**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.

1. **Installation de Rpi.GPIO, la bibliothèque de gestion des Entrées/Sorties**
2. Récupérer la bibliothèque :

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

1. Décompresser  l’archive : **tar xvzf RPi.GPIO-0.3.1a.tar.gz**
2. Aller dans le répertoire de la bibliothèque : **cd RPi.GPIO-0.3.1a**
3. Installer la bibliothèque en tapant la commande suivante : **sudo python setup.py install**
4. Détruire l’archive et le dossier d’installation   
    **cd ..  
    sudo rm –tf RPi.GPIO-0.3.1a/  
    rm RPi.GPIO-0.3.1a.tar.gz**

Exemple d’utilisation de la biliothèque 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)**

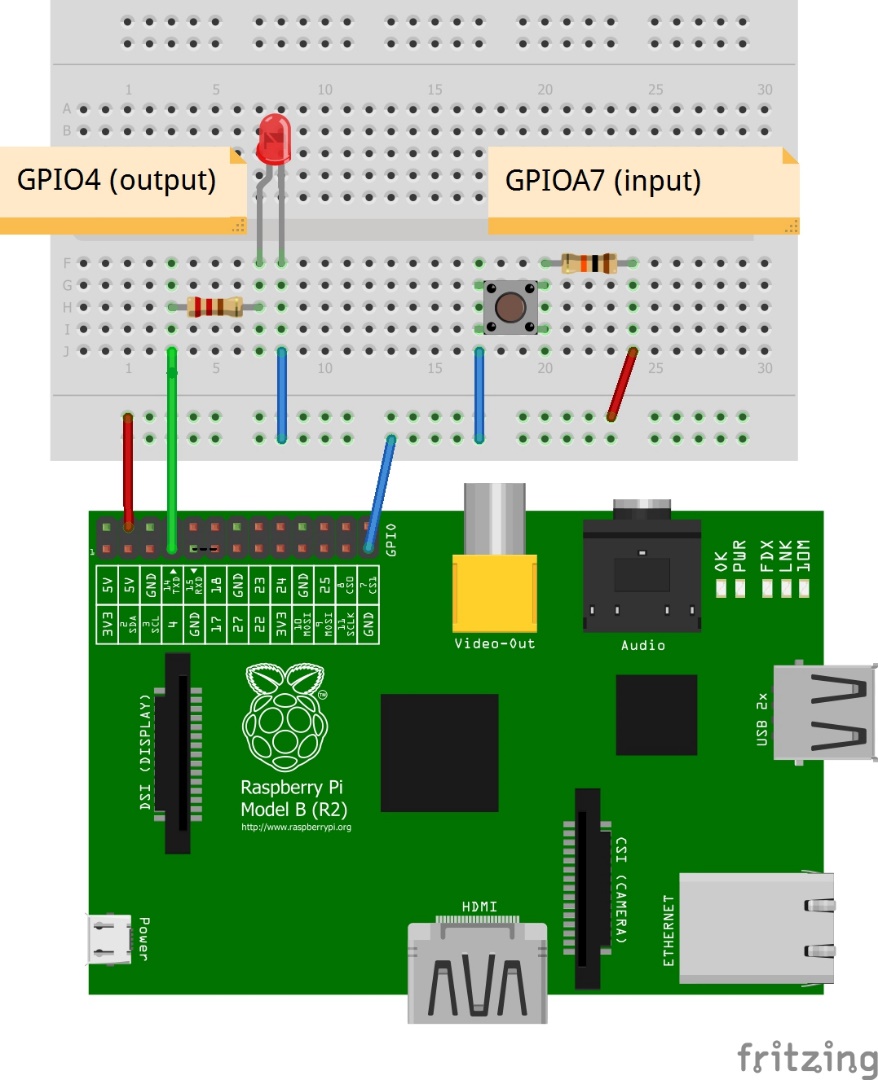
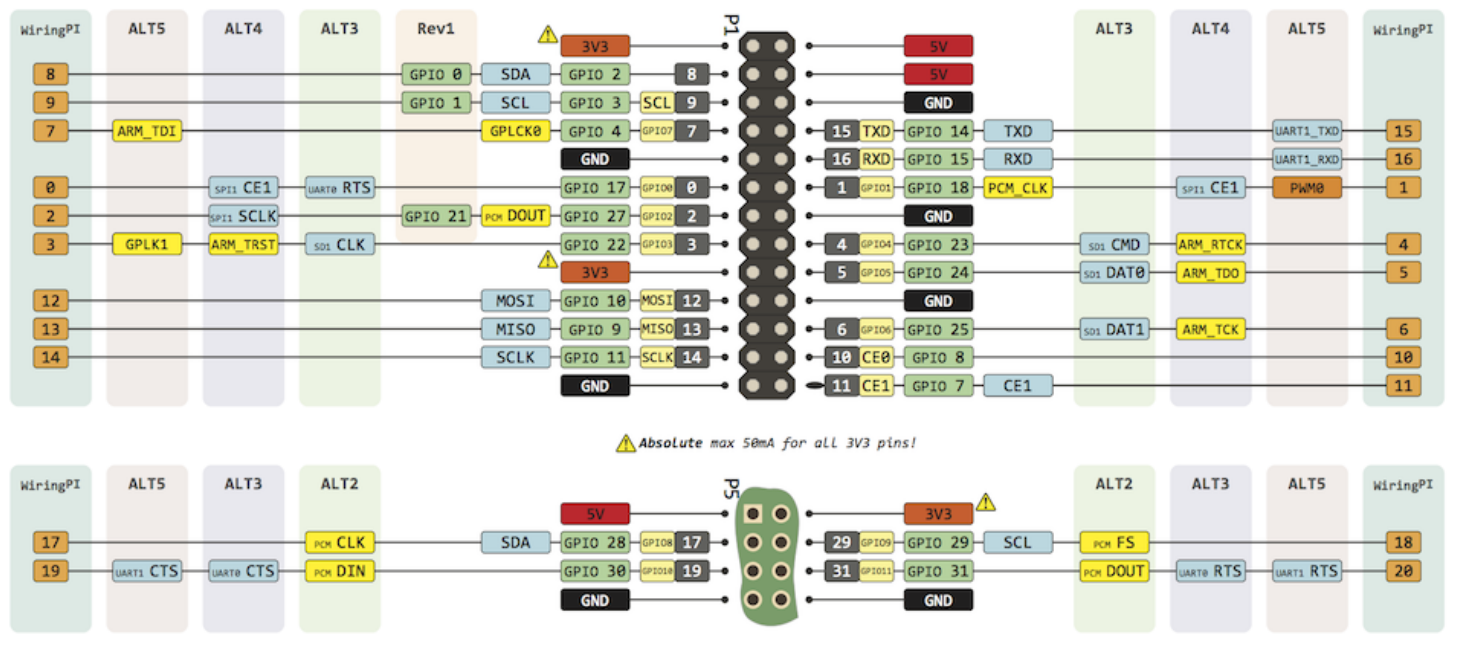
**GPIO.input(7) # lit l’état du GPIO  
 # ex a= GPIO.input(7) ou if (GPIO.input(7)) :**

**GPIO.cleanup() # replace tous les GPIO en entrées**

**GPIO.RPI\_REVISION # version de la carte**

**GPIO.VERSION # version de la bibliothèque**

1. **Mise en œuvre des entrées/sorties TOR du Raspberry :**
2. 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)



**GPIO4 (OUTPUT)**

**GPIO17 (INPUT)**

**Brochage du connecteur GPIO**

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**

1. Lancer l’editeur « nano » en tapant la commande : sudo nano
2. Saisir le programme ci-dessus
3. Pour enregistrer **ctrl+x** et entrer le nom de fichier **GPIO.py**. Ensuite valider en tapant sur **Y**.
4. Pour tester votre programme, taper la commande **sudo python GPIO.py**.
5. Pour sortir du programme, taper sur **ctrl+c**.

Le programme suivant permet de lire l’état de la broche GPIO17 dépendant de l’appui ou non sur le bouton poussoir :

**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**

**GPIO.setup(17, GPIO.IN) #GPIO17 configuree en sortie**

**while True: # Boucle infinie**

**EtatBP = GPIO.input(17) #Memorisation de l’état de la GPIO17 dans la variable EtatBP**

**if EtatBP==False :**

**print("Bouton enfonce")**

**else :**

**print("Bouton relache")**

1. Lancer l’éditeur « nano ».
2. Saisir le programme ci-dessus
3. Pour enregistrer **ctrl+x** et entrer le nom de fichier **GPIO1.py**. Ensuite valider en tapant sur **Y**.
4. Pour tester votre programme, taper la commande **sudo python GPIO1.py**.
5. En vous aidant des lignes de code précédentes, modifier le script précédent afin que la diode ne clignote que si le BP est enfoncé (la diode devra être éteinte sinon).
6. Enregistrer le nouveau script sous le nom **GPIO2.py et tester le.**