



# Raspberry Pi - Le port GPIO

[Mise à jour le 7/7/2019]

- **Sources**

- Python et le port GPIO : [Developper.com](https://developper.com)
- Un module pour contrôler les entrées, sorties du Raspberry pi : [RPi.GPIO x.x.x](#)
- Exemples et documentation : [raspberry-py-gpio-python](#)

- **Opérations nécessaires à la mise en oeuvre du port GPIO Python**

Pour mettre en oeuvre le port GPIO en Python :

1. Mettre à jour le module RPi.GPIO (préinstaller dans Raspbian)
2. Dans le programme :
  1. Importer le module Rpi.GPIO
  2. Identifier le mode de numérotation des broches utilisées
  3. Configurer les broches (entrée, sortie)
  4. Lire ou écrire les broches

Ces opérations sont décrites ci-dessous.



- **IDE utilisé**

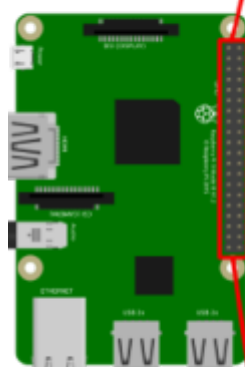
- Environnement de développement intégré disponible avec Raspbian : [Thonny](#)

---

## 1. Aspect matériel

### 1.1 Le connecteur GPIO du raspberry Pi 3

Les broches GPIO<sup>1)</sup> du Raspberry Pi 3 sont reliées à un connecteur **40 points**. **26 broches GPIO sont accessibles** (GPIO2 à GPIO27). Ce sont des entrées/sorties<sup>2)</sup> logiques capables de fournir ou de recevoir des signaux sous la forme de tensions (0 ou 3,3V). Certaines broches permettent de constituer un bus (I2C, SPI, UART).



3.3V PWR	1		2	5V PWR
GPIO2 (SDA1 , I2C)	3		4	5V PWR
GPIO3 (SCL1 , I2C)	5		6	GND
GPIO4 (GPIO_GCLK)	7		8	(UART_TXD0) GPIO14
GND	9		10	(UART_RXD0) GPIO15
GPIO17 (GPIO_GEN0)	11		12	(GPIO_GEN1) GPIO18
GPIO27 (GPIO_GEN2)	13		14	GND
GPIO22 (GPIO_GEN3)	15		16	(GPIO_GEN4) GPIO23
3.3V PWR	17		18	(GPIO_GEN5) GPIO24
GPIO10 (SPI0_MOSI)	19		20	GND
GPIO9 (SPI0_MISO)	21		22	(GPIO_GEN6) GPIO25
GPIO11 (SPI0_CLK)	23		24	(SPI_CE0_N) GPIO8
GND	25		26	(SPI_CE1_N) GPIO7
ID_SD (I2C EEPROM)	27		28	ID_SC (I2C EEPROM)
GPIO5	29		30	GND
GPIO6	31		32	GPIO12
GPIO13	33		34	GND
GPIO19	35		36	GPIO16
GPIO26	37		38	GPIO20
GND	39		40	GPIO21

## 2. Aspect logiciel

### 2.1 Mise à jour du module RPi.GPIO

Le module [RPi.GPIO](#) facilite l'accès aux GPIO en Python. Il est installé par défaut dans Raspbian. Pour effectuer une mise à jour, consulter ce [lien](#).



Ce module n'est pas prévu pour des applications temps réel.

### 2.2 Numérotation des broches dans le programme

Dans RPi.GPIO, il existe deux façons de numéroté les broches d'entrée / sortie dans un programme Python.

GPIO26	37
GND	39

- La première consiste à utiliser le système de **numérotation de la carte**. On fait référence aux numéros des broches du connecteur. Ainsi, quelle que soit la révision de la carte du RPi, il ne sera pas nécessaire de recâbler le connecteur ou de changer le code.

\*.py

```
GPIO.setmode(GPIO.BOARD)
```

Exemple 1 : une LED est connectée à la broche 37 du connecteur

\*.py

```
LED = 37
GPIO.setmode(GPIO.BOARD) # On précise que 37 est le numéro de la broche
                             du connecteur
GPIO.setup(LED,GPIO.OUT) # Cette broche doit se comporter comme une
                             sortie
```

- Le deuxième système de numérotation utilise les **numéros BCM**. On fait référence aux numéros de canaux sur le Broadcom SOC. Si on change le Soc, il sera peut être nécessaire de modifier la numérotation.

\*.py

```
GPIO.setmode(GPIO.BCM)
```

Exemple 2 : une LED est connectée à la broche 37 du connecteur mais repéré par la fonctionnalité du SOC

\*.py

```
LED = 26
GPIO.setmode(GPIO.BCM) # On précise que 26 est le numéro de canal du
                             SOC
GPIO.setup(LED,GPIO.OUT) # Cette broche doit se comporter comme une
                             sortie
```

## 2.3 Exemples de mise en oeuvre du GPIO

- **Affichage de la version de RPi.GPIO**

\*.py

```
#!/usr/bin/python3
import Rpi.GPIO as GPIO
print("vers : " + GPIO.VERSION)
print(GPIO.RPI_INFO)
```

- **Résultat attendu** : vers : x.x.x et des informations sur la carte utilisée
- **Configuration d'une broche du GPIO en sortie**
  - **Commande d'une led**

## blink.py

```
import RPi.GPIO as GPIO # module de gestion du GPIO
import time #module de gestion du temps

# Identification de la broche utilisée
LED = 37 # Une LED est connectée sur la broche 37
GPIO.setmode(GPIO.BOARD)
# Autre solution
#LED = 26
#GPIO.setmode(GPIO.BCM)

#Configuration de la broche en sortie
GPIO.setup(LED,GPIO.OUT)

while True:
    GPIO.output(LED,GPIO.HIGH) # Activation de la LED
    time.sleep(1) # Attente 1s
    GPIO.output(LED,GPIO.LOW) # Désactivation de la LED
    time.sleep(1)
```

- **Résultat attendu** : la LED clignote
- **Configuration d'une broche du GPIO en entrée**
  - **Lecture de l'état d'un bouton-poussoir**  
A faire

## Pour aller plus loin...

PWM, Interruptions, I2C, SPI : [Developper.com](https://webge.fr/dokuwiki/doku.php?id=raspberrypi:linux:gpio)

1)

Les ports **GPIO** (anglais : **G**eneral **P**urpose **I**nput/**O**utput, littéralement Entrée-sortie à **usage général**) sont des ports d'entrées-sorties très utilisés dans le monde des microcontrôleurs, en particulier dans le domaine de l'électronique embarquée.[Wikipédia](https://webge.fr/dokuwiki/doku.php?id=raspberrypi:linux:gpio)

2)

Dans un système à base de processeur, de microprocesseur, de microcontrôleur ou d'automate, on appelle entrées-sorties les échanges d'informations entre le processeur et les périphériques qui lui sont associés. De la sorte, le système peut réagir à des modifications de son environnement, voire le contrôler. Elles sont parfois désignées par l'acronyme I/O, issu de l'anglais Input/Output ou encore E/S pour Entrées/Sorties. [Wikipédia](https://webge.fr/dokuwiki/doku.php?id=raspberrypi:linux:gpio)

From:

<https://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:

<https://webge.fr/dokuwiki/doku.php?id=raspberrypi:linux:gpio>

Last update: **2021/08/12 08:25**

