblage et de la numération des pins, le nom de pins de la Raspberry relié au fil jaune et noir.

Le fil jaune est relié à la broche _____ qui correspond au port _____.

Le fil noir est relié à la broche _____ qui correspond au port _____.

Créer le programme :

Une fois que tout est branché, vous pouvez démarrer votre Raspberry Pi, et vous connecter avec un clavier/souris ou par SSH pour écrire votre premier programme en Python sur le Raspberry Pi.

Commençons par créer un fichier. Le système Raspbian étant une distribution de Linux, les commandes entrées sont des commandes Linux que vous connaissez peut-être déjà Utilisez simplement la commande :

```
sudo nano nomFichier.extension
```

On peut par exemple appeler le programme comme suit :

```
sudo nano led2.py
```

Cette commande crée le fichier led2.py dans le répertoire principal (/home/pi par défaut) et l'ouvre dans un éditeur de texte.

Ecriture du programme python

Le programme ci-contre fait varier l'intensité de l'éclairement de la LED sur GPIO4.

```
1  import time
2  import RPi.GPIO as GPIO
3
4  GPIO.setmode(GPIO.BCM) # numerotation BCM
5  GPIO.setup(4, GPIO.OUT) #GPIO4 en sortie
6
7  ledPWM = GPIO.PWM(4, 1000) # GPIO4 frequence PWM 1000Hz
8  ledPWM.start(0)
9
10 try:
11     while 1: # dc represente le rapport cyclique
12         for dc in range(0, 101, 5): # allumage progressif
13             ledPWM.ChangeDutyCycle(dc)
14             time.sleep(0.1)
15         for dc in range(100, -1, -5): # extinction progressive
16             ledPWM.ChangeDutyCycle(dc)
17             time.sleep(0.1)
18 except KeyboardInterrupt: # CTRL-C pour quitter
19     pass
20 ledPWM.stop()
21 GPIO.cleanup()
22
```

La LED est allumée durant l'état haut de la sortie. La puissance lumineuse restituée est proportionnelle au rapport cyclique $\eta = t_h/T$ (t_h = temps état haut, T est la période ici 1KHz). La persistance rétinienne fait office de filtre passe bas. (Il est admis que l'œil humain ne perçoit pas les variations de lumière inférieure à 30ms)

Analysez et recopiez le programme ci-contre.

Enregistrement du programme écrit

Une fois que nous avons fini l'édition du fichier, on doit l'enregistrer et quitter l'éditeur. Pour se faire, on appuie sur Ctrl+X, sur y (pour yes) (et o si le Raspberry Pi est en français) et enfin sur la touche enter du clavier. On retourne donc dans l'invite de commande et nous avons fermé l'éditeur de texte.

Exécution du programme Python

Pour lancer le programme que nous avons créé, il suffit d'entrer :

```
sudo python led2.py
```

Par conséquent, la commande générale pour lancer un programme Python est :

```
sudo python nomFichier.py
```

Pour aller plus loin

Un signal PWM peut permettre de faire clignoter automatiquement une LED , ici la fréquence est de 1Hz, le rapport cyclique est de $\frac{1}{2}$ ($t_h=T/2$). Un fois le périphérique configuré et activé, le clignotement est automatique et n'est plus géré par le logiciel :

```
1  import RPi.GPIO as GPIO
2  GPIO.setmode(GPIO.BCM)
3  GPIO.setup(4, GPIO.OUT)
4  ledPWM = GPIO.PWM(4, 0.5) #instance avec periode 1s (2x0.5)
5  ledPWM.ChangeDutyCycle(50)# rapport cyclique 1/2
6  ledPWM.start(1) # c'est parti
7  input('Appuyer sur une touche pour terminer')
8  ledPWM.stop()
9  GPIO.cleanup()
10
```