

# Annexe : Installation de Raspberry OS sur Raspberry Pi 4

CoVAPSy : Premiers programmes python sur la voiture réelle

Culture Sciences de l'Ingénieur

La Revue  
3E.I

Anthony JUTON<sup>1</sup>

Édité le  
16/11/2023

école normale supérieure paris-saclay

<sup>1</sup> Professeur agrégé de physique appliquée au DER Nikola Tesla, ENS Paris-Saclay

Cette ressource est une annexe à la ressource « CoVAPSy : Premiers programmes python sur la voiture réelle » [1] et fait partie du N° 111 de La Revue 3EI de janvier 2024. Cette annexe est mise à jour au moment de sa publication, les mises à jour ultérieures sont disponibles sur [2].

Le nano-ordinateur Raspberry Pi 4 peut être utilisé comme un PC avec écran et clavier. Il est utilisé ici comme un serveur « Headless », c'est à dire sans écran ni clavier. On y accède par un bureau virtuel ou par une console depuis un PC personnel. Ce document présente l'installation de Raspberry OS et les outils nécessaires pour un accès au nano-ordinateur, sans écran ni clavier.

Les informations concernant la carte Raspberry Pi 4 sont disponibles sur le site de Raspberry : <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>

## 1 - Installation de l'OS

### 1.1 - Création de l'image

Le site Raspberry explique en détail comment installer le système d'exploitation sur la PI 4.

Pour installer Raspberry OS sur la carte, on crée une image du système d'exploitation sur la carte SD du nano-ordinateur via un PC personnel.

Pour créer cette image, on insère la carte SD dans le PC et on commence par télécharger Raspberry Pi Imager : <https://www.raspberrypi.com/software/>

On choisit Raspberry OS 64bits et la carte SD.



Figure 1 : Fenêtre du logiciel Raspberry Imager avant écriture de la carte SD

Le bouton *Suivant* puis l'option *Modifier les réglages* permettent d'ajouter une configuration de connexion au réseau WIFI (ce peut être un point d'accès WIFI créé par un smartphone) en indiquant

un nom de réseau Wifi (SSID) et la clé WPA associée (password). Activer également SSH (pour un accès à distance)

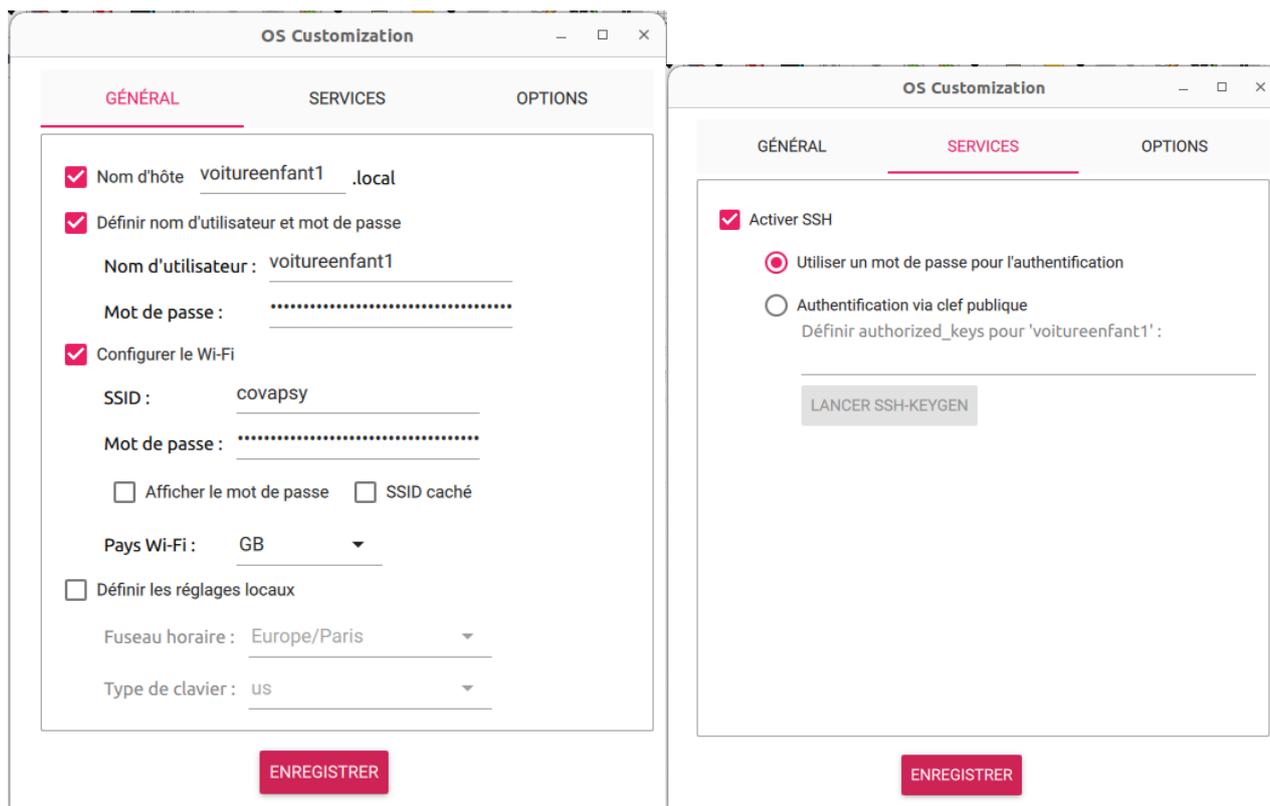


Figure 2 : Paramètres de l'image Raspberry OS

Attention, le nano-ordinateur se trouvant connecté à Internet, il est nécessaire de ne surtout pas laisser le login pi associé au mot de passe raspberry.

La configuration est terminée. Générer la carte (bouton *Ecrire*) et l'insérer dans la raspberry pi.

## 1.2 - Recherche de l'IP du nano-ordinateur

Une fois ces configurations effectuées, le nano-ordinateur démarre. Pour s'y connecter à distance, il est nécessaire d'avoir son adresse IP, adresse notée sous la forme de 4 nombres 192.168.1.4 par exemple (en IPv4).

Le PC doit être bien évidemment connecté sur le même réseau local.

- **Une solution simple** est de faire un *ping* avec une recherche DNS locale, depuis une console (nommée *invite de commande* sous windows) avec le hostname défini dans la partie précédente : `ping voitureenfant1.local`  
Le nano-ordinateur répond alors avec son adresse IP.

```
ajuton@juton-Dell-5410:~$ ping voitureenfant1.local
PING voitureenfant1.local (192.168.1.46) 56(84) bytes of data:
64 bytes from voitureenfant1.home (192.168.1.46): icmp_seq=1 ttl=64 time=2.55 ms
```

Figure 3 : Réponse à une recherche DNS locale

- **Si le nano-ordinateur est connecté sur un réseau Ethernet filaire ou wifi avec un serveur DHCP** (c'est-à-dire qu'une machine, le routeur ou la box habituellement sur les petits réseaux, distribue les adresses IP), on peut y trouver l'adresse IP du nanoordinateur. Pour

cela, on se rend sur la page web de la box ou du routeur (la page web de la box internet pour une utilisation à domicile, souvent accessible à l'adresse 192.168.1.1). On présente ici les pages d'une LiveBox. Les routeurs grand public et les box ont la même apparence avec une vue globale (*overview*) du réseau où on trouve les IP des équipements connectés et des pages de configuration avancée (parfois nommées *LAN network*), notamment pour la réservation des adresses IP.

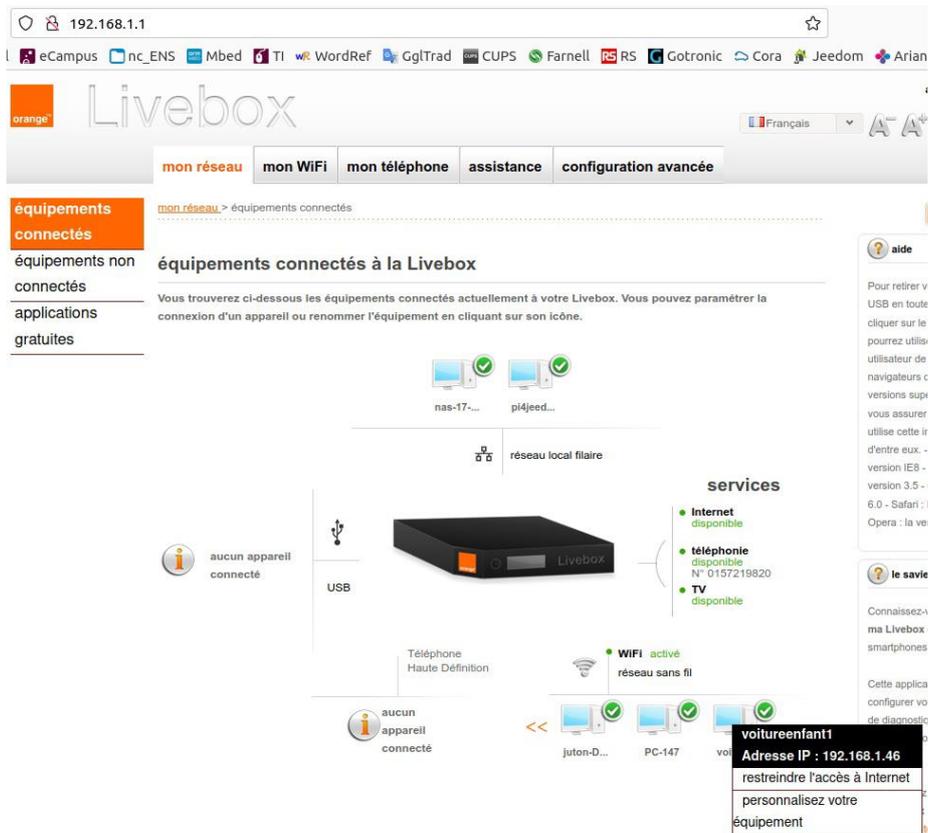


Figure 4 : Présentation des équipements connectés au réseau par une LiveBox

Il est alors possible de réserver l'adresse IP dans le serveur DHCP, de sorte que le nano-ordinateur obtienne toujours la même adresse lors de ces nouvelles connexions.

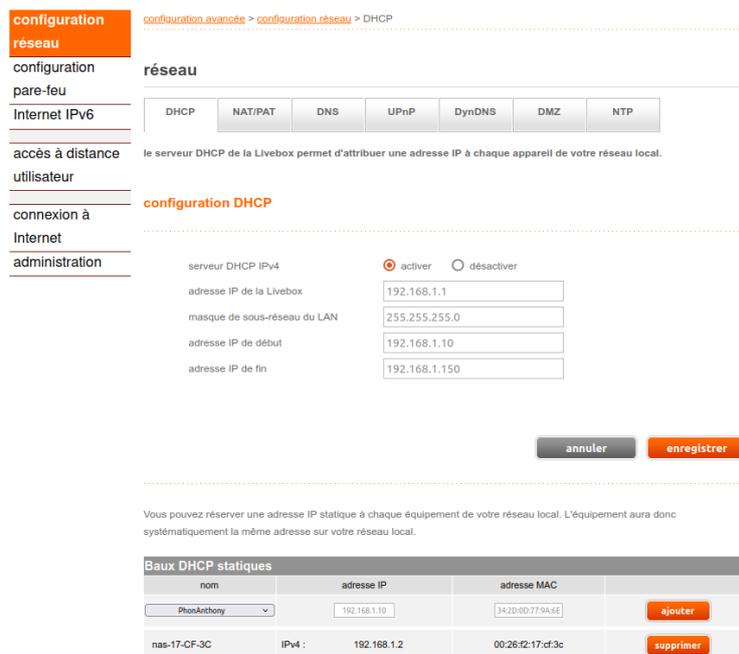


Figure 5 : Page de configuration DHCP d'une livebox où sont attribuées les adresses IP statiques (ou IP réservées)

## 1.3 - Connexion SSH

Une fois l'IP découverte, il est aisé de se connecter en *SSH (Secure Shell)*. *SSH* est un protocole de communication sécurisée permettant de dialoguer via une console du *serveur SSH* ouverte sur le *client SSH*. Le nano-ordinateur est *serveur SSH* et le PC utilisateur est *client SSH*.

Très utile pour les accès à distance, le *serveur SSH* est installé par défaut sur les Raspberry Pi (et il a été activé dans les options lors de la création de la carte SD).

Le *client SSH* est installé sur les PC Linux (et sans doute Mac). Sur les PC Windows, on peut installer Putty par exemple.

Depuis une console (terminal Linux ou Putty sous Windows), on tape :

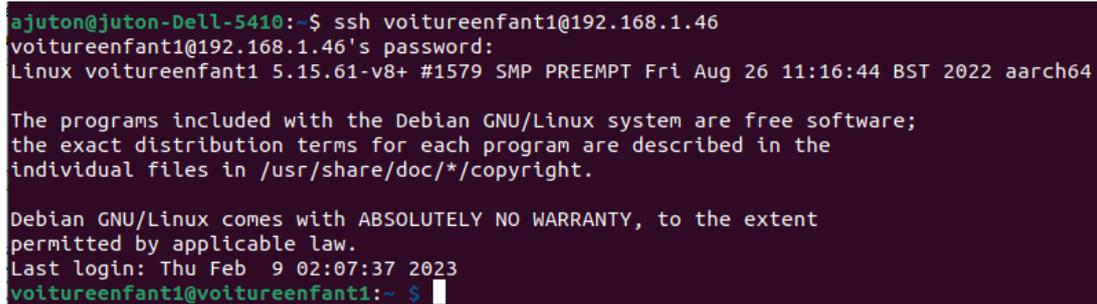
```
ssh login@<adresse IP de la raspberry pi>
```

L'authenticité de ce serveur SSH n'étant pas attesté, une demande de confirmation attend la réponse *yes*, lors de la première connexion, puis le mot de passe de l'utilisateur.

Lorsque la connexion est réussie, le prompt indique :

`le_nom_de_l_utilisateur@le_nom_de_la_machine`, ici `voitureenfant1@voitureenfant1`

La commande *exit* permet de couper la connexion.



```
ajuton@juton-Dell-5410:~$ ssh voitureenfant1@192.168.1.46
voitureenfant1@192.168.1.46's password:
Linux voitureenfant1 5.15.61-v8+ #1579 SMP PREEMPT Fri Aug 26 11:16:44 BST 2022 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Feb  9 02:07:37 2023
voitureenfant1@voitureenfant1:~$
```

Figure 6 : Connexion ssh à la Raspberry pi depuis un PC relié sur le même réseau local

## 2 - Bureau à distance

Pour faciliter l'utilisation du nano-ordinateur, les outils graphiques sont plus accessibles que la ligne de commande.

### 2.1 Mise en œuvre d'un bureau à distance VNC

*VNC (Virtual Network Computing)* est un protocole de bureau à distance. Le *serveur VNC* est la Raspberry Pi et le *client VNC* est le PC utilisateur.

Le site [raspberrypi.com](https://www.raspberrypi.com) propose là encore des solutions :

<https://www.raspberrypi.com/documentation/computers/remote-access.html#virtual-networkcomputing-vnc>

#### 2.1.1 Installation du client sur le PC

Raspberry propose la solution gratuite mais propriétaire de RealVNC. Il est possible de télécharger le client *vnc* (nommé *VNC Viewer*) : sur le site (<https://www.realvnc.com/en/>), onglet *Download*, choisir la version adaptée de *VNC Viewer* à l'OS de l'ordinateur et l'installer.

## 2.1.2 Activation du serveur VNC sur le nano-ordinateur Raspberry Pi

*Realvnc-vnc-serveur* est installé avec l'image de Raspberry OS. Pour l'activer, utiliser la fenêtre de configuration *raspi-config*, depuis l'accès SSH :

```
sudo raspi-config
```

Ensuite, choisir **Interfacing Options > VNC > Yes**.

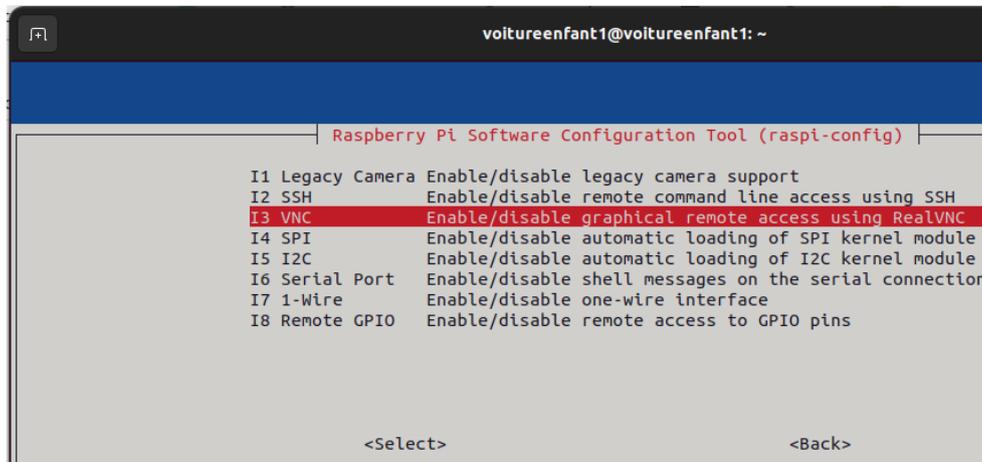


Figure 7 : Extrait de la fenêtre *raspi-config*

## 2.1.3 Connexion locale au serveur vnc

Pour se connecter en VNC depuis un PC connecté sur le même réseau que le nano-ordinateur, lancer VNC Viewer, indiquer l'adresse IP de la Raspberry Pi, accepter les recommandations de précautions, introduire son login et son mot de passe.

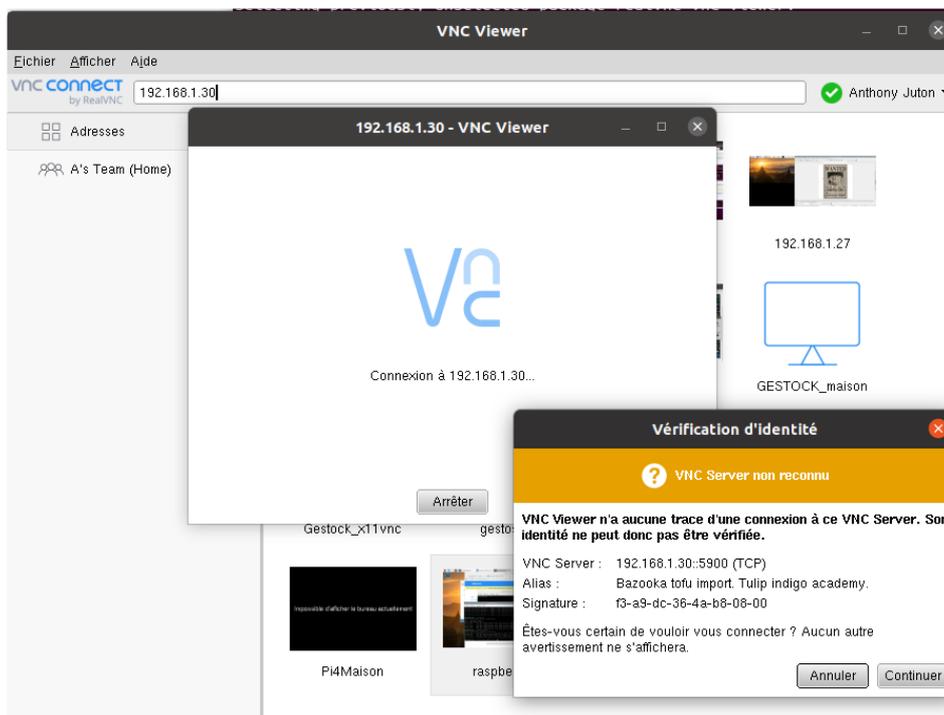


Figure 8 : Client VNC Viewer lors de la 1ère connexion à la Raspberry Pi

On accède alors au bureau à distance, qui lors de la première connexion demande à faire quelques mises à jour. Suite à un redémarrage on peut profiter du bureau à distance du nano-ordinateur.

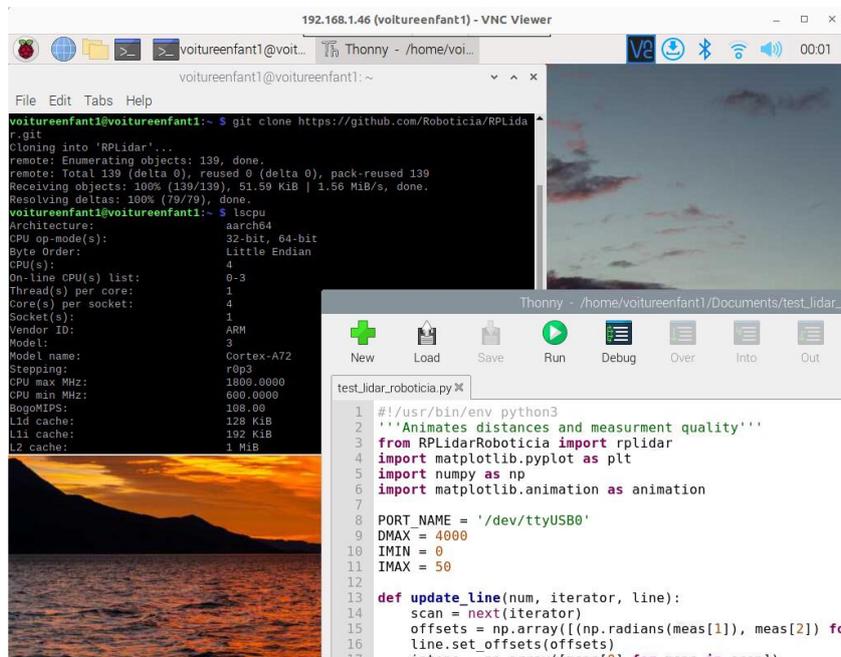
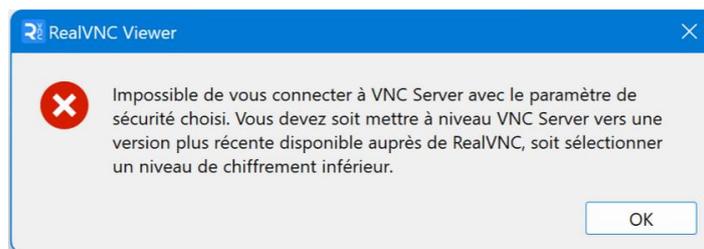


Figure 9 : Bureau à distance de la Raspberry Pi via RealVNC

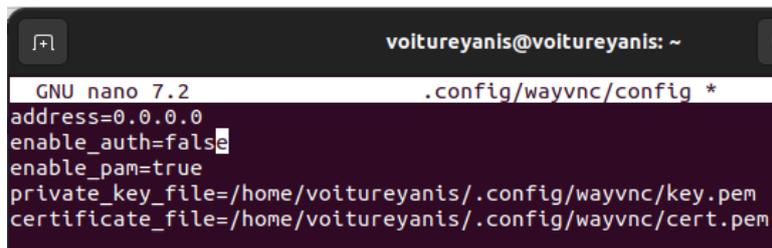
**Remarque :** Sur les dernières versions de Raspberry OS (Bookworm), vnc server demande par défaut un chiffrement des données (peu utile si on fonctionne uniquement sur le réseau local) incompatible avec VNC client.



Pour désactiver ce chiffrement des données, il faut, depuis la connexion SSH décrite précédemment, modifier un fichier de configuration, depuis un terminal :

```
sudo nano .config/wayvnc/config
```

Puis, dans le fichier, modifier `enable_auth` de `True` à `False`.



Ensuite, rebooter et tenter la connexion depuis VNC Viewer

### 3 - Remarques complémentaires

#### 3.1 - Changement de la résolution du bureau à distance

Pour plus de confort, il est possible d'augmenter la résolution du bureau à distance et d'affecter plus de mémoire RAM au GPU, via la fenêtre Raspberry Pi Configuration accessible depuis le menu principal (la framboise en haut à gauche)

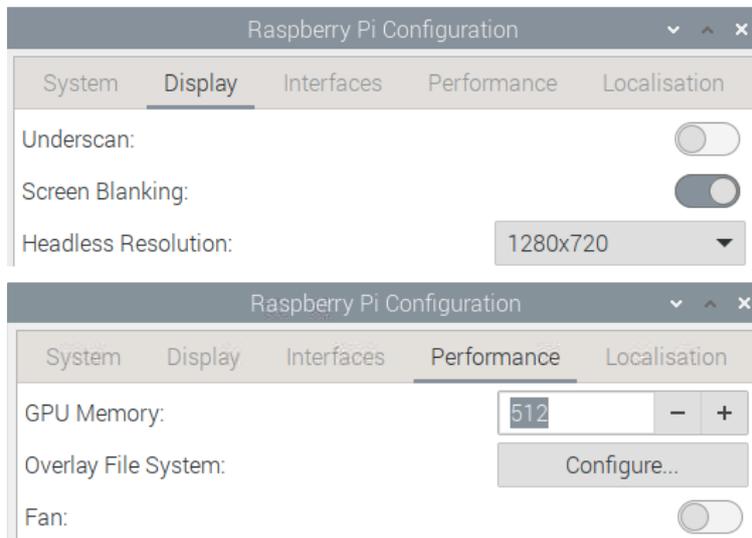


Figure 10 : Extraits de la fenêtre Raspberry Pi Configuration

### 3.2 - Imposer une adresse ip fixe

Pour permettre de se connecter à la Raspberry pi depuis son PC, même sans réseau accessible, il est intéressant d'attribuer une IP fixe à la carte Ethernet (eth0). Sur la dernière version de Raspberry OS (BookWrom), cela se fait depuis l'interface graphique :

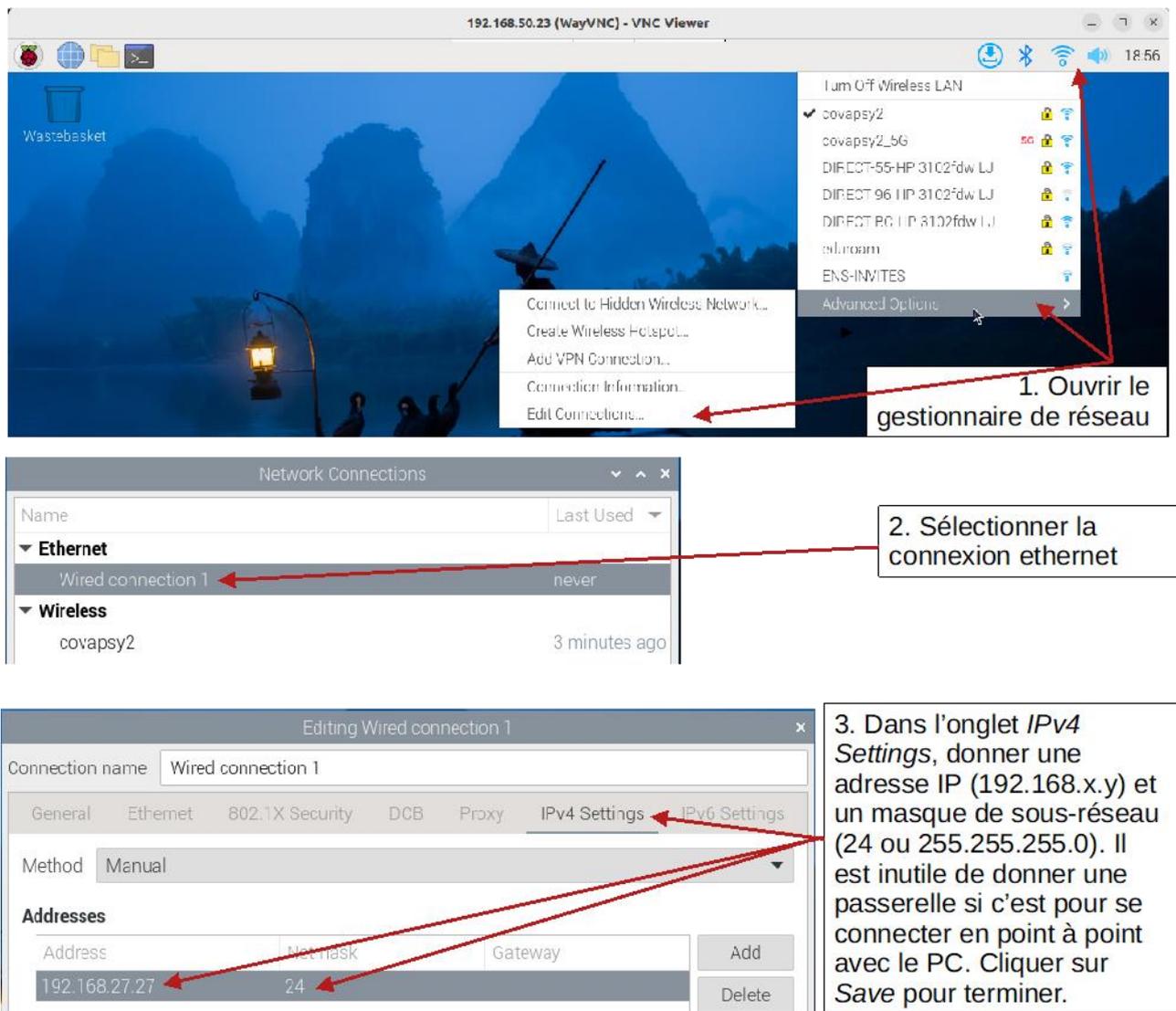


Figure 11 : Configuration d'une adresse IP fixe sur Raspberry OS bookworm

Sur l'ancienne version de Raspberry OS (Buster), cela se fait via le fichier dhcpd.conf. On ouvre le fichier en mode super-utilisateur depuis la console :

```
sudo nano /etc/dhcpd.conf
```

```
GNU nano 5.4 /etc/dhcpd.conf
# Generate SLAAC address using the Hardware Address of the interface
#slaac hwaddr
# OR generate Stable Private IPv6 Addresses based from the DUID
slaac private

# Example static IP configuration:
interface eth0
static ip_address=192.168.1.22/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.1.1
static domain_name_servers=192.168.1.1 8.8.8.8
```

Figure 12 : Configuration d'une adresse IP fixe sur Raspberry OS Buster

Attention à ne prendre que des adresses IP privées (non attribuées à des usages publics). Les adresses de type 192.168.x.y sont des adresses privées.

## 4 - Utiliser les entrées/sorties

Pour utiliser les entrées/sorties et les communications, il faut les activer via la fenêtre Raspberry Pi Configuration.

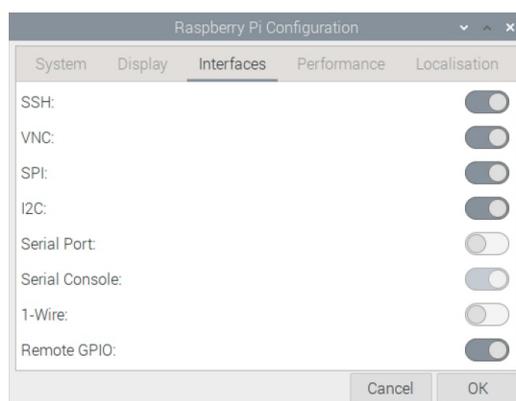


Figure 13 : Activation des entrées/sorties via Raspberry Pi Configuration

Plus d'infos sur <https://gpiozero.readthedocs.io/en/stable/installing.html>

### 4.1 - Installation des paquets utiles

Pour le bon fonctionnement de la voiture, quelques paquets Linux sont utiles. Ils seront présentés plus loin.

Depuis la console, lancer les commandes d'installation suivantes :

- sudo apt update
- sudo apt upgrade
- sudo apt-get install libatlas-base-dev
- pip install matplotlib
- pip install numpy --upgrade
- sudo pip install rpi-hardware-pwm
- sudo apt-get install joystick
- pip install pyPS4Controller
- pip install rplidar-roboticia
- sudo apt install python3-gpiozero

Python demande désormais par défaut d'utiliser des environnements virtuels. La raspberry Pi ne servant qu'à la conduite autonome, on peut installer les modules python hors environnement virtuel, soit en installant le paquet depuis les dépôts linux officiels de la distribution :

```
sudo apt install python3-nomdumodule
```

ou

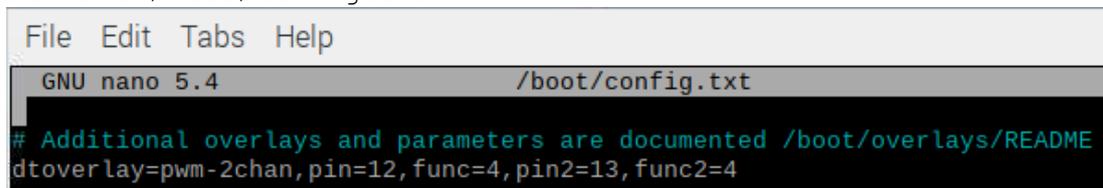
```
pip install nomdumodule --break-system-packages
```

## 4.2 - Les PWM matérielles

Pour la commande des moteurs, on utilise les pwm hardware, plus précises que les pwm software. Les informations détaillées sont sur <https://pypi.org/project/rpi-hardware-pwm/>

On ajoute l'activation des pwm (dtoverlay=pwm-2chan,pin=12,func=4,pin2=13,func2=4) à config.txt :

- `sudo nano /boot/config.txt`

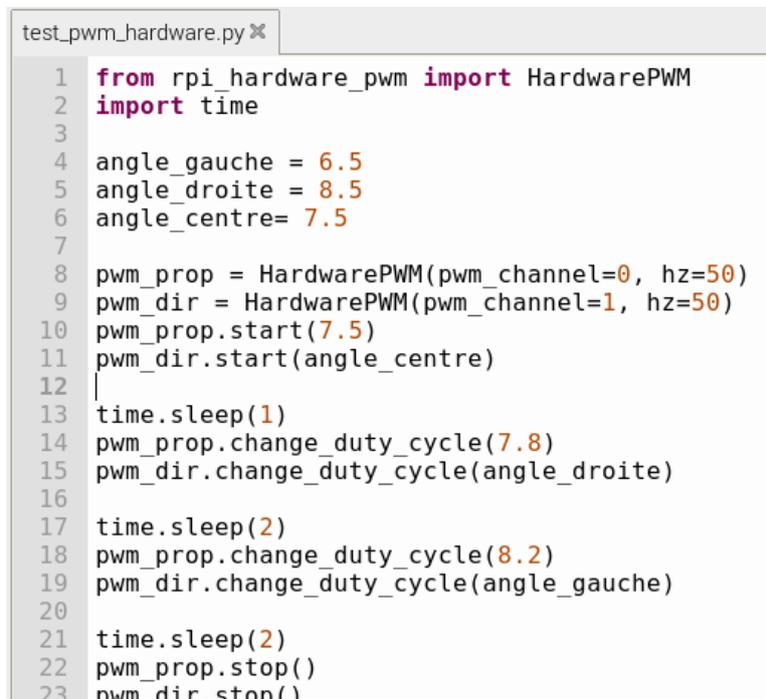


```
File Edit Tabs Help
GNU nano 5.4 /boot/config.txt
# Additional overlays and parameters are documented /boot/overlays/README
dtoverlay=pwm-2chan,pin=12,func=4,pin2=13,func2=4
```

on reboote puis on installe le module de gestion des pwm hardware, si ce n'est déjà fait :

- `pip install rpi-hardware-pwm`

Exemple de programme :



```
test_pwmHardware.py x
1 from rpiHardware_pwm import HardwarePWM
2 import time
3
4 angle_gauche = 6.5
5 angle_droite = 8.5
6 angle_centre = 7.5
7
8 pwm_prop = HardwarePWM(pwm_channel=0, hz=50)
9 pwm_dir = HardwarePWM(pwm_channel=1, hz=50)
10 pwm_prop.start(7.5)
11 pwm_dir.start(angle_centre)
12 |
13 time.sleep(1)
14 pwm_prop.change_duty_cycle(7.8)
15 pwm_dir.change_duty_cycle(angle_droite)
16
17 time.sleep(2)
18 pwm_prop.change_duty_cycle(8.2)
19 pwm_dir.change_duty_cycle(angle_gauche)
20
21 time.sleep(2)
22 pwm_prop.stop()
23 pwm_dir.stop()
```

## 5 - Le Lidar

La bibliothèque semblant la plus intéressante, notamment pour la prise en compte du mode ExpressScan est la suivante : <https://github.com/Roboticia/RPLidar/>

Le module python s'installe de la manière suivante, si ce n'est déjà fait :

- \$ pip install rplidar-roboticia

Le programme suivant permet de remettre le lidar à zéro, quel que soit la coupure du programme précédent.

**Attention** : il faut préciser le baudrate : la version A2M8 (noir et rouge) du RPLidar utilise une communication à 115200 baud et la version A2M12 (noir et violet) une communication à 256000 baud.

```
raz_lidar_roboticia.py ✕ test_lidar_roboticia.py ✕
1 from RPLidarRoboticia import rplidar
2 import time
3
4 lidar = rplidar.RPLidar("/dev/ttyUSB0",baudrate=115200)
5 lidar.disconnect()
6 time.sleep(1)
7 lidar.connect()
8 try :
9     print (lidar.get_info())
10 except :
11     print("la communication ne s'est pas établie correctement")
12 lidar.start_motor()
13 time.sleep(1)
14 lidar.stop_motor()
15 lidar.stop()
16 time.sleep(1)
17 lidar.disconnect()
```

## Références :

[1]: CoVAPSy : Premiers programmes python sur la voiture réelle, T. Boulanger, E. Délègue, K. Hoarau, A. Juton, [https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources\\_pedagogiques/covapsy-premiers-programmes-python-sur-voiture-reelle](https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/covapsy-premiers-programmes-python-sur-voiture-reelle)

[2]: Dépôt github : [https://github.com/ajuton-ens/CourseVoituresAutonomesSaclay/tree/main/Bibliotheques\\_logicielles/](https://github.com/ajuton-ens/CourseVoituresAutonomesSaclay/tree/main/Bibliotheques_logicielles/)