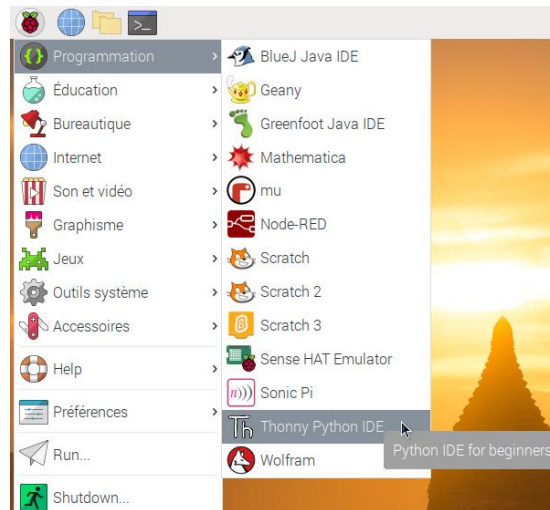




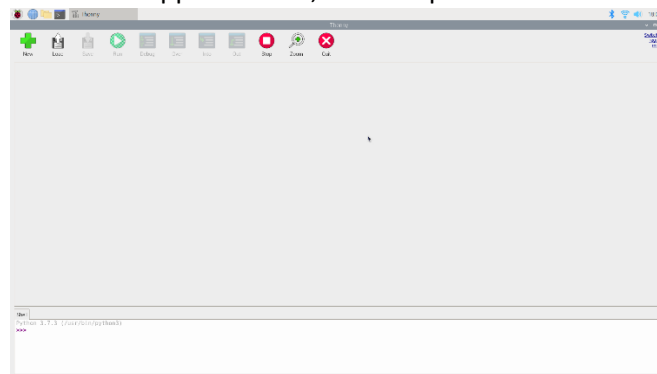
Fiche Tutoriel 7

Raspberry utiliser les GPIOs dans python

Pour pouvoir utiliser les Gpios dans python il faut tout d'abord ouvrir sous Raspbian l'IDE de programmation python Thonny python de Raspbian. Pour cela cliquez sur menu framboise puis programmation et enfin Thonny python. Dans python tous les commentaires sont précédés du signe #. Ils ne seront donc pas interprétés dans le code



La fenêtre de l'IDE apparait donc, c'est ici que vous écrirez votre code.



L'utilisation des GPIOs implique d'utiliser la bibliothèque GPIO il faut donc l'importer comme décrit dans la fiche pratique 5. On écrit donc :

Import RPi.GPIO as GPIO. # importe la bibliothèque Rpi.GPIO un objet GPIO

Concernant l'état des entrées-sorties (E/S), le module RPi.GPIO accepte des variables dédiées, des entiers ou des booléens.

Ainsi, l'état haut peut valoir au choix **GPIO.HIGH, 1** ou encore **True**. De même, l'état bas peut valoir **GPIO.LOW, 0** ou **False**.

La Raspberry Pi autorise deux numérotations : celle de la sérigraphie du connecteur de la carte (**GPIO.BOARD**), ou la numérotation électronique de la puce (**GPIO.BCM**). À vous de choisir celle que vous voulez.



En noir la numérotation GPIO.BOARD. En rouge, la numérotation GPIO.BCM

Dans votre code il faudra donc définir le mode que vous souhaitez utiliser en utilisant les commandes :

GPIO.setmode (GPIO.BOARD) (numérotation en noir)

GPIO.setmode (GPIO.BCM) (numérotation en rouge)

Enfin dans le code afin d'initialiser une E/S, il suffit de préciser son numéro, fonction de la configuration précédemment choisie, son paramétrage (entrée ou sortie) et éventuellement son état initial (pour une sortie uniquement) par les codes suivants :

GPIO.setup (12, GPIO.IN) # broche 12 est une entree numerique

GPIO.setup (12, GPIO.OUT) # broche 12 est une sortie numerique

GPIO.setup (12, GPIO.OUT, initial=GPIO.HIGH)# broche 12 est une sortie initialement a l'etat haut

A l'inverse, si le GPIO est une entrée et que l'on désire lire son état comme par exemple l'état d'un bouton (on= on effectue une chose, off=on ne fait rien) on utilise la fonction :

GPIO.input(12) # on interroge la broche 12 pour son état.

Pour changer l'état d'une sortie numérique, c'est-à-dire la faire passer de High à Low ou de LOW à HIGH on utilise le code suivant :

GPIO.output (12, GPIO. LOW) # la broche 12 est mise à 0 BAS

GPIO.output (12.GPIO.HIGH) # la broche 12 est mise à 1 HAUT

Il existe aussi une possibilité pour inverser l'état d'une broche de sortie par la fonction :

GPIO.output(12, not GPIO.input(12)) #inverse l'état de la pin 12

Dans certain programme il sera aussi nécessaire de connaître la configuration d'une entrée/sortie numérique. Les valeurs renvoyées sont alors **GPIO.INPUT, GPIO.OUTPUT, GPIO.SPI, GPIO.I2C, GPIO.HARD_PWM, GPIO.SERIAL** ou **GPIO.UNKNOWN**

state=GPIO.gpio_funtion(12) #on stocke dans une variable state la configuration de la broche 12

print(state) #on écrit dans la console la configuration de la broche

enfin en fin de programme, je vous conseille d'effectuer une purge des ressources afin de remettre tout en place comme au départ. Pour cela on utilise la fonction :

GPIO.cleanup() #nettoie toutes les entrées/sortie utilisés

Pour plus de renseignement sur l'utilisation des GPIOs je vous conseille le site suivant :

<https://deusyss.developpez.com/tutoriels/RaspberryPi/PythonEtLeGpio/>