

Installation de Raspberry OS sur Raspberry Pi 4

CoVAPSy : Premiers programmes python sur la voiture réelle

Anthony JUTON¹

Culture Sciences de l'Ingénieur

*L*_a
3E.I

Version 1.5 du
17/10/24

école
normale
supérieure
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 3E.I 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 - Crédation 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)

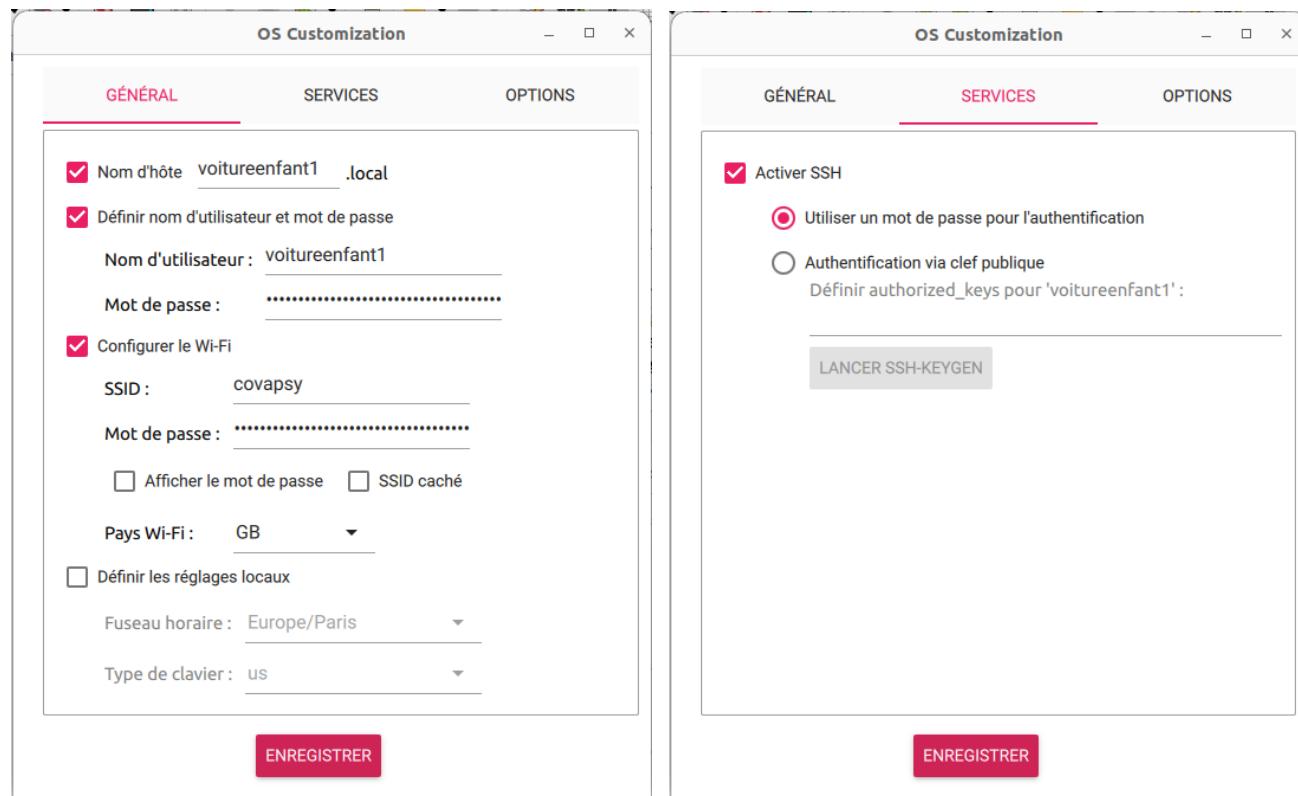


Figure 2 : Paramètres de l'image Raspberry OS

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. Cliquer sur *Enregistrer* puis *Yes* à la demande d'appliquer les modifications. 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, connecté soit sur le WIFI si il a été configuré lors de la préparation de la carte SD, soit branché par un câble Ethernet à un routeur apte à lui donner une adresse IP (la box de la maison ou un routeur du commerce). Pour s'y connecter à distance, il est alors 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.

- Il est possible de connecter un écran sur le port micro-hdmi et un clavier et une souris. Cette solution demandant du matériel supplémentaire et peu ergonomique pour la suite, ce document se concentre sur les solutions headless.
- **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, par exemple : `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

Parfois, notamment sous Windows, le DNS local ne fonctionne pas.

- **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 nano-ordinateur. 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).

Hostname	IPv4 address	MAC address	Lease time remaining	Static Lease
cobranthony2 (cobranthony2.lan)	192.168.1.194	B8:27:EB:81:03:24	11h 57m 47s	Set Static

Figure 4: Page d'accueil du routeur OpenWRT avec une carte raspeberry pi de connectée. Le bouton "Set static" permet de réserver l'adresse.

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 local 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.

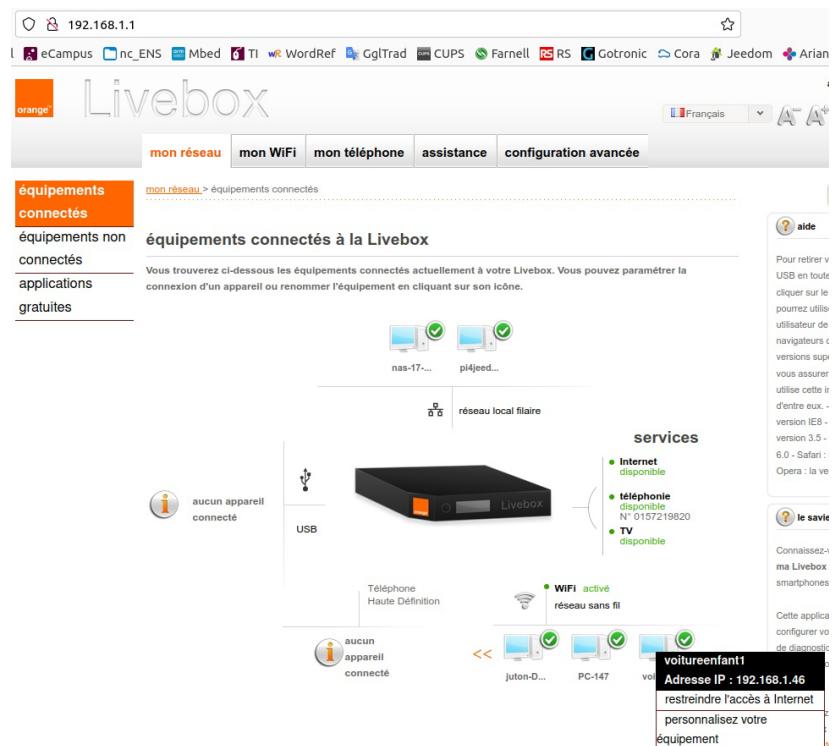


Figure 5: Présentation des équipements connectés au réseau par une LiveBox

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.

The screenshot shows the 'DHCP' tab of the 'réseau' (network) configuration section. It includes fields for the server's IP (192.168.1.1), subnet mask (255.255.255.0), start and end IP addresses (192.168.1.10 and 192.168.1.150). Below this, a table titled 'Baux DHCP statiques' (static DHCP leases) lists a single entry: 'PhonAnthony' with IP 192.168.1.10 and MAC 34:20:00:77:9A:6E. Buttons for 'ajouter' (add) and 'supprimer' (remove) are visible.

Figure 6 : 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 aisément de se connecter en **SSH** (*Secure Shell*). **SSH** est un protocole de communication sécurisé 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) et parfois sur Windows. Si cela n'est pas le cas sur Windows, on peut installer Putty par exemple.

Depuis une console (terminal Linux, invite de commande Windows ou Putty), on tape :

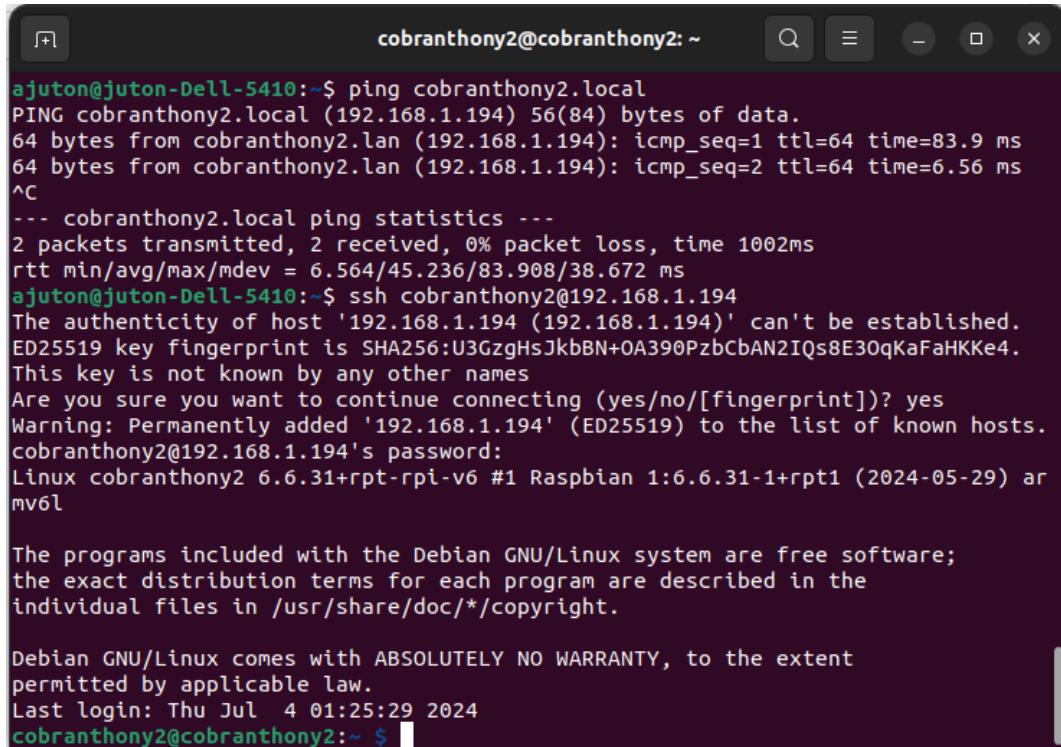
```
ssh login@<adresse IP de la raspberry pi>
```

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:~$ ping cobranthony2.local
PING cobranthony2.local (192.168.1.194) 56(84) bytes of data.
64 bytes from cobranthony2.lan (192.168.1.194): icmp_seq=1 ttl=64 time=83.9 ms
64 bytes from cobranthony2.lan (192.168.1.194): icmp_seq=2 ttl=64 time=6.56 ms
^C
--- cobranthony2.local ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 6.564/45.236/83.908/38.672 ms
ajuton@juton-Dell-5410:~$ ssh cobranthony2@192.168.1.194
The authenticity of host '192.168.1.194 (192.168.1.194)' can't be established.
ED25519 key fingerprint is SHA256:U3GzgHsJkbBN+OA390PzbCbAN2IQs8E30qKaFaHKKe4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.194' (ED25519) to the list of known hosts.
cobranthony2@192.168.1.194's password:
Linux cobranthony2 6.6.31+rpt-rpi-v6 #1 Raspbian 1:6.6.31-1+rpt1 (2024-05-29) ar
mv6l

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 Jul 4 01:25:29 2024
cobranthony2@cobranthony2:~ $
```

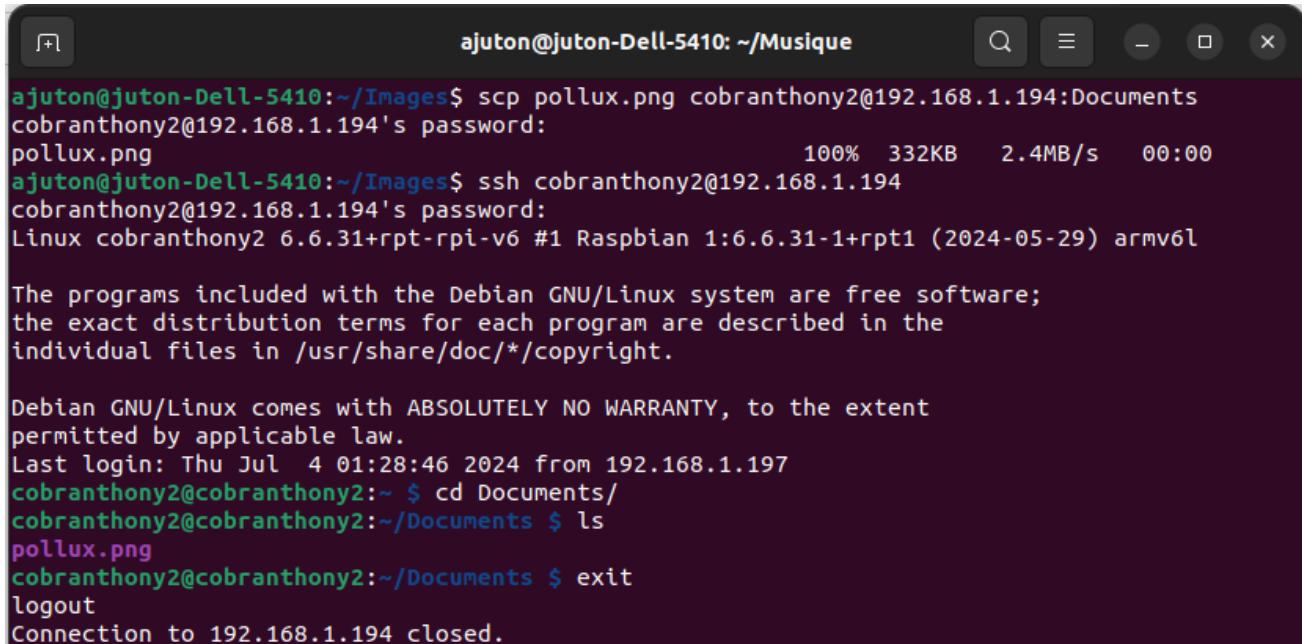
Figure 7: Connexion ssh à la Raspberry pi depuis un PC relié sur le même réseau local

Copie de fichiers à distance scp

Il est possible d'utiliser ssh pour transférer des fichiers entre l'ordinateur (PC) et le nano-ordinateur (raspberry pi). La commande s'appelle *scp* pour *ssh copy*.

Pour copier un fichier depuis le PC vers le dossier Documents de la raspberry pi. On se place dans le dossier correspondant du PC (sous linux, on clique droit dans le dossier dans l'explorateur de Fichier et on choisit *Ouvrir dans un terminal*). Sous windows, il est possible de se déplacer dans les dossiers depuis l'invite de commande avec la commande *cd* (*change directory*) :

`scp nom_du_fichier le_nom_de_l_utilisateur@le_nom_de_la_machine:Documents`



```

ajuton@juton-Dell-5410:~/Images$ scp pollux.png cobranthony2@192.168.1.194:Documents
cobranthony2@192.168.1.194's password:
pollux.png                                         100% 332KB  2.4MB/s  00:00
ajuton@juton-Dell-5410:~/Images$ ssh cobranthony2@192.168.1.194
cobranthony2@192.168.1.194's password:
Linux cobranthony2 6.6.31+rpt-rpi-v6 #1 Raspbian 1:6.6.31-1+rpt1 (2024-05-29) armv6l

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 Jul  4 01:28:46 2024 from 192.168.1.197
cobranthony2@cobranthony2:~ $ cd Documents/
cobranthony2@cobranthony2:~/Documents $ ls
pollux.png
cobranthony2@cobranthony2:~/Documents $ exit
logout
Connection to 192.168.1.194 closed.

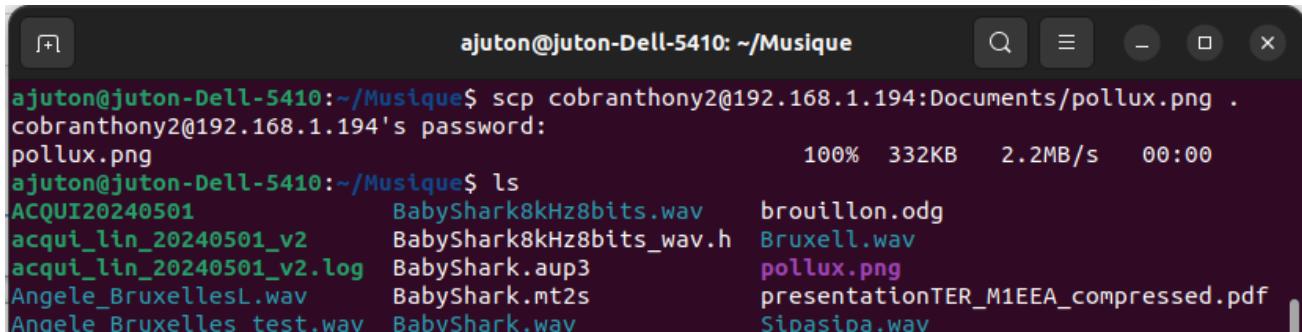
```

Figure 8: Console d'une copie de fichier du PC vers la raspberry Pi

Pour copier un fichier depuis le dossier Documents de la raspberry pi vers le PC :

`scp le nom de l utilisateur@le nom de la machine:Documents/nom_du_fichier .`

Le point à la fin permet d'indiquer que le fichier doit être déposé à l'emplacement d'où l'on envoie la commande scp.



```

ajuton@juton-Dell-5410:~/Musique$ scp cobranthony2@192.168.1.194:Documents/pollux.png .
cobranthony2@192.168.1.194's password:
pollux.png                                         100% 332KB  2.2MB/s  00:00
ajuton@juton-Dell-5410:~/Musique$ ls
ACQUI20240501           BabyShark8kHz8bits.wav    brouillon.odg
acqui_lin_20240501_v2    BabyShark8kHz8bits.wav.h  Bruxell.wav
acqui_lin_20240501_v2.log BabyShark.aup3          pollux.png
Angele_BruxellesL.wav    BabyShark.mt2s          presentationTER_M1EEA_compressed.pdf
Angele_Bruxelles_test.wav BabyShark.wav          Sipasipa.wav

```

Figure 9: Copie depuis le PC d'un fichier de la raspberry Pi vers un dossier du PC

2 - Bureau à distance

Pour faciliter l'utilisation du nano-ordinateur dans un premier temps, les outils graphiques sont plus accessibles que la ligne de commande. Celui-ci étant gourmand en ressource, il peut être intéressant de le désactiver une fois que la configuration est correcte et qu'il n'y a que du code à écrire.

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 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 *Interface Options > VNC > Yes.*

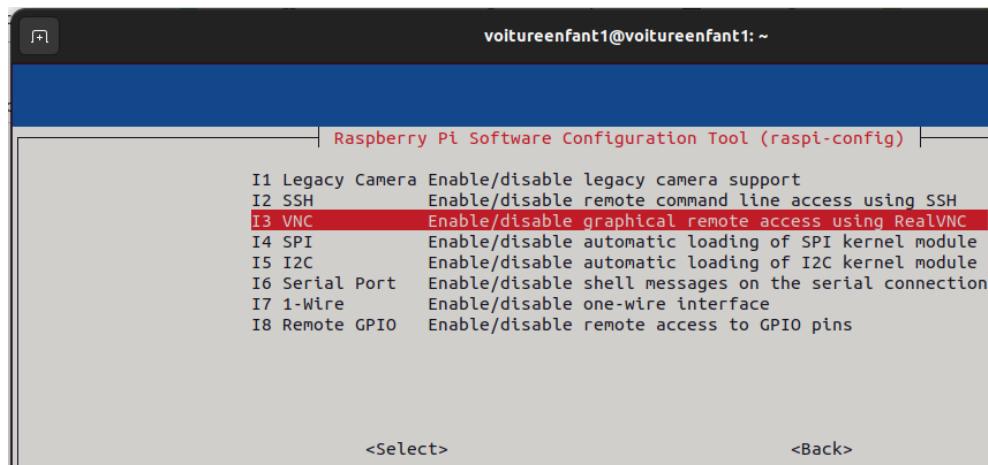


Figure 10 : Extrait de la fenêtre *raspi-config*

Enfin, aller sur *Ok* puis *Finish* pour fermer *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.

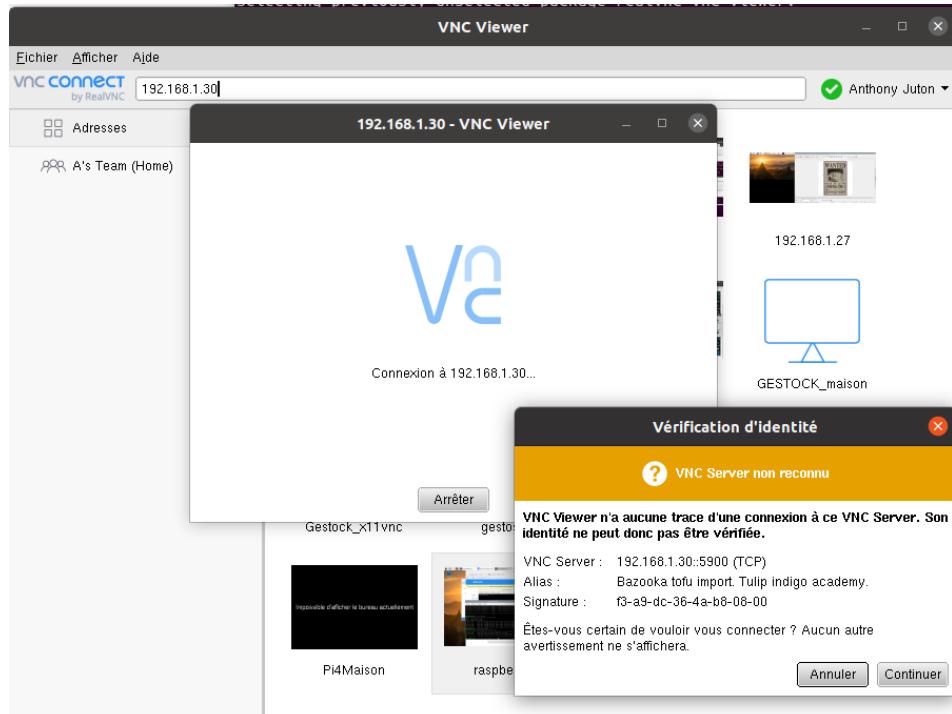


Figure 11 : Client VNC Viewer lors de la 1ère connexion à la Raspberry Pi

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.

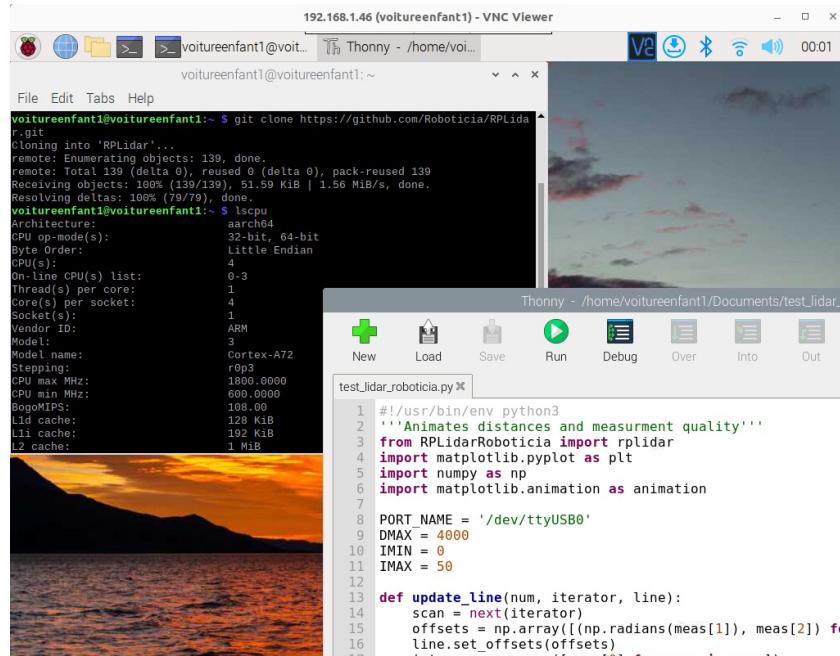


Figure 12 : Bureau à distance de la Raspberry Pi via RealVNC

Depuis le bureau à distance, il est possible d'ajouter ou retirer des réseaux wifi (par exemple celui du PC ou du smartphone de l'étudiant) : clic sur l'icône wifi puis Advanced Options puis Edit Connections...

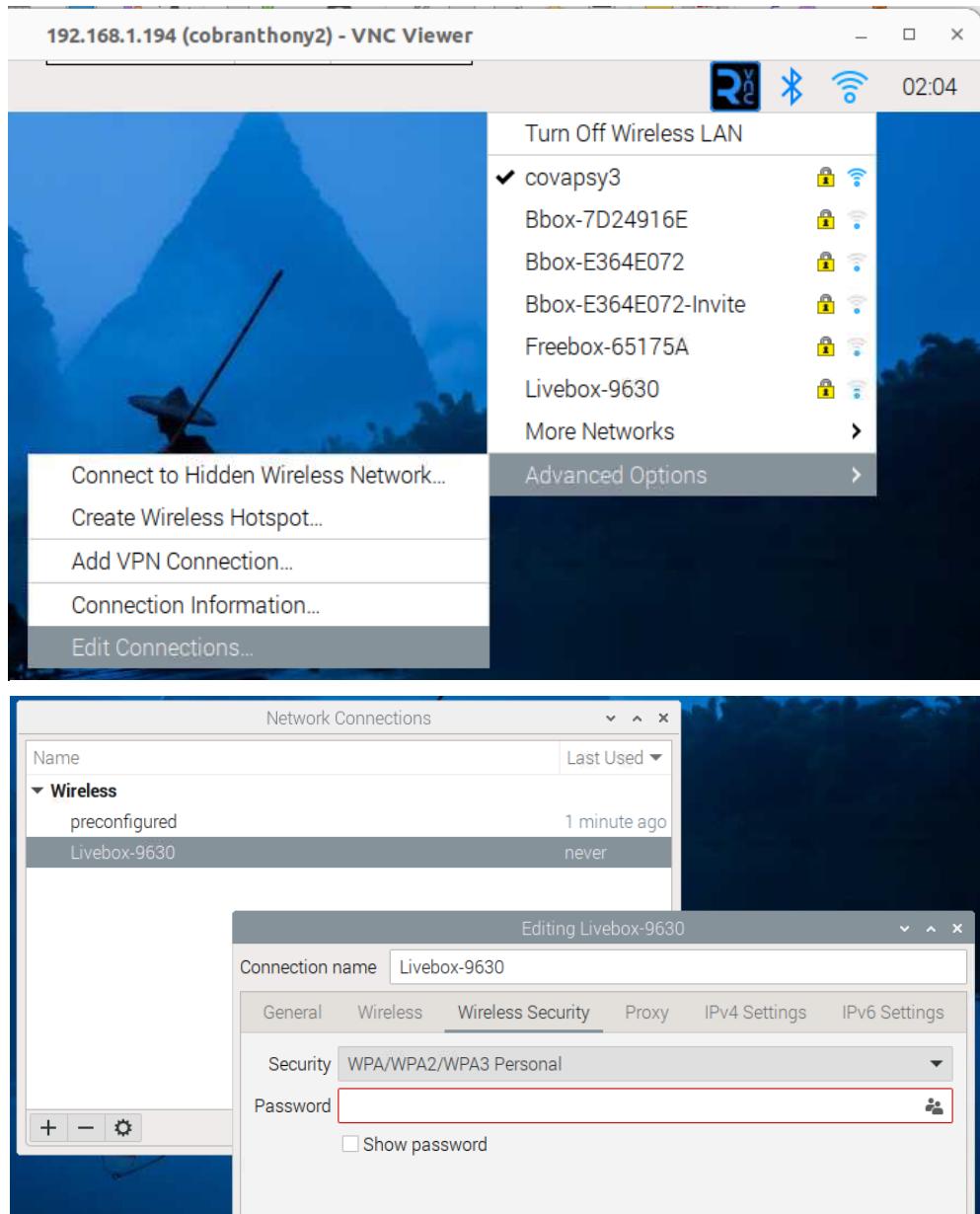


Figure 13: Configuration d'un 2nd réseau wifi depuis le bureau à distance

3 - Utiliser les entrées/sorties

Pour utiliser les périphériques de communications, il faut les activer via la fenêtre Raspberry Pi Configuration (Menu Raspberry > Preferences > Raspberry Pi Configuration). Sont utilisées ici ssh, VNC, SPI et i2c.

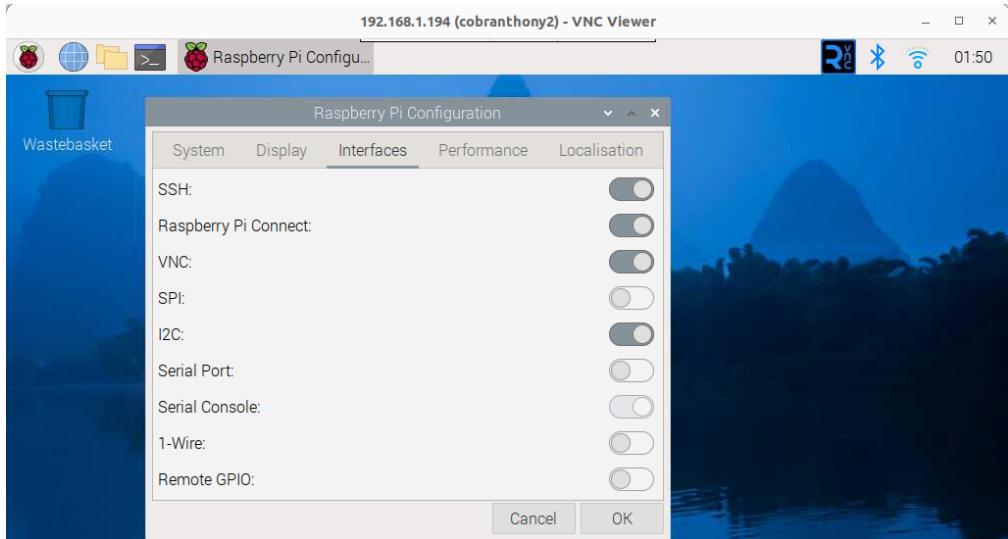


Figure 14: Activation des bus i2c et SPI via Raspberry Pi Configuration

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

3.1 - 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 pour mettre à jour le système et installer la bibliothèque matplotlib et les bibliothèques pour les pwm et le lidar :

```
sudo apt update && sudo apt upgrade -y
sudo apt install python3-matplotlib -y
pip install rpi-hardware-pwm rplidar-roboticia --break-system-packages
```

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
```

3.2 - 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/>

Ajouter l'activation des pwm à config.txt :

```
dtoverlay=pwm-2chan, pin=12, func=4, pin2=13, func2=4
```

- sudo nano /boot/firmware/config.txt
ou sur les anciennes versions : sudo nano /boot/config.txt

```
GNU nano 7.2          /boot/firmware/config.txt *
```

```
# Additional overlays and parameters are documented
# /boot/firmware/overlays/README
dtoverlay=pwm-2chan,pin=12,func=4,pin2=13,func2=4
```

Enregistrer et fermer (*CTRL+X* puis *Y* sur l'éditeur *nano*), redémarrer (*sudo reboot*) puis installer le module de gestion des pwm hardware, si ce n'est déjà fait :

- `pip install rpi-hardware-pwm`

Exemple de programme :

```
test_pwm_hardware.py ✘
```

```
1  from rpi.hardware_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()
```

4 - 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 --break-system-packages`

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_robocar.py x test_lidar_robocar.py x
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()

```

5 - Test des programmes de base CoVAPSy

Copier les programmes python du dépôt git sur la raspberry pi (en allant sur le git depuis le bureau de la voiture, soit en les chargeant sur un PC et en les copiant sur la raspberry via scp) :

https://github.com/ajuton-ens/CourseVoituresAutonomesSaclay/tree/main/Bibliotheques_logicielles/programmes_python_base_lidar_propulsion_direction_conduite

Programme `test_direction.py`

Lancer le programme `test_direction.py` et régler les butées (G/g pour déplacer la butée gauche et D/d pour déplacer la butée droite). Si le moteur guide la direction dans le sens inverse, ‘l’ permet d’inverser le sens.

Une fois la direction réglée, copier les nouveaux paramètres dans `test_direction.py` et dans `conduite_autonome_basique.py`.

Programme `test_propulsion.py`

Faire de même pour la propulsion. Attention, pour régler la marche arrière, il faut d’abord demander un « freinage » (consigne -8000 à fond en arrière) puis repasser par 0 (consigne 0) puis demander une consigne raisonnable en marche arrière (-1000).

Une fois la direction réglée, copier les nouveaux paramètres dans `test_direction.py` et dans `conduite_autonome_basique.py`.

Programme `test_lidar.py`

Régler la vitesse de transmission dans le programme (256000 pour les RPLIDAR A2-M12 115200 pour le RPLIDAR A2-M8). Lancer le programme.

Le programme `raz_lidar.py` (configuré avec la bonne vitesse de transmission) permet de remettre le lidar à 0 si il a été mal arrêté.

Une fois le lidar testé, copier la vitesse de transmission dans `conduite_autonome_basique.py`

Programme `conduite_autonome_basique.py`

Lancer le programme `conduite_autonome_basique.py`, la voiture doit se déplacer en asservissant sa position au milieu de la piste.

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
- [2]: Dépôt github : https://github.com/ajuton-ens/CourseVoituresAutonomesSaclay/tree/main/Bibliotheques_logicielles/

Ressource publiée sur Culture Sciences de l'Ingénieur : <https://eduscol.education.fr/sti/si-ens-paris-saclay>