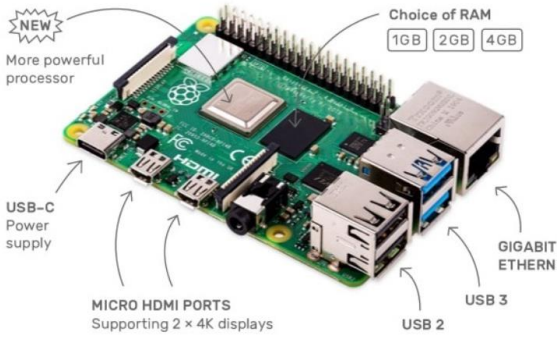
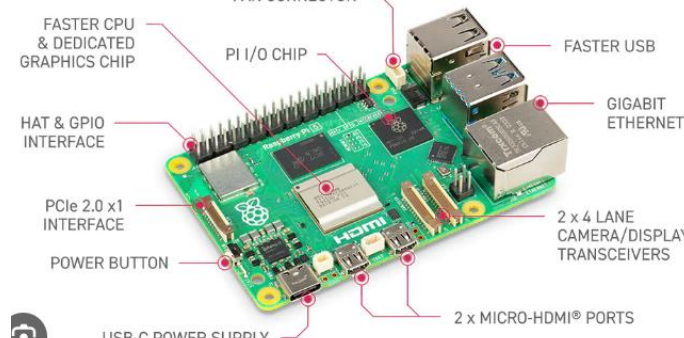


# Chapitre 5: Objets IOT à base de carte µProcesseur(SBC) : cas de Raspberry Pi

Le Raspberry Pi est un nano-ordinateur mono-carte à processeur ARM conçu à l'université de Cambridge. Cet ordinateur, qui a la taille d'une carte de crédit, est destiné à encourager l'apprentissage de la programmation informatique ; il peut fonctionner avec plusieurs variantes du système d'exploitation libre GNU/Linux (notamment Debian : Raspbian) et des logiciels compatibles. Mais il fonctionne également avec l'OS Microsoft Windows : Windows 10 IoT Core3 et celui de Google, Android Pi4. Le Raspberry Pi 4 dispose de ports USB type C, d'un SoC avec CPU quad-core 1,5 GHz, GPU VideoCore VI prenant en charge OpenGL ES 3.0, décodage HEVC 4K à 60 i/s, deux ports micro HDMI prenant en charge des écrans 4K, et de 1 Go jusqu'à 8 Go de RAM.

## Les composants du Raspberry Pi 4/Pi 5 :

Raspberry Pi 4	Raspberry Pi 5
 <p>NEW More powerful processor</p> <p>Choice of RAM 1GB 2GB 4GB</p> <p>USB-C Power supply</p> <p>MICRO HDMI PORTS Supporting 2 x 4K displays</p> <p>USB 2</p> <p>USB 3</p> <p>GIGABIT ETHERNET</p>	 <p>FASTER CPU &amp; DEDICATED GRAPHICS CHIP</p> <p>FAN CONNECTOR</p> <p>PI I/O CHIP</p> <p>HAT &amp; GPIO INTERFACE</p> <p>PCle 2.0 x1 INTERFACE</p> <p>POWER BUTTON</p> <p>USB-C POWER SUPPLY</p> <p>FASTER USB</p> <p>GIGABIT ETHERNET</p> <p>2 x 4 LANE CAMERA/DISPLAY TRANSCEIVERS</p> <p>2 x MICRO-HDMI® PORTS</p>

## Caractéristiques :

La version 4	La version 5
<p>processeur ARM v8 Broadcom BCM2711 4 coeurs Cortex-A72 à 1,5 Ghz</p> <p>mémoire vive de 1, 2 ou 4 Go de LPDDR4 2.4 Ghz selon le modèle</p> <p>bus PCIe (PCI Express) Gen2 pour transférer les données USB et Ethernet</p> <p>GPU VideoCore VI</p> <p>2 ports micro HDMI 2.0 type D supportant des résolutions jusqu'à 4K</p> <p>alimentation via un port USB-C (5v 3A)</p> <p>Ethernet 1 Go</p> <p>2 ports USB 3</p> <p>WiFi 802.11 b / g / n / ac</p> <p>Bluetooth 5</p>	<p>CPU - ARM</p> <ul style="list-style-type: none"> <li>- Broadcom BCM2712 2.4GHz (64 bit)</li> <li>- Cortex-A76</li> <li>- Extensions cryptographiques</li> <li>- 512 Ko cache L2 par coeur</li> <li>- 2 Mo cache L3 partagé</li> </ul> <p>RAM 4 Go / 8 Go LPDDR4X-4267 SDRAM-</p> <p>GPU - VIDEOCORE VII</p> <ul style="list-style-type: none"> <li>- 2x transceivers MIPI 4 cahaux à 1,5 Gbps</li> <li>- Vidéo jusqu'à 4K60fps</li> <li>- HDR</li> <li>- OpenGL</li> <li>- Vulkan</li> </ul> <p>Stockage</p> <ul style="list-style-type: none"> <li>- via microSD</li> <li>- Prise en charge du mode de bande passante élevée SDR104</li> </ul> <p>Ports USB</p> <ul style="list-style-type: none"> <li>- 2x USB 3.0 (jusqu'à 5 Gbps chacun en simultané)</li> <li>- 2x USB 2.0</li> </ul> <p>Ethernet - 1x GbE avec PoE</p> <p>Connectivité sans fil</p>

	<ul style="list-style-type: none"> <li>- WiFi 802.11ac dual band</li> <li>- Bluetooth 5.0</li> <li>- BLE</li> </ul> <p>Interfaces</p> <ul style="list-style-type: none"> <li>- GPIO 40 pins</li> <li>- 2x transceivers MIPI</li> <li>- PCIe 2.0 (via Hat M.2 ou autre adaptateur)</li> </ul> <p>Alimentation - 5V/5A via USB-C ou - PoE</p>
--	---

## Installation et mise en marche de la carte Raspberry Pi

### Outils matériels et logiciels nécessaires

1. Carte Raspberry Pi ,
2. Adaptateur d'alimentation
3. Une carte MicroSD catégorie 10 (minimum 16 Go)
4. Logiciel pour flasher le système d'exploitation Raspbian sur la carte SD ( Raspberry Pi Imager)
5. Image du système d'exploitation choisi
6. Putty
7. VNC viewer

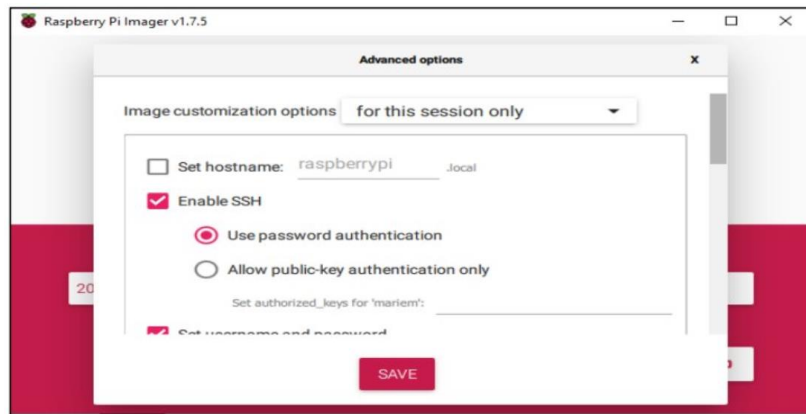
De nombreux systèmes d'exploitation sont disponibles pour Raspberry Pi, y compris le système d'exploitation Raspberry Pi. « [Raspberry Pi Imager](#) » est le moyen rapide et facile d'installer un système d'exploitation sur une carte microSD prête à être utilisée avec votre Raspberry Pi.

Pour la mise en marche il faut soit :

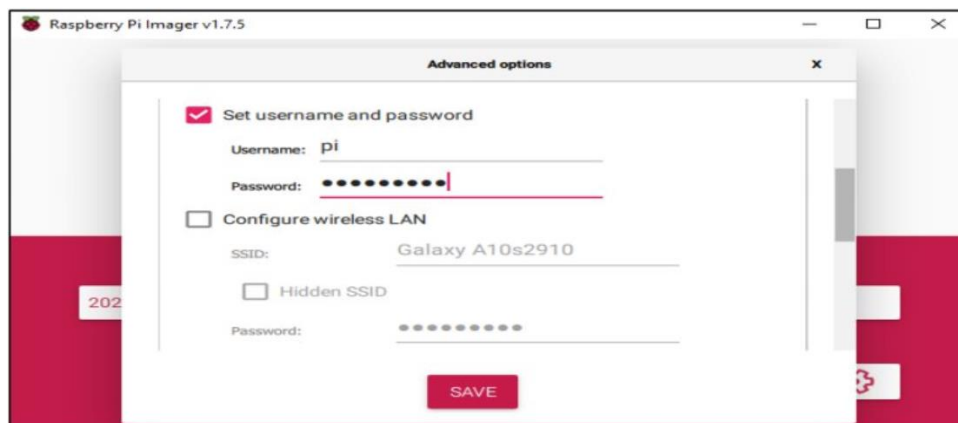
- un écran avec port HDMI, Clavier, souris et câble MicroHDMI - HDMI ou bien
- un émulateur de terminal (Putty, MobaxTerm...) sur PC doté d'un client pour les protocoles SSH, Telnet...

### Etapes d'installation :

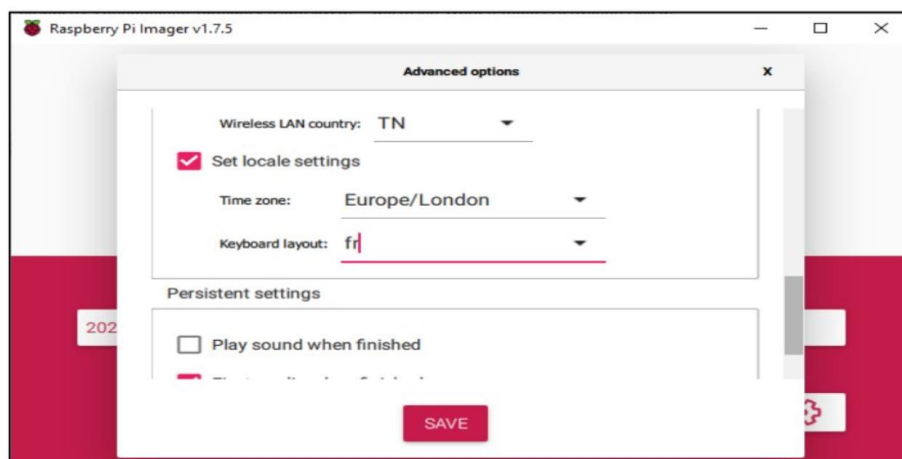
1. Télécharger et installer le [Raspberry Pi Imager](#)
2. Télécharger le système d'exploitation pour la carte Raspberry  
<https://www.raspberrypi.org/software/>
3. Placer la carte SD dans le lecteur du PC
4. Avant de flasher le système d'exploitation, cliquer sur le bouton configuration
5. Saisir la configuration suivante
  - Activez la communication SSH afin d'établir une connexion SSH à distance avec le raspberry pi.  
Notez ici que le nom d'hôte par défaut est « raspberrypi.local »



- Saisir le nom du compte et le mot de passe, par défaut utiliser le compte : *pi* et le mot de passe : *raspberry*
- Saisir les paramètres Wifi de ton WLAN(Optionnel)



- 
- Spécifier le pays et le type de clavier qui doit être utilisé par la carte Raspberry



- Sauvegarder le paramétrage



- Ecrire le système d'exploitation sur la carte

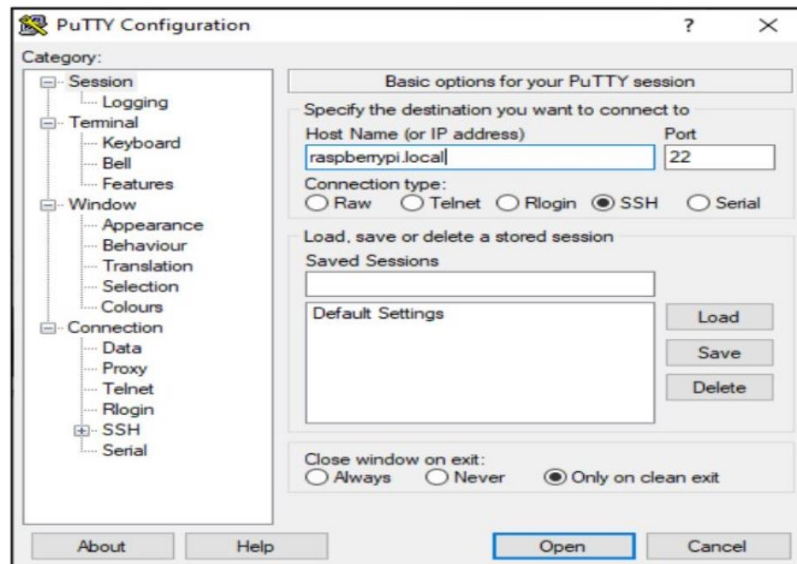
8) Une fois les processus d'écriture et de vérification terminés avec succès, insérez la carte SD dans la plate-forme.

9) il est possible de **manipuler localement** la carte Raspberru en branchant un clavier une souris et un écran (ou vidéo projecteur) avec un câble micro-hdmi et allumez votre Raspberry Pi.

10) Afin **de manipuler à distance** la carte Raspberry pi, il est possible d'établir une communication graphique avec l'outil de visualisation VNC via l'ordinateur de l'utilisateur. Pour ce faire, il faut donc activer l'option VNC sur la carte Raspberry. Pour ce faire, il est possible d'établir une communication SSH entre l'ordinateur exécutant le logiciel PuTTY d'un côté, et le Raspberry pi de l'autre côté. Ici il y a deux méthodes : si vous connaissez l'adresse IP de votre Raspberry, ou si vous ne connaissez pas l'adresse IP de votre Raspberry.

a- Si vous ne connaissez pas l'adresse IP car votre carte n'est pas connectée à un réseau sans fil. Dans ce cas, vous devez établir une connexion avec un câble RJ45 entre l'ordinateur et le Raspberry. Ensuite, ouvrez le logiciel Putty et mettez le nom du nom d'hôte

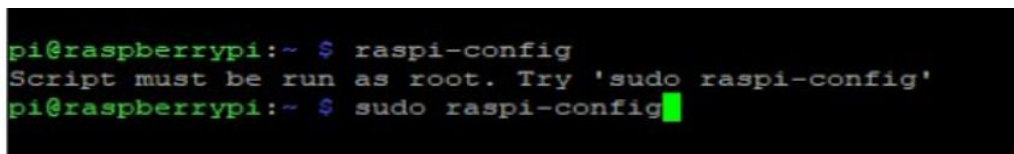
b- Si vous connaissez l'adresse IP de votre carte(connectée à un réseau sans fil), saisir l'adresse IP dans PuTTY



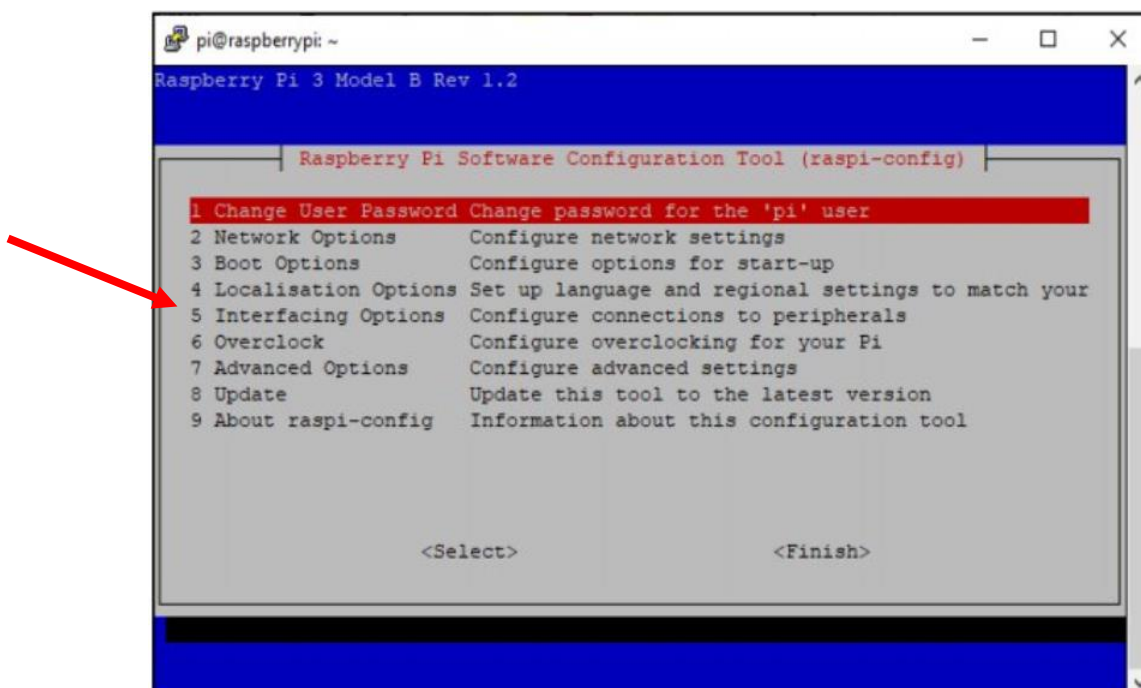
Ensuite le système demande le nom du compte et le mot de passe



Lancer la commande : *raspi-config*



Choisir « interfacing options » et activer le VNC



## Les ports d'entrée/sortie GPIO:

Les périphériques GPIO comportent un ensemble de ports d'entrée/sortie, ces port sont indispensables pour des systèmes embarqués, en effet ils peuvent fonctionner aussi bien en entrée qu'en sortie.

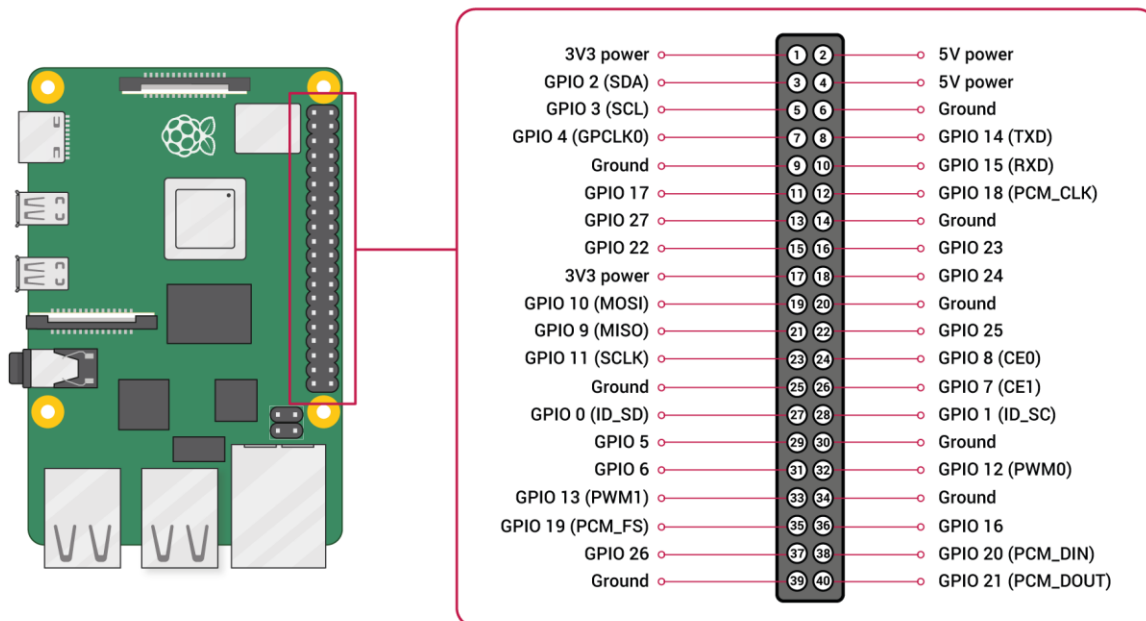
Ils sont à usage multiple :

- en **entrée numérique** tout ou rien (1 ou 0), par exemple pour détecter l'état d'un interrupteur
- en **sortie numérique** tout ou rien (1 ou 0), pour activer un relais par exemple
- en **sortie numérique MLI2**, pour contrôler la puissance moyenne d'une Diode électroluminescente (LED)
- en protocole I2C, permet d'échanger avec une ou plusieurs puces
- en protocole SPI, idem
- en protocole UART, d'échanger avec une seule puce (ou un PC)

D'autres usages sont possibles (audio PCM, vidéo sur les connecteurs DSI et CSI).

Plusieurs connecteurs donnent accès aux GPIO, mais le principal est un connecteur comportant deux rangées de 13 picots mâles (raspberry pi 1) ou 20 picots mâles (raspberry pi 3).

Pour les premier 26 broches, 17 broches sont dédiées au GPIO (permetts des changements d'état on/off). Les autres broches ont le rôle d'alimentation (3.3V et 5V) et de masse (ground).



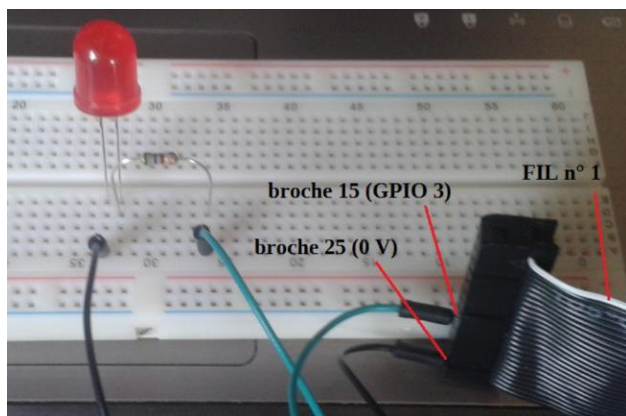
**NB :** le numéro de GPIO n'est pas sa position sur le connecteur, mais son numéro dans les registres de la puce ARM BCM2835. C'est donc celui qui sera utilisé dans la plupart des bibliothèques d'accès.



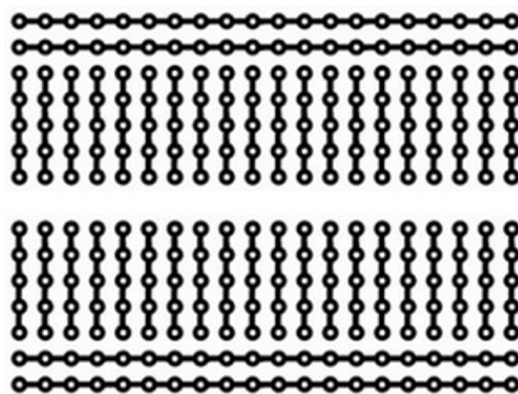
```
pi@raspberrypi:~ $ gpio readall
```

Pi Zero											
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
		3.3v			1	2		5v			
2	8	SDA.1	ALT0	1	3	4		5V			
3	9	SCL.1	ALT0	1	5	6		0v			
4	7	GPIO. 7	IN	1	7	8	1	ALT0	TxD	15	14
		0v			9	10	1	ALT0	RxD	16	15
17	0	GPIO. 0	IN	0	11	12	0	IN	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13	14		0v			
22	3	GPIO. 3	IN	0	15	16	0	IN	GPIO. 4	4	23
		3.3v			17	18	0	IN	GPIO. 5	5	24
10	12	MOSI	IN	0	19	20		0v			
9	13	MISO	IN	0	21	22	0	IN	GPIO. 6	6	25
11	14	SCLK	IN	0	23	24	1	IN	CE0	10	8
		0v			25	26	1	IN	CE1	11	7
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31	1
5	21	GPIO.21	IN	1	29	30		0v			
6	22	GPIO.22	IN	1	31	32	0	IN	GPIO.26	26	12
13	23	GPIO.23	IN	0	33	34		0v			
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27	16
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28	20
		0v			39	40	0	IN	GPIO.29	29	21
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
Pi Zero											

**Exercice d'application :** on connecte la Raspberry au montage ci-contre, avec un signal d'entrée/sortie (interrupteur ou LED) sur la patte 15 du connecteur, correspondant à la GPIO 3 de la Raspberry Pi. Le script blink va faire clignoter le LED à une fréquence de 60 Hz.



Montage



Connexion interne d'une plaque d'expérimentation

## **Code en script Shell :**

```
#!/bin/bash
```

```
setup()
```

```
{  
    PIN=3  
    gpio mode $PIN out  
}
```

```
# basculer entre l'état on et off
```

```
basculer()
```

```
{  
    gpio write $PIN 1  
    sleep 0.5  
    gpio write $PIN 0  
    sleep 0.5  
}
```

```
# programme principal
```

```
setup
```

```
while true; do
```

```
    basculer
```

```
done
```



## Code en code C/C++:

La bibliothèque fournit avec WiringPi permet également de programmer en langage C/C++. Voici le code le programme **blink.c**

```
#include <stdio.h>
#include <wiringPi.h>

// définition des entrée/sortie
const int LED = 3; // LED Pin - wiringPi pin 3 is BCM_GPIO 15

void setup()
{
    printf("Raspberry Pi blink \n");
    pinMode(LED, OUTPUT);
}

void basculer()
{
    digitalWrite(LED, HIGH); // On
    delay (500); // 500 milli-seconde

    digitalWrite(LED, LOW); // Off
    delay (500); // 500 milli-seconde
}

//programme principal
int main (void)
{ wiringPiSetup();
  setup();

  For(;;)
    basculer();
  return;
}
```

Pour compiler le programme blink.c, il faut lancer la commande **make** et ensuite lancer l'exécution du programme compilé avec les privilèges du compte **root**.

```
gcc -Wall -o blink blink.c -lwiringPi
sudo ./blink
```

**make blink**  
**sudo ./blink**

## **Code en python**

Le pilotage des ports GPIO, sous Python s'appuie sur des bibliothèques. Les bibliothèques **RPi.GPIO** et **gpiozero** sont incluses dans les versions de Python. Nous utiliserons également les bibliothèques **pigpio** et **wiringpi** qui devront être installées sous Python.

Avec un éditeur de texte ouvre un fichier texte vide appelé blink.py Voici Le contenu du programme :

```
#!/usr/bin/python3
# -*- coding:utf-8 -*-
import RPi.GPIO as GPIO          #bibliothèque RPi.GPIO
import time                      #bibliothèque time
#-----
GPIO.setwarnings(False)         #désactive le mode warning
GPIO.setmode(GPIO.BCM)          #utilisation des numéros de ports du processeur
GPIO.setup(22, GPIO.OUT)         #mise en sortie du port GPIO 3 (broche #15 du connecteur)
while True :                    #boucle infinie
    GPIO.output(22, GPIO.HIGH)   #sortie 22 high
    time.sleep(1)               #attente 1 seconde
    GPIO.output(22, GPIO.LOW)    #sortie 22 low
    time.sleep(1)               #attente 1 seconde
```

### **ou bien on peut se référer au numéro de broche physique sur le board**

```
#!/usr/bin/python3
# -*- coding:utf-8 -*-
import RPi.GPIO as GPIO          #bibliothèque RPi.GPIO
import time                      #bibliothèque time
#-----
GPIO.setwarnings(False)         #désactive le mode warning
GPIO.setmode(GPIO.BOARD)         #utilisation des numéros de ports du processeur
GPIO.setup(15, GPIO.OUT)         #mise en sortie du port GPIO 3 (broche physique #15 du connecteur)
while True :                    #boucle infinie
    GPIO.output(A, GPIO.HIGH)    #sortie 15 high
    time.sleep(1)               #attente 1 seconde
    GPIO.output(15, GPIO.LOW)    #sortie 15 low
    time.sleep(1)               #attente 1 seconde
```