

## TP2 Raspberry PI

### Utilisation des GPIO : Les entrées/sorties numériques

17 GPIO (*General Purpose Input/Output*) sont disponibles sur le connecteur de la Raspberry Pi. La plupart des broches supportant les GPIO peuvent être réassignées à des périphériques de communication. En mode GPIO ces broches peuvent être configurées soit en entrée binaire TOR soit en sortie TOR.

#### I. Installation de Rpi.GPIO, la bibliothèque de gestion des Entrées/Sorties

1. Récupérer la bibliothèque :

```
wget http://pypi.python.org/packages/source/R/RPi.GPIO/RPi.GPIO-0.3.1a.tar.gz
```

2. Décompresser l'archive : `tar xvzf RPi.GPIO-0.3.1a.tar.gz`

3. Aller dans le répertoire de la bibliothèque : `cd RPi.GPIO-0.3.1a`

4. Installer la bibliothèque en tapant la commande suivante : `sudo python setup.py install`

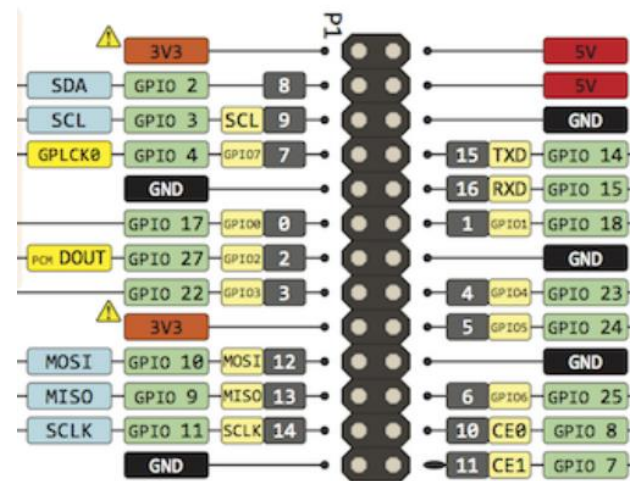
5. Détruire l'archive et le dossier d'installation

```
cd ..  
sudo rm -rf RPi.GPIO-0.3.1a/  
rm RPi.GPIO-0.3.1a.tar.gz
```

#### Exemple d'utilisation de la bibliothèque RPi.GPIO dans un code Python

```
import RPi.GPIO as GPIO          # charge la bibliotheque  
GPIO.setmode(GPIO.BOARD)        # numérotation du connecteur Raspberry, ex 7 pour GPIO4  
GPIO.setmode(GPIO.BCM)          # numérotation du circuit BCM2835, ex 4 pour GPIO4  
  
GPIO.setup(11, GPIO.OUT)         # place la broche (ou le GPIO) en sortie  
GPIO.setup(7, GPIO.OUT, initial=GPIO.HIGH) # GPIO en sortie , initialisé à 1  
GPIO.setup(17, GPIO.IN)          # place la broche (ou le GPIO) en entree  
  
GPIO.setup(channel, GPIO.IN, pull_up_down=GPIO.PUD_UP) # active une resistance de tirage  
GPIO.setup(channel, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)  
  
GPIO.output(11, True)            # met le GPIO à l'état 1 (True ou 1)  
GPIO.output(11, False)           # met le GPIO à l'état 0 (False ou 0)  
  
GPIO.input(7) # lit l'état du GPIO  
# ex a= GPIO.input(7) ou if (GPIO.input(7)) :  
  
GPIO.cleanup() # replace tous les GPIO en entrées  
GPIO.RPI_REVISION # version de la carte  
GPIO.VERSION      # version de la bibliothèque
```

1. Réaliser le montage suivant. La LED rouge est en série avec une résistance de 220 ohms entre la broche 7 du connecteur (GPIO4) et le 0v et un bouton poussoir avec une résistance de pull-up (10K) sur la broche 11 (GPIO17)



## Brochage du connecteur GPIO

fritzing

Le programme suivant fait clignoter la LED avec une période de 1s.

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)    #Utilisation des numeros de broches du processeur
GPIO.setup(4, GPIO.OUT)   #GPIO4 configuree en sortie

while True:               # Boucle infinie
    print("Allume")
    GPIO.output(4,1)      # Allume la LED
    time.sleep(0.5)       # Tempo 0,5s
    print("Eteind")
    GPIO.output(4,0)      # eteind la LED
    time.sleep(0.5)       # Tempo 0,5s
```

2. Lancer l'éditeur « nano » en tapant la commande : **sudo nano**
3. Saisir le programme ci-dessus
4. Pour enregistrer **ctrl+x** et entrer le nom de fichier **GPIO.py**. Ensuite valider en tapant sur **Y**.
5. Pour tester votre programme, taper la commande **sudo python GPIO.py**.
6. Pour sortir du programme, taper sur **ctrl+c**.
7. En vous aidant des lignes de code fournies en page 1, modifier le script précédent afin que la diode ne clignote que si le BP est enfoncé (la diode devra être éteinte sinon).
8. Enregistrer le nouveau script sous le nom **GPIO1.py** et tester.