

# Raspberry Pi

- [Configuration SD et installation d'un Raspberry Pi](#)
- [GPIO & Python sur Pi 4](#)

# Configuration SD et installation d'un Raspberry Pi

## Introduction

Le Raspberry Pi est un nano-ordinateur monocarte à processeur ARM très adapté au "digital making". D'un coût très accessible à l'origine (~40 €) et supporté par une communauté Open-Source et de Makers, de nombreux projets ont vu le jour intégrant le Raspi. Par exemple, l'écran tactile de l'AMR Sherpa est piloté par un Raspi. À la fin de 2019, le total des ventes du Raspi atteint 30 millions d'unités. Les crises du Covid et des Semi-Conducteurs provoquent cependant une augmentation spectaculaire des tarifs (x3 à x4) ou alors des délais de livraison à 6 mois pour des tarifs "classiques" ...

### Raspberry Pi 4 B 4 Go

Code commande RS: **182-2096** | Référence fabricant: **Raspberry Pi 4 4G Model B** | Marque: **Raspberry Pi**







En stock à partir du 14/11/2022, livraison sous 3 jour(s)

1  Unité

Uniquement disponible en livraison standard

Prix pour la pièce

<b>54,65 €</b>	<b>65,58 €</b>
HT	TTC

Unité	Prix par unité
1 +	54,65 €

## Avis perso :

- Les Raspberry Pi sont d'excellents outils pour l'apprentissage aux systèmes avec un Linux embarqué. Ils permettent de bénéficier de l'éco-système Linux ainsi que de la communauté associée au projet.

- Il est à noter que les tarifs ont augmenté à chaque génération passant de 30€ pour la Pi 1 à plus de 100€ pour la Pi4 avec 8Go de Ram, sans oublier l'ajout d'une alimentation USB-C, de la carte SD, d'un boîtier etc. À ce tarif, on se rapproche des PC portables ou des Intel Nuc d'occasion.
- Pour des problématiques de type IA, les cartes JETSON sont une excellente alternative aux Raspberry.

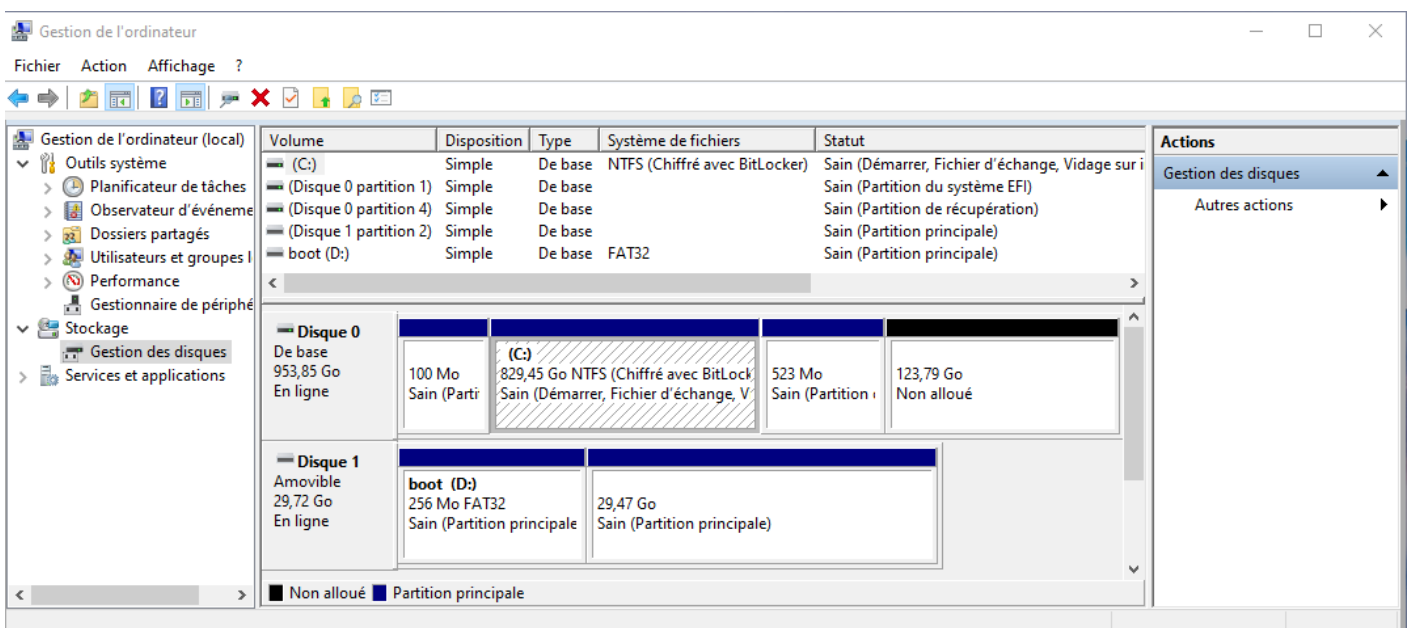
# Préparation de la carte SD

Pour créer une carte SD bootable sur le Raspberry pi, il faut dans un premier temps formater la carte. Trois cas de figure peuvent se présenter

- Vous utilisez une carte SD neuve -> vous pouvez sauter l'étape de formatage avancé et passer à [l'installation de l'OS](#).
- Vous utilisez une carte SD formatée en FAT32 ou NTFS et windows lit la taille complète de la carte -> il suffit de faire un clic droit et de formater la carte en FAT32 -> vous pouvez sauter l'étape de formatage avancé et passer à [l'installation de l'OS](#).
- Vous utilisez une ancienne carte SD, formatée pour un Raspberry Pi mais vous souhaitez réaliser la mise à jour, suivez la procédure de formatage avancée

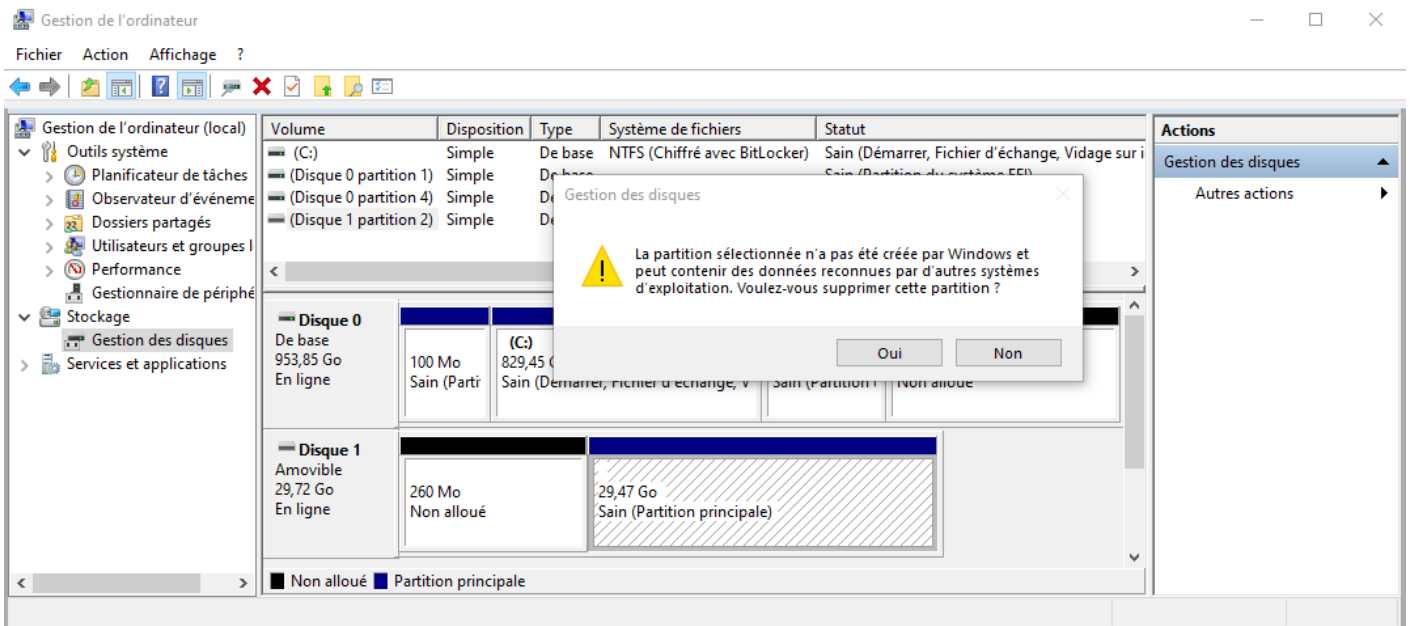
# Formatage avancé de la carte SD

Avec l'outil Gestion de l'Ordinateur, dans Stockage - Gestion des disques, voit apparaître la partition de la carte SD (256 Mo en Fat 32 lisible par Windows) et (29,47 Go en EXT4, pas visible par Windows). Il faudra dans un premier temps supprimer ces partitions, créer un nouveau volume, et le formater.



# Attention

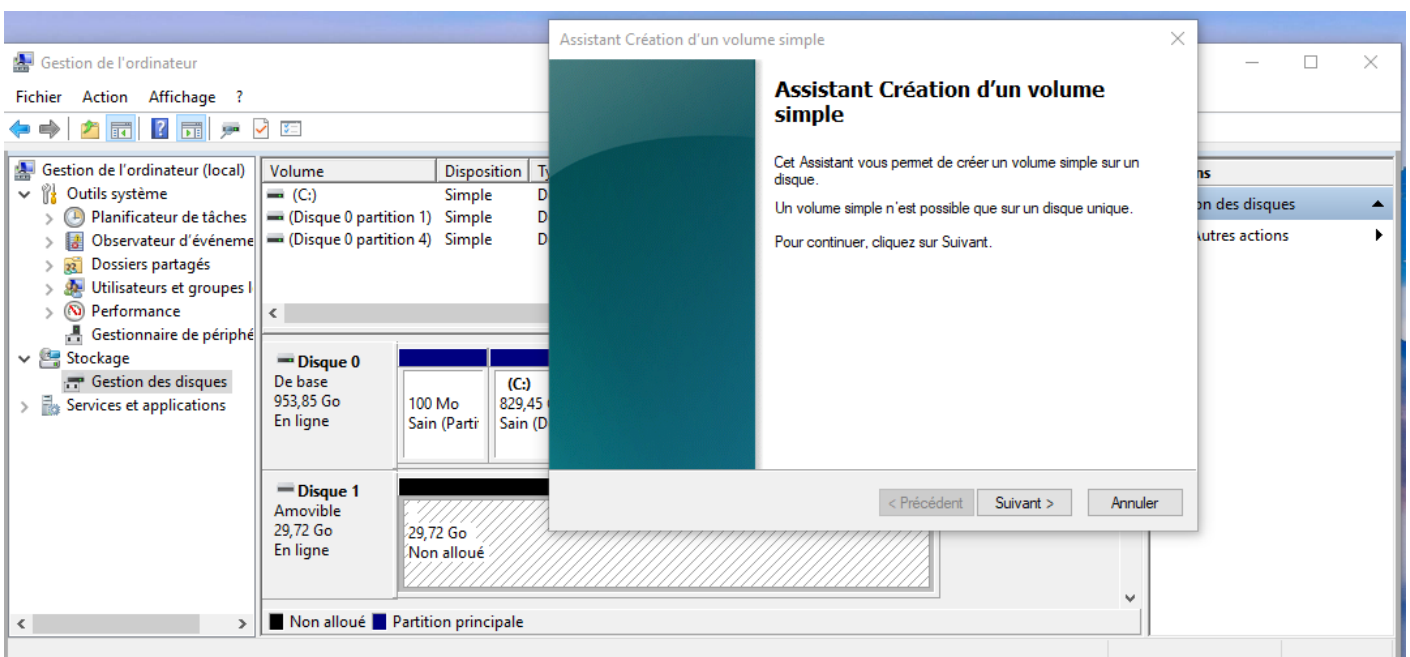
Attention, si vous supprimez un volume/partition sur votre disque dur système, vous êtes bon pour réinstaller votre PC! Vérifier 2x avant de vous lancer !



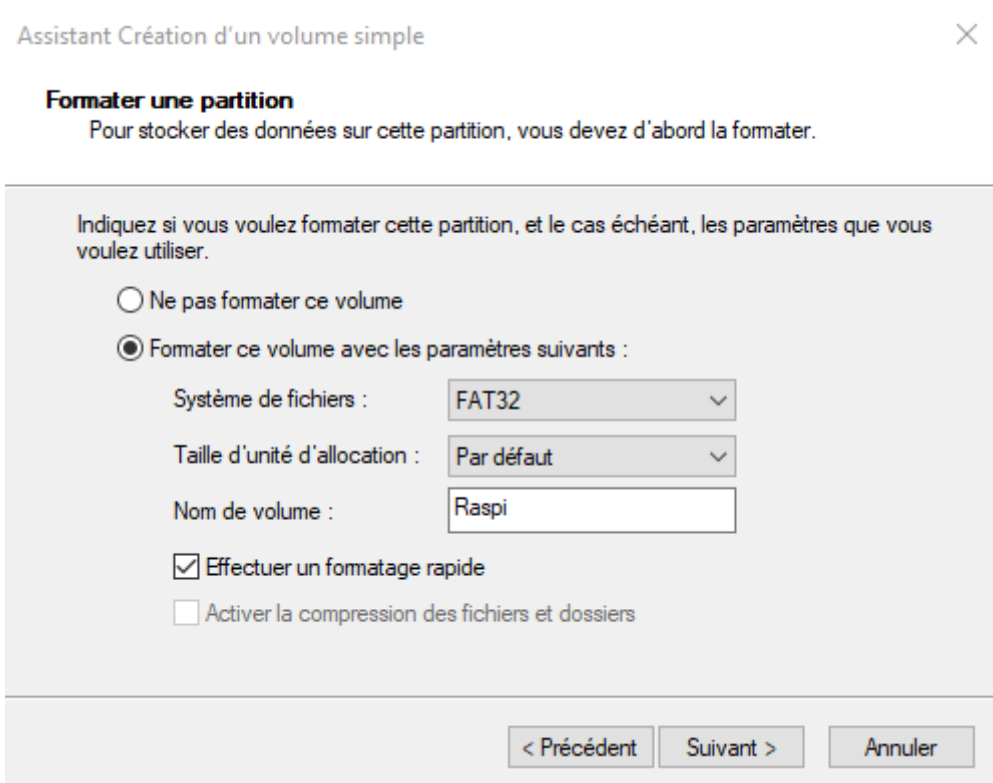
# Attention

Avant d'appuyer sur OUI, on vérifie !

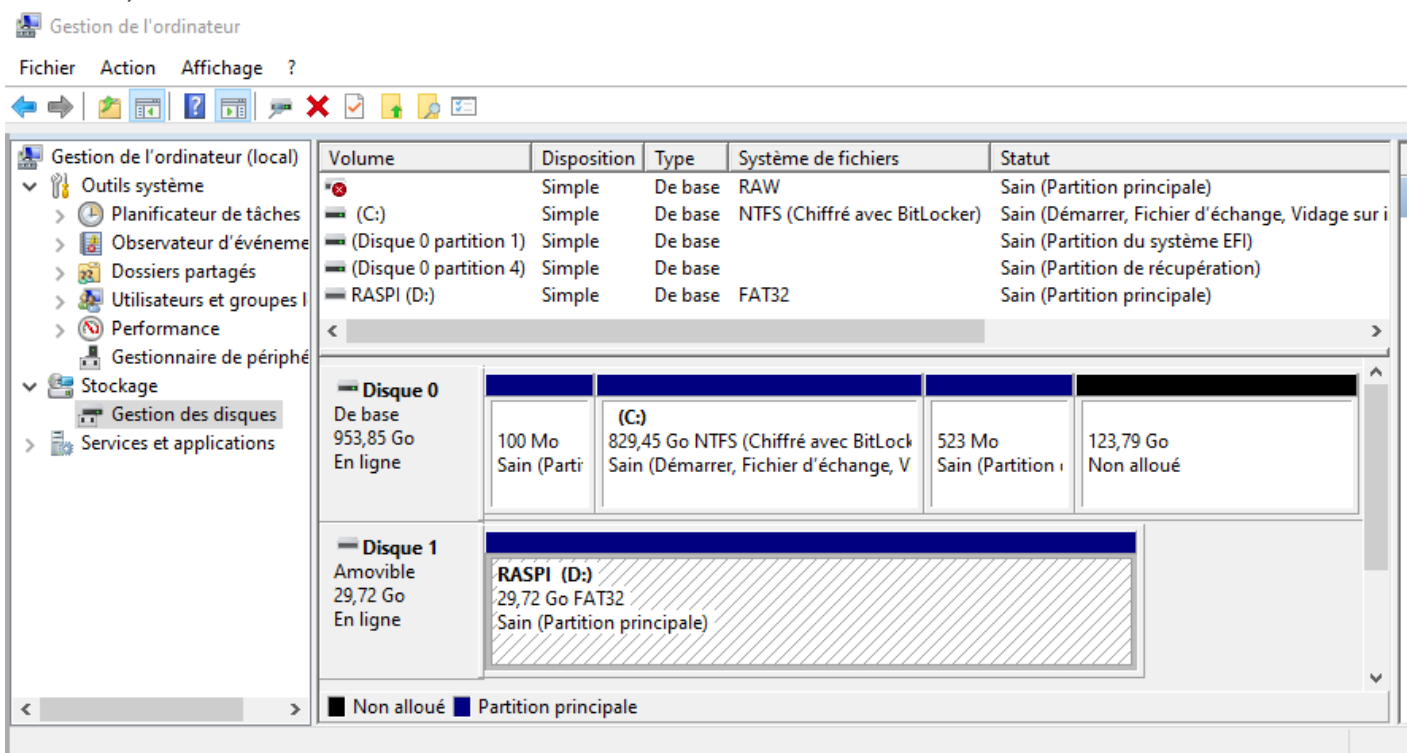
Maintenant que les anciennes partitions de la carte SD sont supprimées, on peut créer un nouveau volume avec un clic droit, créer un nouveau volume :



Pour le formatage, choisir FAT32, pour le nom de Volume, j'ai choisi Raspi.



On vérifie que le formatage et le nom de volume est correctement associé (on est OK pour l'étape suivante)



# Installation de l'OS

On télécharge sur le site [Raspberry Pi Imager](https://www.raspberrypi.com/software/) le logiciel Raspberry Pi Imager

system.

## Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. [Watch our 45-second video](#) to learn how to install an operating system using Raspberry Pi Imager.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

[Download for Windows](#)



Dans le choix de l'OS, j'ai choisi la version avec Desktop en 32 Bits. Il faut savoir que le Raspberry Pi 4 est compatible 64 Bits, mais certains logiciels comme CodeSys que nous utiliserons ne le sont pas à ce jour.



# Raspberry Pi

Système d'exploitation

Stockage

CHOISISSEZ L'OS

CHOISISSEZ LE STOCK...

ÉCRIRE

## Système d'exploitation

X



### Raspberry Pi OS (32-bit)

A port of Debian Bullseye with the Raspberry Pi Desktop (Recommended)

Sorti le : 2022-04-04

En ligne - 0.8 GO téléchargé



### Raspberry Pi OS (other)

Other Raspberry Pi OS based images



### Other general-purpose OS

Other general-purpose operating systems



### Media player OS

Media player operating systems

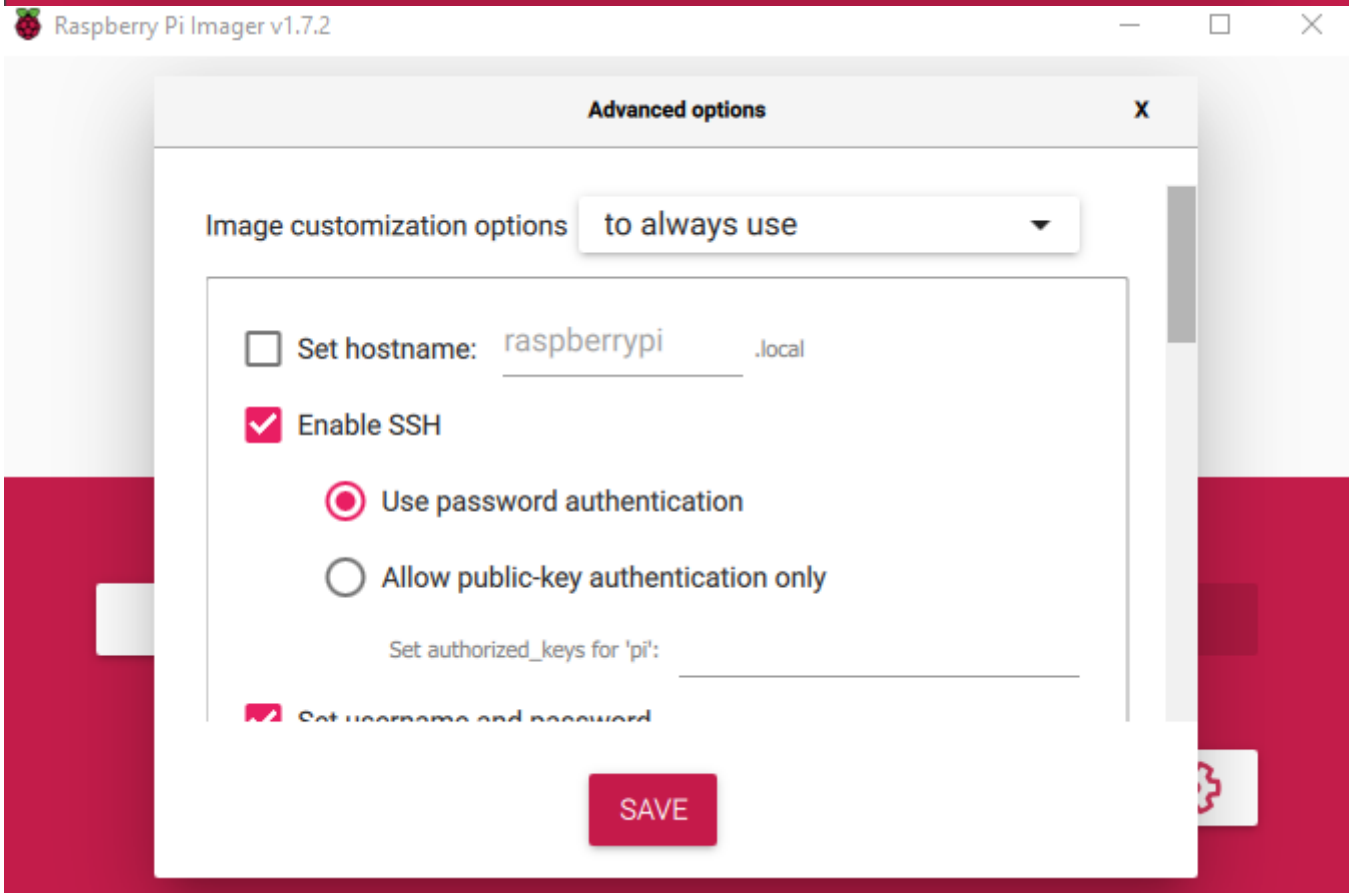
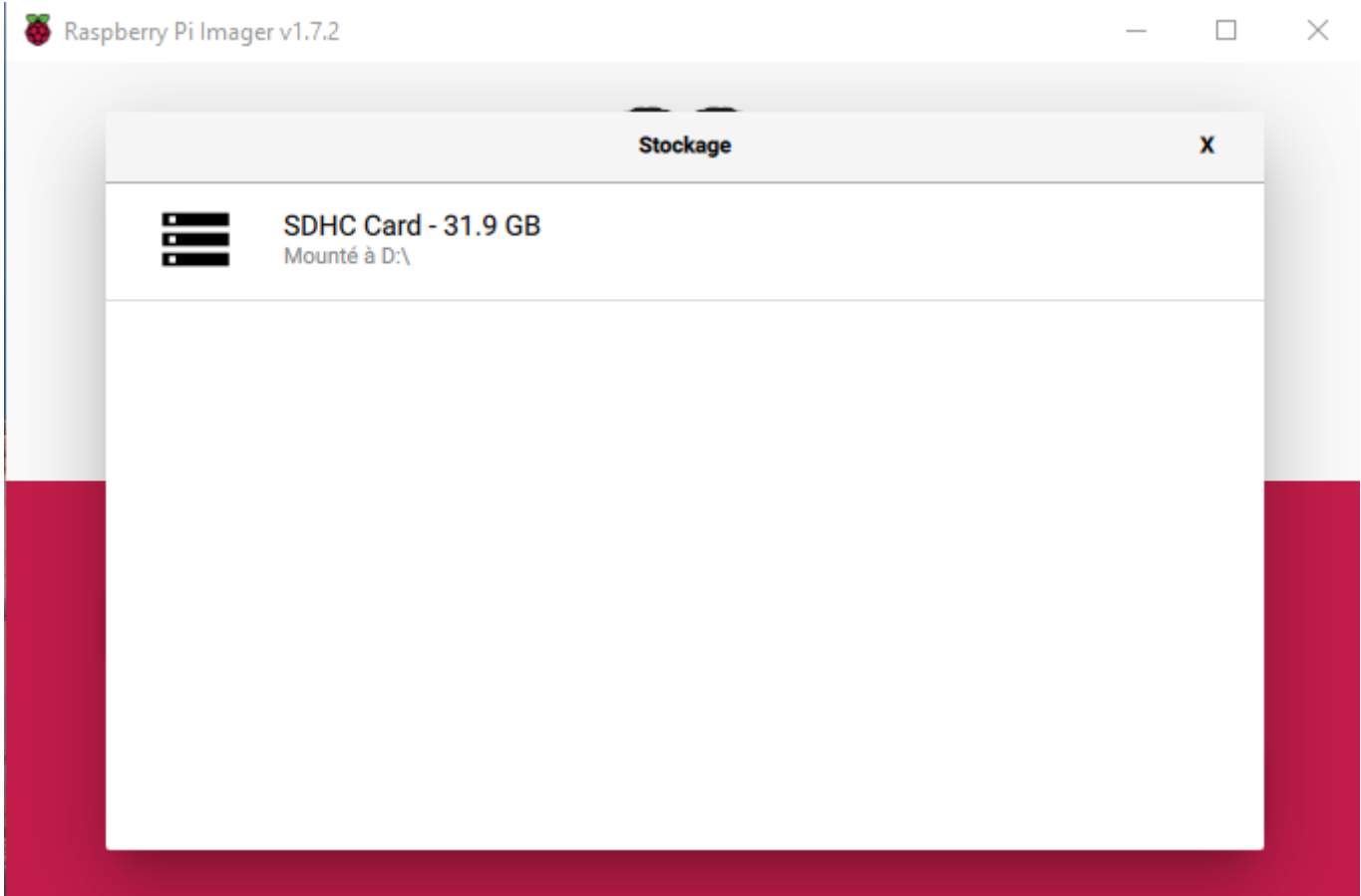


### Emulation and game OS



Pour le Choose Storage, vérifier qu'il s'agisse bien de la carte SD! Si vous formatez votre clé USB, vous perdrez les données. **Cliquer sur la petite roue crantée pour paramétrer la configuration de l'OS.**

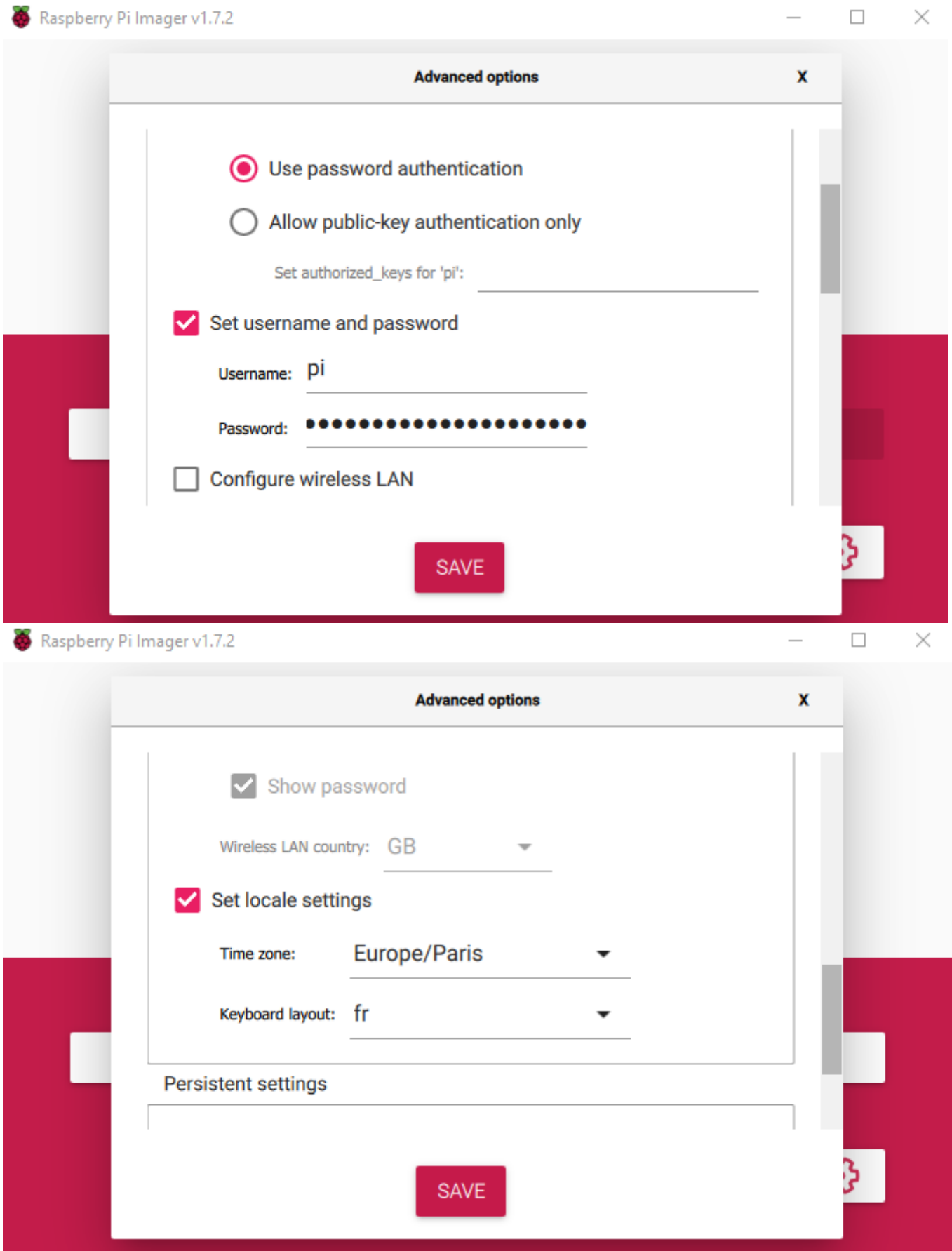
- On autorise un lien SSH en cliquant Enable SSH



- On crée un utilisateur avec son mot de passe (ici le user est pi et le mdp est 3.14)



- Dans Set Locals, on choisit le Layout FR pour le clavier



On vérifie qu'on est bien sur la carte SD et l'on fait Write



# Serveur DHCP et SSH

On met la carte SD dans le Raspberry Pi éteint, ensuite on l'alimente avec le câble USB-C. Le Raspberry Pi se connecte de base sur un serveur DHCP pour récupérer une adresse IP. L'idéal est de faire ce premier démarrage sur votre BOX. Il suffit alors de se connecter sur la box pour voir l'adresse IP qui a été affectée au port Ethernet (ici 192.168.1.15)

Il existe cependant deux autres solutions pour se passer de Box pour affecter une adresse IP au port Ethernet

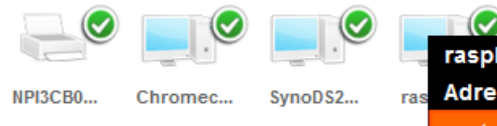
- On branche un écran, un clavier, une souris sur le Raspi et l'on configure le port Ethernet depuis l'interface graphique
- On change directement un fichier de configuration pour figer une adresse IP Statique. Cela nécessite d'avoir un PC sous Linux avec un lecteur de carte SD.

Pour avoir accès au terminal Linux du Raspi depuis un PC portable et sans avoir à brancher un écran/clavier/souris, le plus rapide est de passer par une communication SSH. Un client SSH comme Putty est installé sur les postes de l'IUT:

équipements  
connectéséquipements non  
connectés  
applications  
gratuites[mon réseau](#) > équipements connectés

## équipements connectés à la Livebox

Vous trouverez ci-dessous les équipements connectés actuellement à votre Livebox. Vous pouvez paramétrer la connexion d'un appareil ou renommer l'équipement en cliquant sur son icône.



**raspberrypi**  
**Adresse IP : 192.168.1.15**  
restreindre l'accès à Internet  
personnalisez votre  
équipement

## PuTTY Configuration



Category:

- Session
  - Logging
- Terminal
  - Keyboard
  - Bell
  - Features
- Window
  - Appearance
  - Behaviour
  - Translation
  - Selection
  - Colours
- Connection
  - Data
  - Proxy
  - SSH
  - Serial
  - Telnet
  - Rlogin
  - SUPDUP

## Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address) Port  
192.168.1.15 22

Connection type:

☒ SSH ☐ Serial ☐ Other: Telnet

Load, save or delete a stored session

Saved Sessions

Default Settings  
Raspi

Load

Save

Delete

Close window on exit:

☐ Always ☐ Never ☒ Only on clean exit

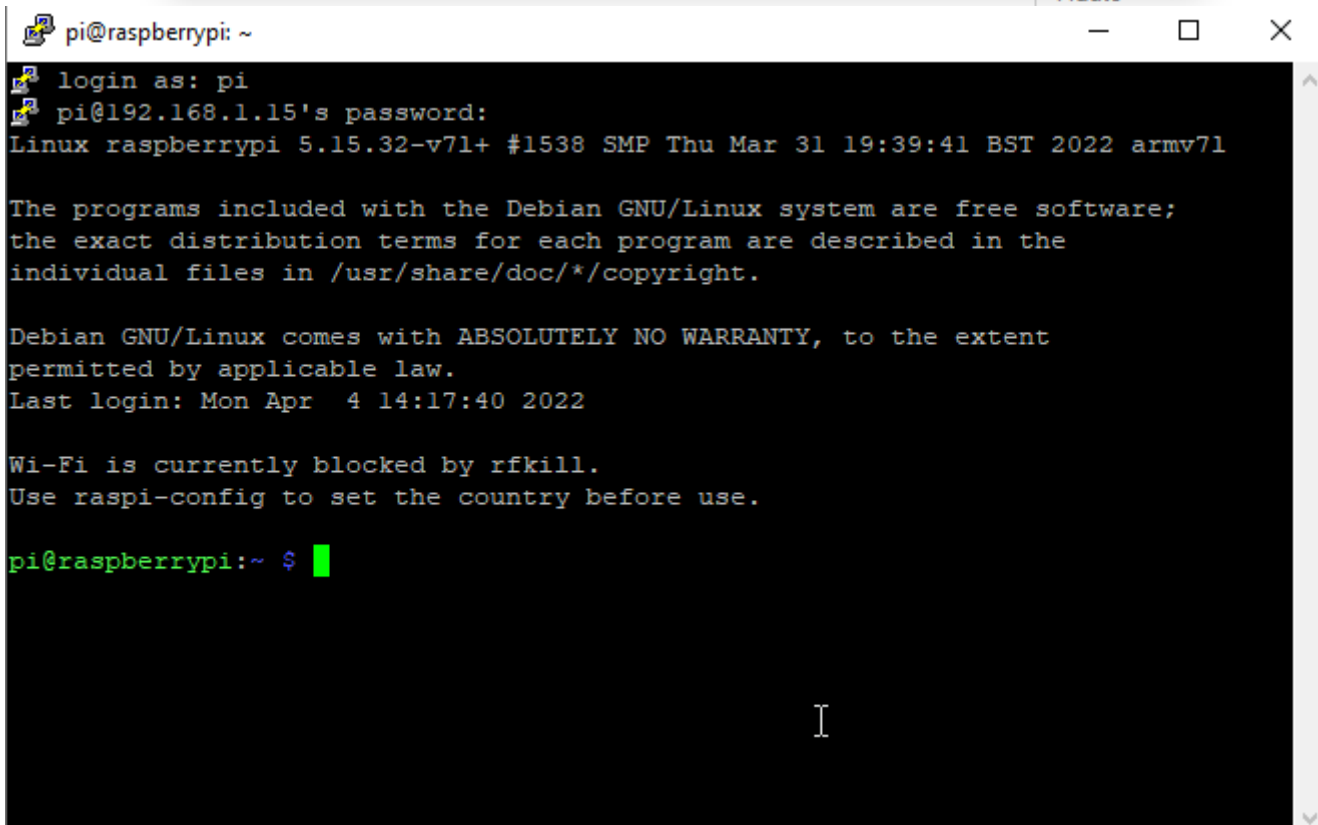
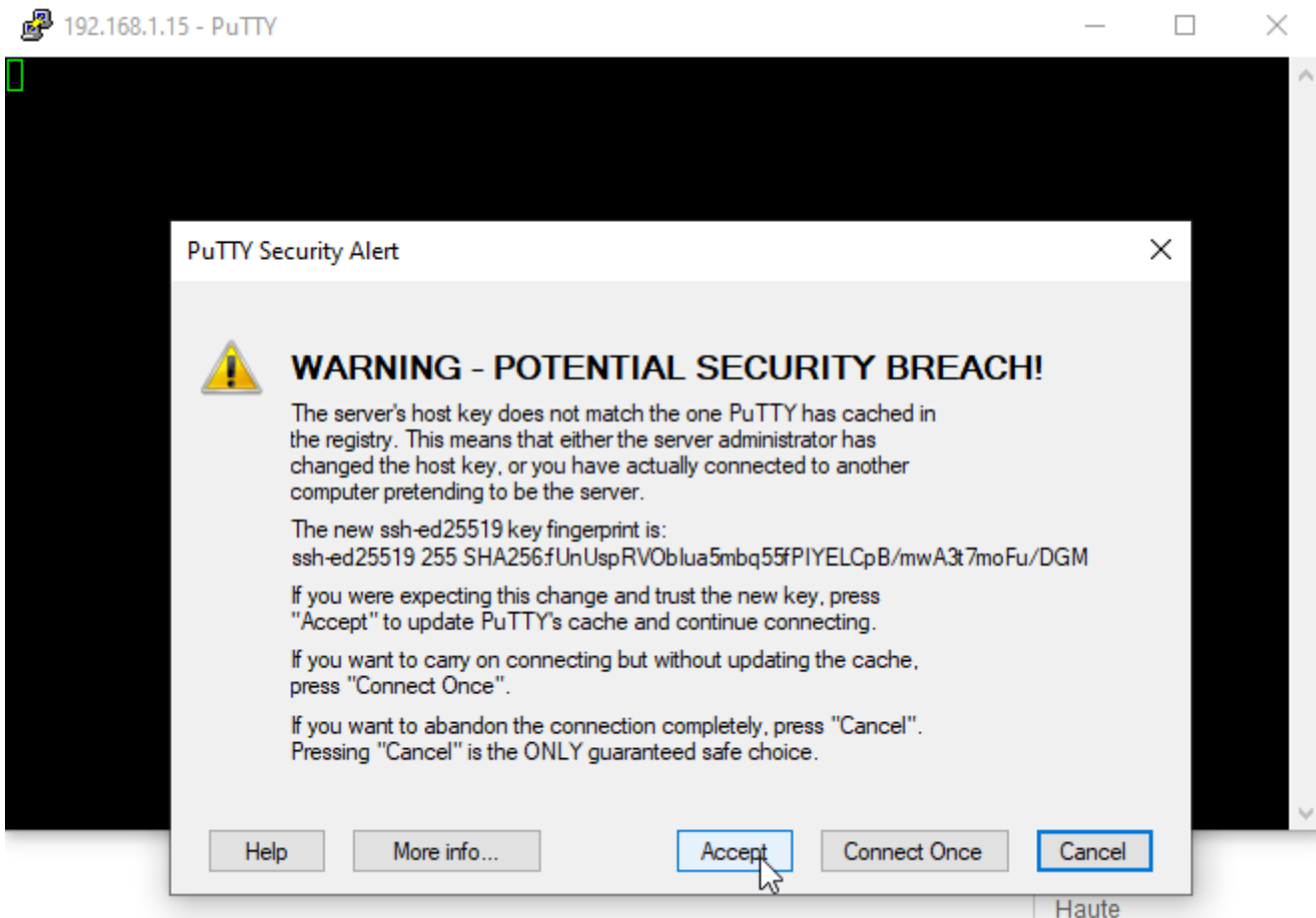
About

Help

Open

Cancel

On accepte la clé de cryptage (Accept) et l'on arrive sur le terminal Linux du Raspberry Pi depuis son PC portable. L'inconvénient, c'est qu'il ne s'agit d'une interface texte, moins conviviale qu'une interface graphique. Nous verrons dans un autre article comment se connecter sur l'interface graphique à distance avec Remote Desktop ou VNC



On peut lancer la commande de mise à jour des paquets de la distribution linux en faisant : `sudo apt update`

```
pi@raspberrypi:~ $ sudo apt update
Hit:1 http://archive.raspberrypi.org/debian bullseye InRelease
Hit:2 http://raspbian.raspberrypi.org/raspbian bullseye InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
162 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

L'upgrade de la distribution se fait en faisant : `sudo apt upgrade` (Cliquer sur Y pour valider l'upgrade)

```
pi@raspberrypi:~ $ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  dos2unix
The following packages will be upgraded:
  base-files bash chromium-browser chromium-browser-l10n chromium-codecs-ffmpeg-extra cifs-utils cups cups-client cups-common cups-core-drivers cups-daemon cups-ipp-utils cups-pk-helper cups-ppdc cups-server-common dirmngr
  distro-info-data dhcpcd-wpafile dpkg dpkg-dev ffmpeg firmware-atheros firmware-brcm80211 firmware-libertas firmware-misc-nonfree firmware-realtek gir1.2-gtk-3.0 gnome-session-bin gnome-session-common gnome-session-no-shell
  gnutls gnutls-l10n gnutls-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm gpgv gtk-update-icon-cache gzip libavcodec58 libavdevice58 libavfilter7 libavformat58 libavresample4 libavutil56
  libbasicsageenvironment1 libbc-bin libbc-dev-bin libbc-devtools libbc-l10n libbc libbc-dbg libbc-dev libbcamera-apps libbcamera-tools libbcamera0 libbcups2 libbcupsimage2 libbdk-perl libegl-mesa0 libexpat1 libexpat1-dev
  libfreecype-dev libfreecype6 libfreecype-dev libfribidi0 libgail libgail-mesa-dri libglapi-mesa libgles2-mesa libglx-mesa0 libgsm1.0 libgroupsock3 libgtk-3-0 libgtk-3-common libldap-2.4-2 liblame0 liblame-dev liblms5
  libmutter-7-0 libnm0 libntfs-9g893 libpam-chksshpwd libpam-modules libpam-modules-bin libpam-runtime libpam0g libpostproc55 libpulse-mainloop-glib0 libpulse0 libpulse0 libpulse0 libraspberrypi-bin libraspberrypi-dev
  libraspberrypi-doc libraspberrypi0 libstdc++2.0-0 libstdc++11 libswresample3 libswscale5 libusageenvironment3 libvlc-bin libvlc libvlccore9 libvobclient0 libxml2 locales logrotate lxdm lxdm-data
  lxplug-pbatt lxplug-updater mesa-va-drivers mesa-vaapi-drivers mutter mutter-common nano ntfs-3g openssh-client openssh-server openssh-sftp-server openssh-pcmanfm pi-bluetooth piwiz pulseaudio pulseaudio-module-bluetooth
  pulseaudio-utils python-pip-whl python3-pip python3-pip python3-plugin-base vlc-plugin-access-extra vlc-plugin-base vlc-plugin-notify vlc-plugin-qt vlc-plugin-samba vlc-plugin-skins2 vlc-plugin-video-output vlc-plugin-video-splitter vlc-plugin-visualization wireless-regdb
  xcompmgr xz-utils
162 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 292 MB of archives.
After this operation, 11.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Votre Raspberry Pi est maintenant à jour, nous pouvons y déployer le Remote Desktop et le VNC pour simplifier sa programmation.

# Bureau à distance / Remote Desktop

Un des outils de base installé sur Windows est le bureau à distance (Remote Desktop).

## Installation d'XRDP sur le Raspi

Pour que le bureau à distance du Raspberry Pi soit disponible depuis Windows, il est nécessaire d'installer le paquet xrdp. Depuis un terminal sur le Raspberry Pi ou depuis une liaison SSH faire `sudo apt install xrdp` (Y pour valider l'installation)

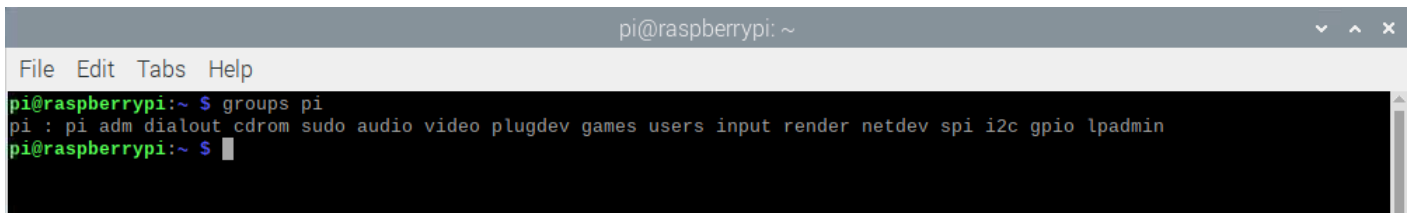
```
pi@raspberrypi:~ $ sudo apt install xrdp
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libglul-mesa libxxf86dgal x11-apps x11-session-utils x11-utils xbitmaps
  xfonts-100dpi xfonts-75dpi xfonts-base xfonts-encodings xfonts-scalable
  xfonts-utils xorg xorg-docs-core xorgxrdp
Suggested packages:
  mesa-utils xorg-docs x11-xfs-utils guacamole
The following NEW packages will be installed:
  libglul-mesa libxxf86dgal x11-apps x11-session-utils x11-utils xbitmaps
  xfonts-100dpi xfonts-75dpi xfonts-base xfonts-encodings xfonts-scalable
  xfonts-utils xorg xorg-docs-core xorgxrdp xrdp
0 upgraded, 16 newly installed, 0 to remove and 0 not upgraded.
Need to get 15.6 MB of archives.
After this operation, 23.4 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

et depuis la version BullsEye d'avril 2022 du Raspberry Pi, cela ne fonctionne plus correctement avec le user pi qui possède les droits admin sur la carte.

## Bug xrdp

C'est un bug sur la distribution BullsEye et le contournement trouvé et de retirer certains groupes au user pi.

Classiquement, le user pi appartient aux groupes suivants :

A screenshot of a terminal window titled 'pi@raspberrypi: ~'. The window has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. The terminal shows the command 'pi@raspberrypi:~ \$ groups pi' and the output 'pi : pi adm dialout cdrom sudo audio video plugdev games users input render netdev spi i2c gpio lpadmin'. The prompt 'pi@raspberrypi:~ \$' is followed by a cursor.

```
pi@raspberrypi:~ $ groups pi
pi : pi adm dialout cdrom sudo audio video plugdev games users input render netdev spi i2c gpio lpadmin
pi@raspberrypi:~ $
```

Il faut retirer les groupes render et video du user pi pour que le bureau à distance fonctionne. Cela a pour conséquence d'introduire des effets de bords sur certains logiciels qui ne vont peut être plus fonctionner. Pour développer des scripts python, ou faire du CodeSys, cela n'a pas d'influence. Pour retirer les deux groupes :

```
sudo gpasswd -d video pi
sudo gpasswd -d render pi
```

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo gpasswd -d render pi  
Removing user render from group pi  
gpasswd: user 'render' is not a member of 'pi'  
pi@raspberrypi:~ $  
  
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo gpasswd -d video pi  
Removing user video from group pi  
gpasswd: user 'video' is not a member of 'pi'  
pi@raspberrypi:~ $
```

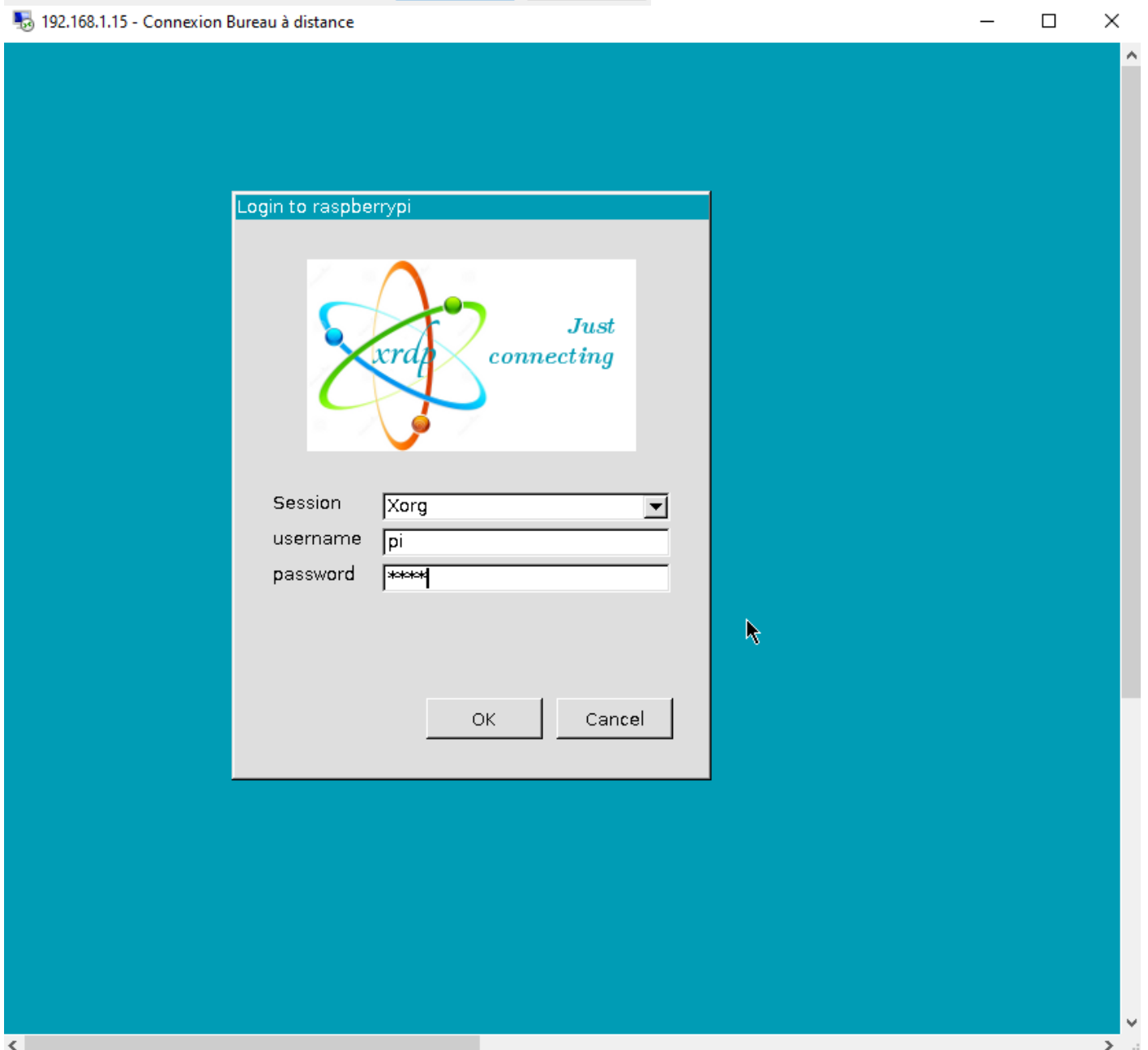
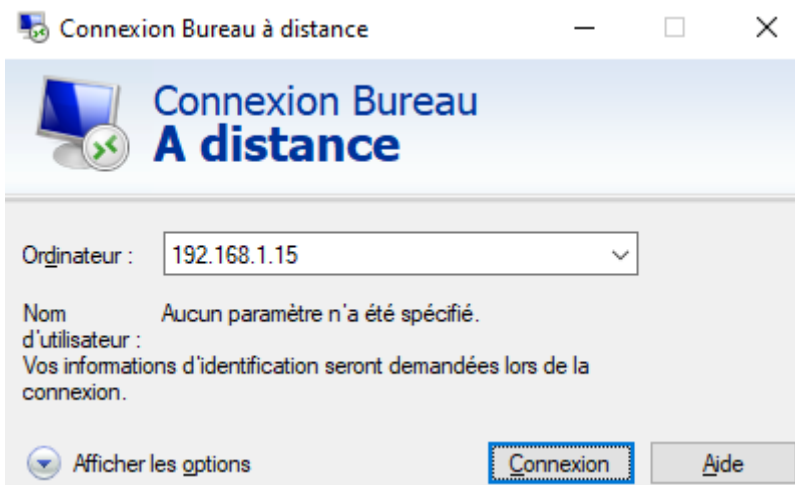
- on fait un reboot pour activer les modifications: `sudo reboot`
- (pour faire un shutdown: `sudo shutdown -h now`)

## Remote Desktop sur Windows

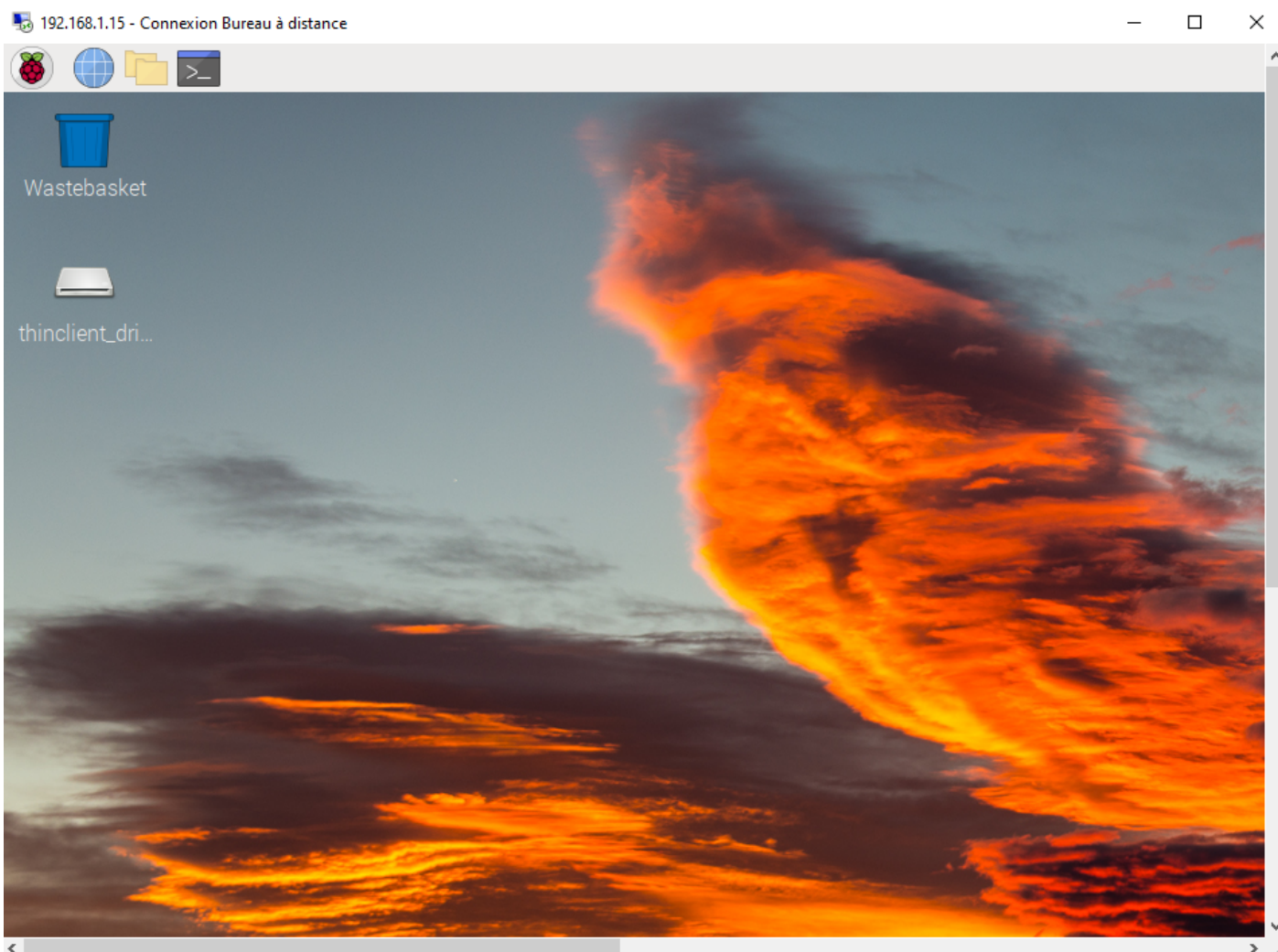
Nous pouvons maintenant utiliser l'outil de connexion de bureau à distance :

- L'adresse IP correspond à l'adresse statique ou DHCP du Raspberry Pi





Et l'on accède à distance au bureau du Raspberry Pi:



Contrairement à la liaison SSH, on peut profiter de l'ergonomie offerte par l'interface graphique du Raspberry Pi. Les puristes se contenteront de la liaison SSH :wink:

- L'avantage de cette méthode est le gain de temps, pas d'écran à installer, ni clavier/souris, ce qui constitue également un gain d'espace.
- On restera attentif au fait que nous avons retiré les groupes render et video du user pi.
- La connexion bureau à distance est un outil standard Windows installé de base avec Windows 10. La communauté Open Source préfère l'outil VNC pour faire des liaisons d'un poste de développement Linux vers le Raspberry Pi (sous Linux également). Sous Linux, les clients VNC sont libres, mais sous Windows, le client VNC le plus utilisé est gratuit mais pas libre (VNC Viewer de RealVNC)

# VNC sur Raspberry Pi

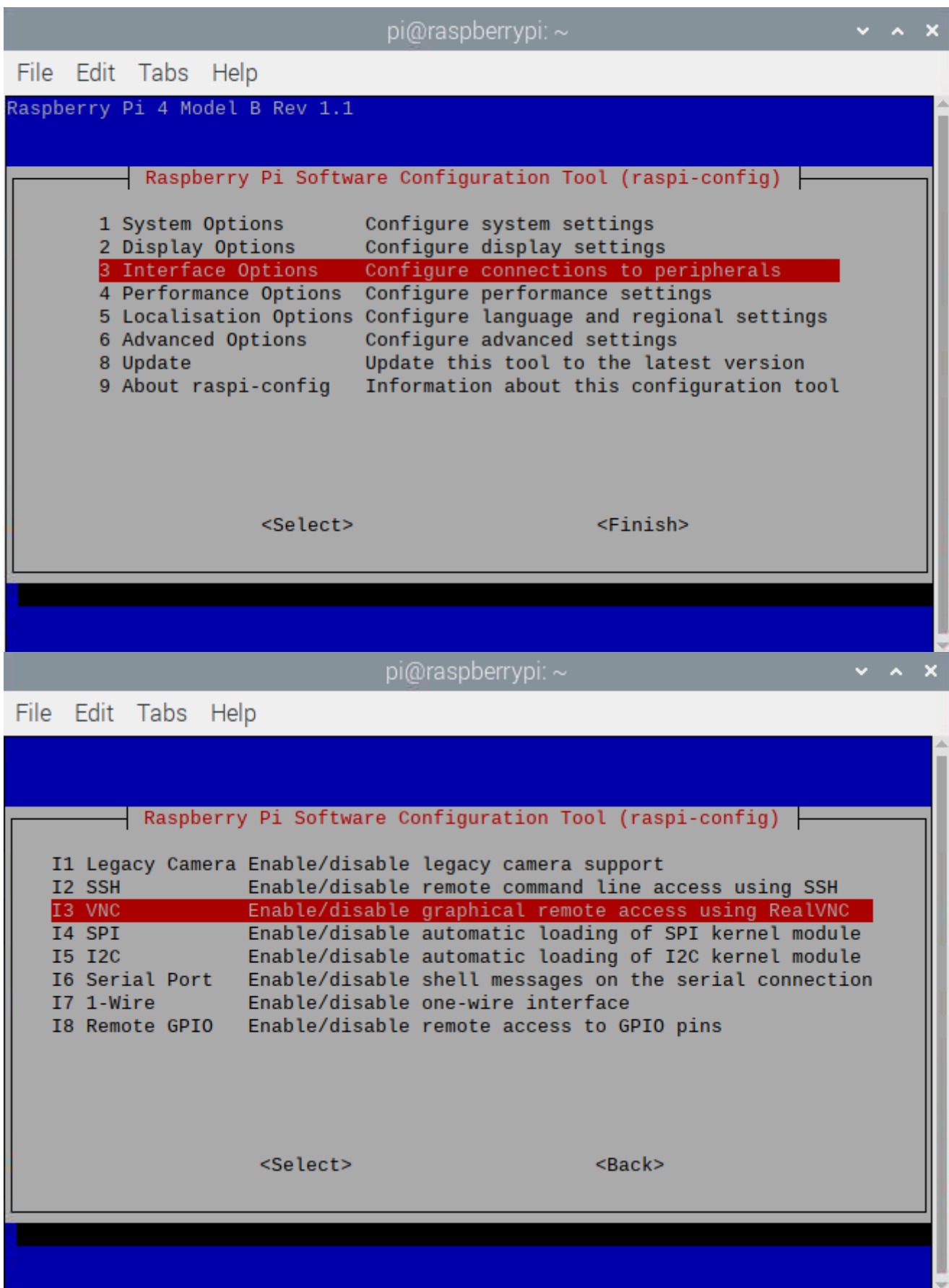
## Activation du serveur VNC sur le Raspi

Depuis un terminal ou une liaison SSH:

- lancer la commande : `sudo raspi-config`

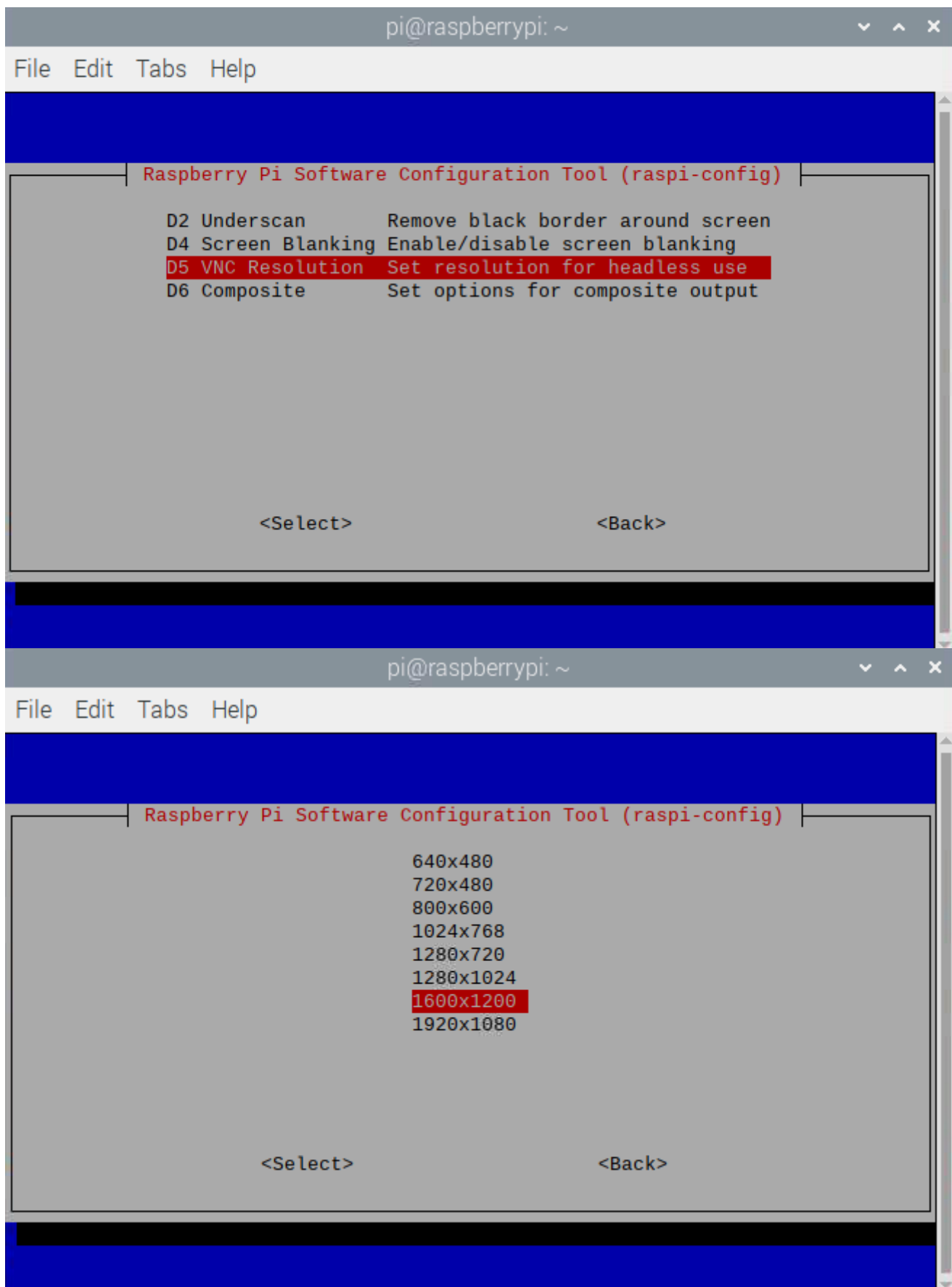
Il faut dans un premier activer le serveur VNC sur le Raspberry Pi

- Interface Options -> Enable graphical remote access using RealVNC



On peut régler la résolution du serveur VNC dans Display Option -> Set resolution for headless use

- A choisir en fonction de la résolution de l'écran sur lequel on développe (la résolution du VNC doit être plus petite que celle de l'écran)

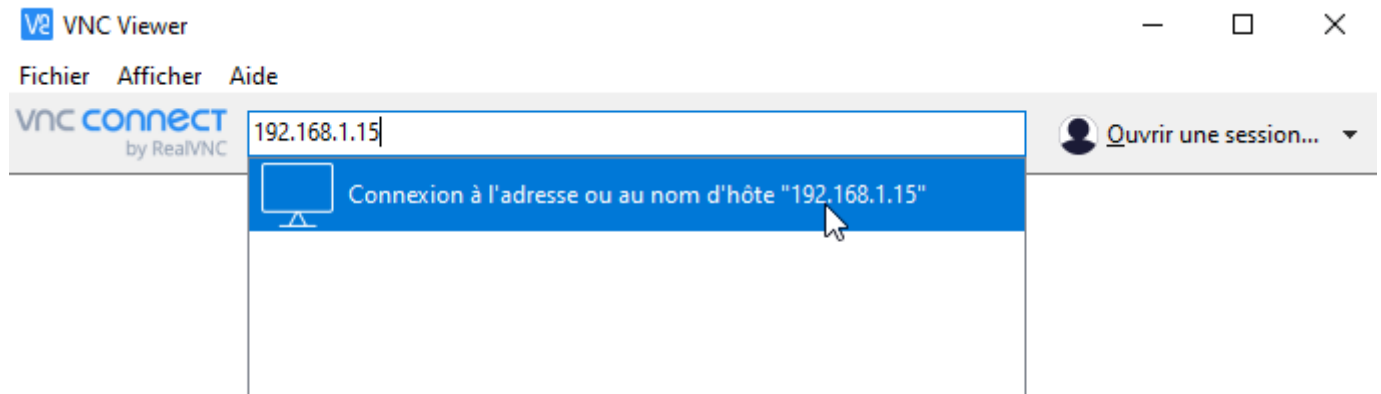


Lorsque l'on valide les modifications, un reboot du Raspi se fait automatiquement.

# Client VNC Windows

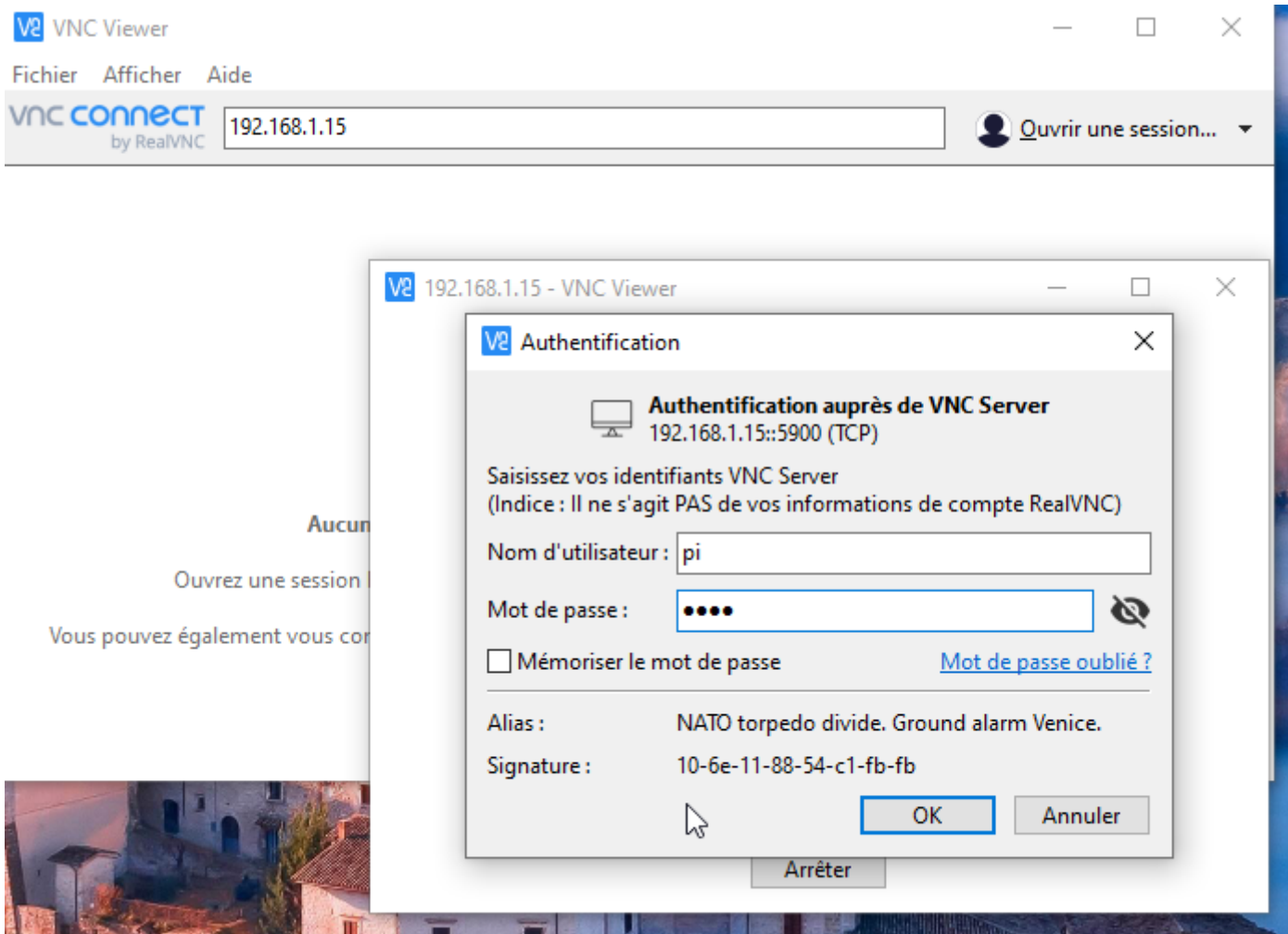
Le client VNC le plus facile à mettre en oeuvre est VNC Viewer [Client RealVNC VNC Viewer](#)

On place l'adresse IP du Raspberry PI et l'on appuie sur Entrée. (Ne pas cliquer sur Ouvrir une session)



On renseigne :

- le user : pi
- le mdp : 3.14 (si c'est votre mdp)

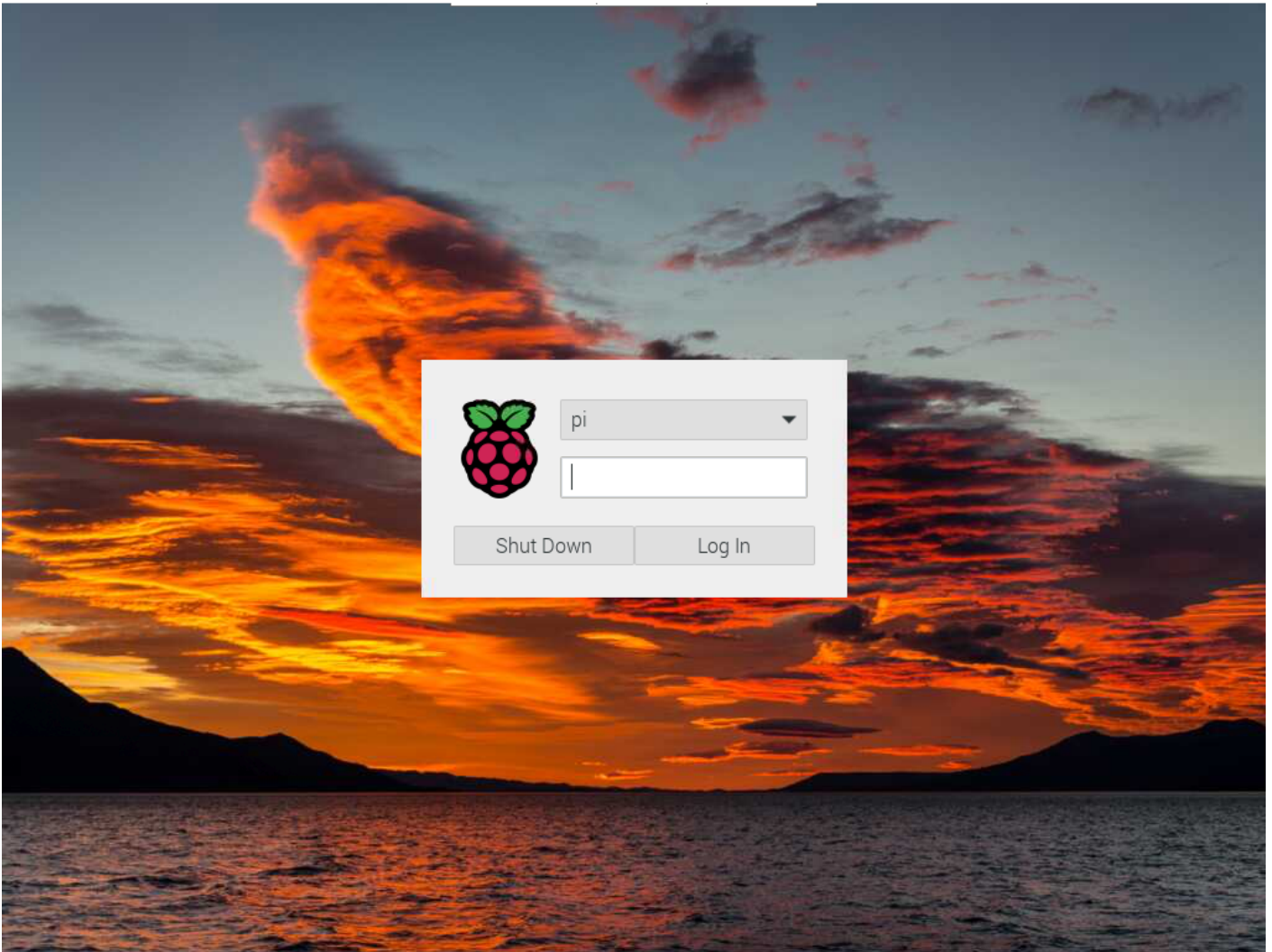




On arrive sur le bureau, il faut à nouveau se logger avec le bon mdp (3.14)

192.168.1.15 (raspberrypi) - VNC Viewer

— □ ×



# Conclusion

Dans le développement en embarqué, il y a l'école :

- SSH : tout en ligne de commande (les puristes)
- Bureau distant : on profite de l'interface graphique. Ce n'est pas toujours nécessaire, mais c'est rassurant quand on débute (les pragmatiques)

Pour les bureaux distants :

- XRDP possède l'avantage d'avoir le client bureau distant installé sur chaque PC Windows, ce qui est pratique quand on ne peut pas installer des logiciels sur le poste école. Dommage pour ce bug sur la version BullsEye du Raspberry Pi qui nécessite de modifier les groupes du user pi. Les forums de support préconisent VNC.
- VNC fait partie de l'écosystème Linux, mais paradoxalement, le client Windows n'est pas libre bien que gratuit.

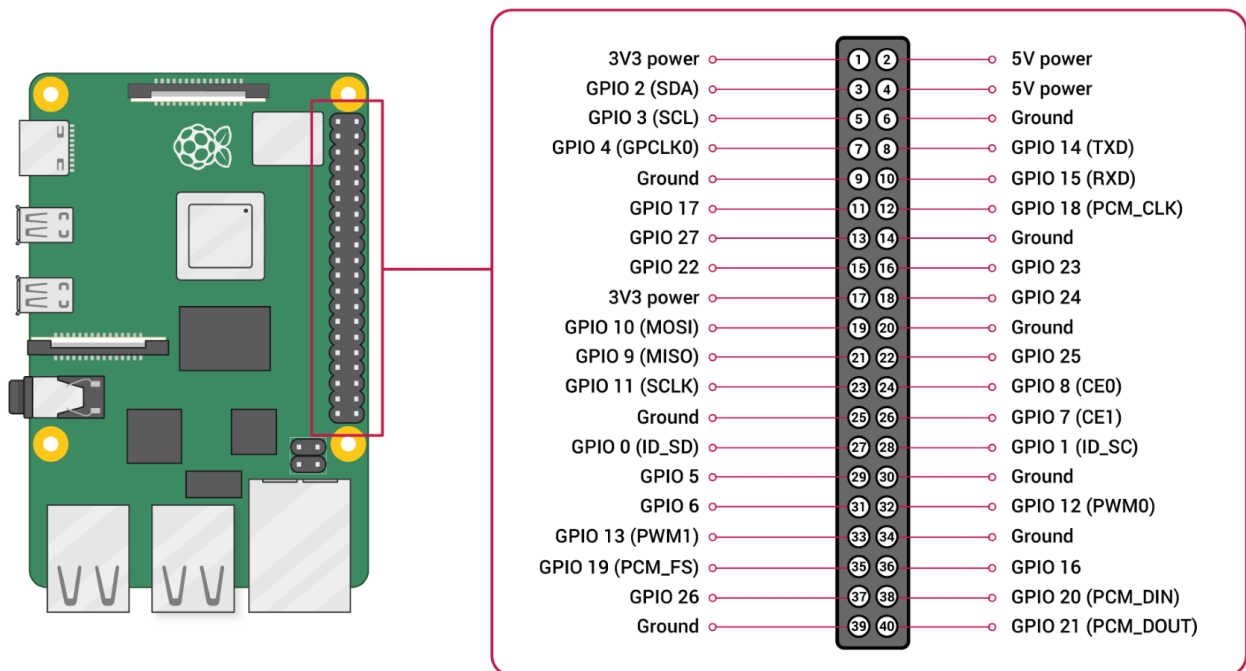




# GPIO & Python sur Pi 4

## GPIO & Python sur Pi 4

GPIO signifie General Purpose Input Output. Les broches GPIO du Raspberry Pi peuvent se configurer soit en entrée, soit en sortie. Certaines broches sont associées à des fonctions de type PWM, I<sup>2</sup>C, SPI, Serial, 1 wire, etc. Comparé à un microcontrôleur MicroChip, l'utilisateur ne rentrera pas dans les détails de configuration de registre bas-niveau (TRIS, LAT, PORT...) le driver s'occupe de cela. Le programmeur pourra se concentrer sur l'applicatif avec des fonctions de haut niveau pour piloter les GPIO.



### Attention

Les GPIO du Raspberry Pi sont compatibles avec des niveaux de tension 0-3.3V. Appliquer du 5V sur ces GPIO va les détruire. N'oubliez pas que le prix des Raspberry ont augmenté de manière importante depuis 2021 et que les délais de livraison sont de 6 mois.

Ci-dessous, une image plus détaillée des fonctions des broches du Raspberry Pi4

PIN	NAME			NAME	PIN
01	3.3V DC Power			5V DC Power	02
03	GPIO02 (SDA1, I <sup>2</sup> C)			5V DC Power	04
05	GPIO03 (SDL1, I <sup>2</sup> C)			Ground	06
07	GPIO04 (GPCLK0)			GPIO14 (TXD0, UART)	08
09	Ground			GPIO15 (RXD0, UART)	10
11	GPIO17			GPIO18(PWM0)	12
13	GPIO27			Ground	14
15	GPIO22			GPIO23	16
17	3.3V DC Power			GPIO24	18
19	GPIO10 (SP10_MOSI)			Ground	20
21	GPIO09 (SP10_MISO)			GPIO25	22
23	GPIO11 (SP10_CLK)			GPIO08 (SPI0_CE0_N)	24
25	Ground			GPIO07 (SPI0_CE1_N)	26
27	GPIO00 (SDA0, I <sup>2</sup> C)			GPIO07 (SCL0, I <sup>2</sup> C)	28
29	GPIO05			Ground	30
31	GPIO06			GPIO12 (PWM0)	32
33	GPIO13 (PWM1)			Ground	34
35	GPIO19			GPIO16	36
37	GPIO26			GPIO20	38
39	Ground			GPIO21	40

Dans un premier temps, nous allons configurer une des GPIO en Output pour faire clignoter une Led. Pour brancher la Led, il nous faut le signal de la GPIO:

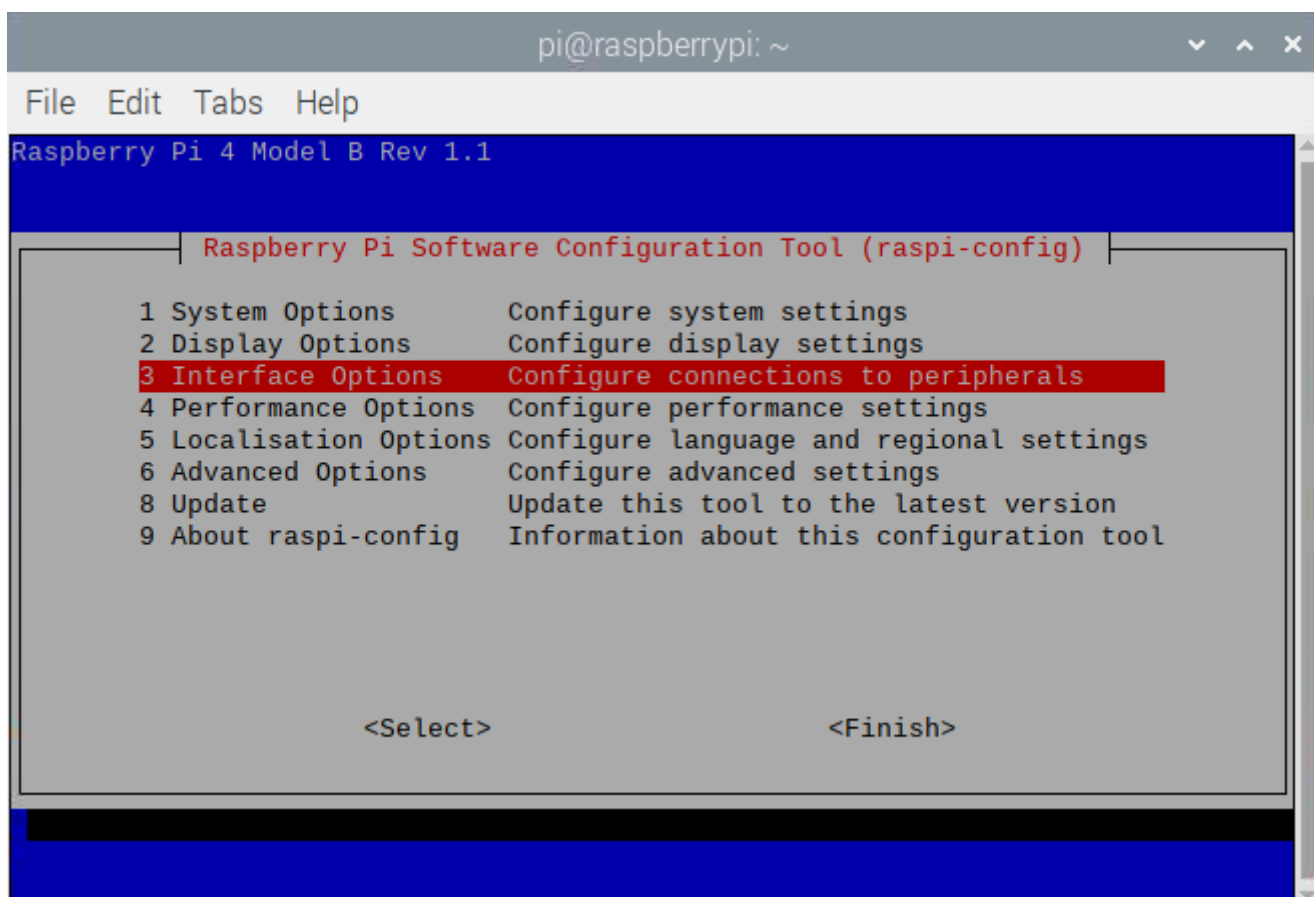
- GPIO 5 par exemple (broche 29)
- Le GND 0V (broche 6)

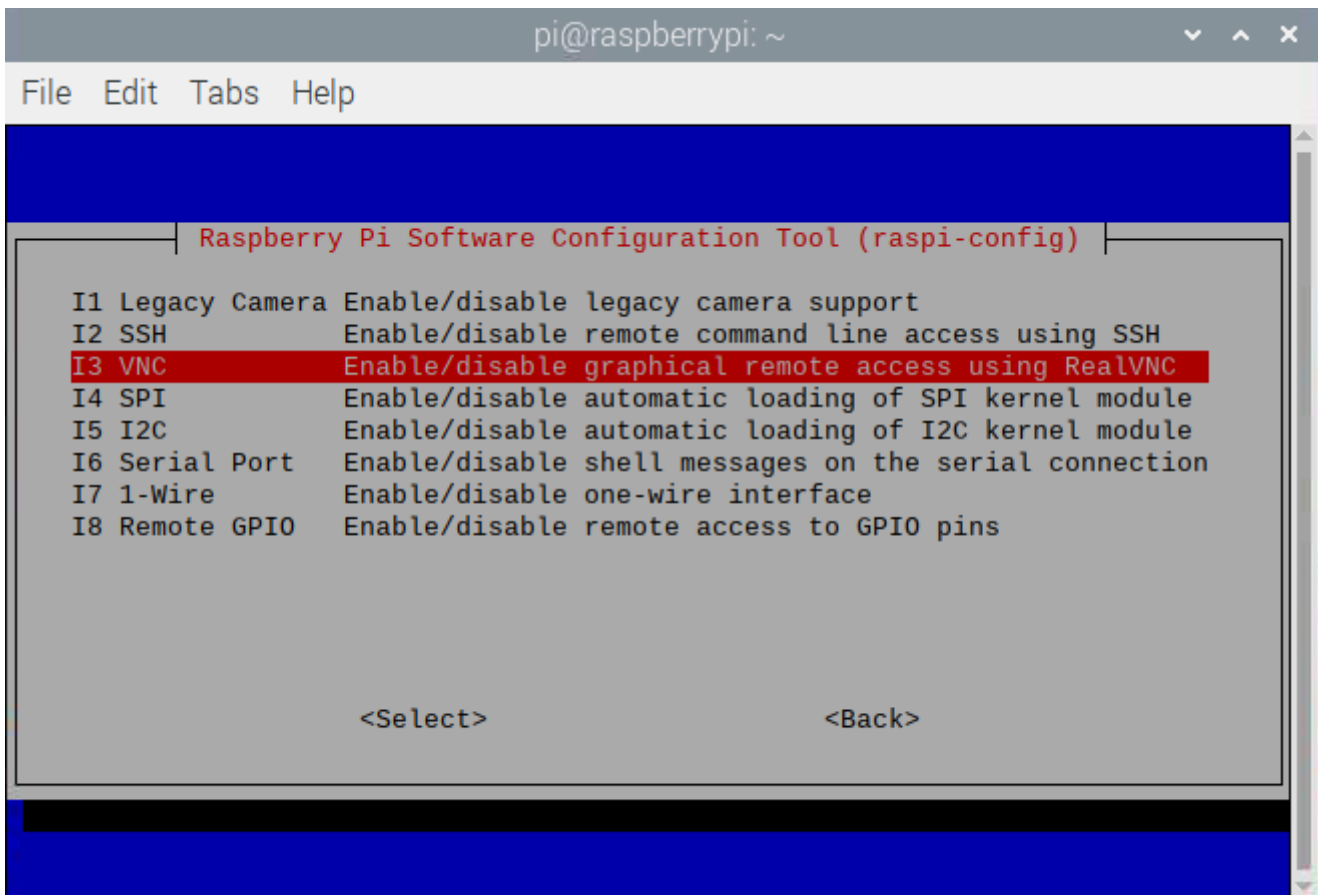
# Validation GPIO avec Raspi-Config

Il est nécessaire que les sorties GPIO soit activées au niveau de la configuration du Raspberry Pi. Depuis un terminal ou une liaison SSH, lancer la commande :

```
sudo raspi-config
```

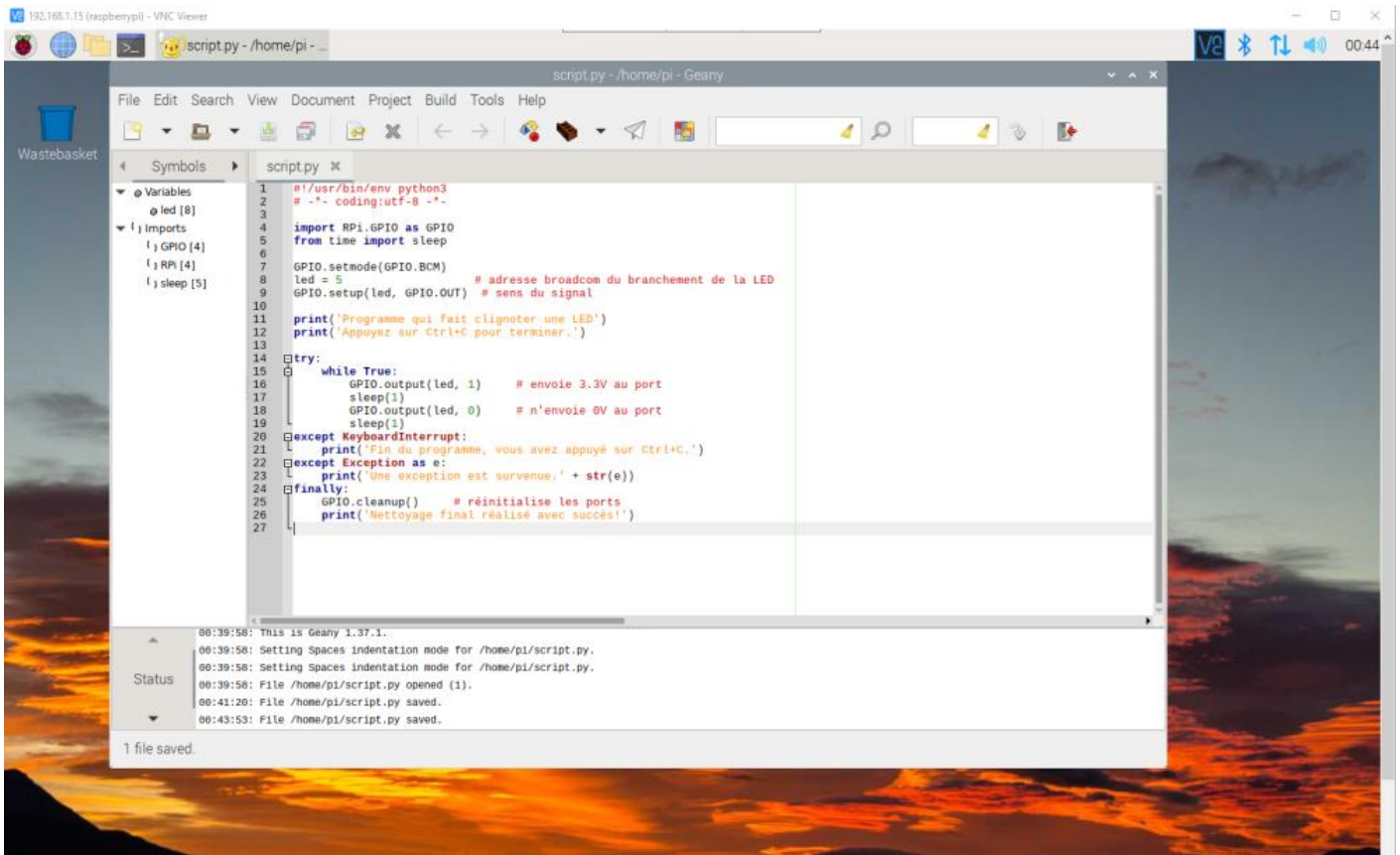
Dans Interface Options -> Activer **I8 Remote GPIO**





# Programmation depuis VNC

On peut utiliser Geany comme éditeur de texte pour écrire le script Python.



Ci-dessous, un exemple de code pour faire clignoter une Led branchée sur GPIO5:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import RPi.GPIO as GPIO
from time import sleep

GPIO.setmode(GPIO.BCM)
led = 5
GPIO.setup(led, GPIO.OUT)

print('It BLINKs')
print('Ctrl+C to quit')

try:
    while True:
        GPIO.output(led, 1)  # envoie 3.3V au port
        sleep(1)
        GPIO.output(led, 0)  # envoie 0V au port
        sleep(1)
except KeyboardInterrupt:
    print('Fin du programme, vous avez appuyé sur Ctrl+C.')
except Exception as e:
    print('Une exception est survenue.' + str(e))
finally:
    GPIO.cleanup()  # réinitialise les ports
    print('Nettoyage final réalisé avec succès!')
```

```
print('End of')
except Exception as e:
    print('Exception occurred' + str(e))
finally:
    GPIO.cleanup()          # init des ports GPIO
```

Pour rendre le script python exécutable, on remarque sur les deux premières lignes

- le SHA-BANG (SHE-BANG): `#!/usr/bin/env python3`
- l'encodage texte : `# -- coding:utf-8 --`

Un script Python n'est pas un fichier compilé, le SHA-BANG précise l'interpréteur (python3) qui va permettre d'exécuter le script.

Pour rendre le script exécutable, on utilise la commande:

```
chmod + x script.py
```

Ensuite, on peut exécuter le script en faisant `./script.py`

```
./script.py
```

## Limites de la lib RPI.GPIO

- Note that this module is unsuitable for real-time or timing critical applications. This is because you can not predict when Python will be busy garbage collecting. It also runs under the Linux kernel which is not suitable for real time applications - it is multitasking O/S and another process may be given priority over the CPU, causing jitter in your program. If you are after true real-time performance and predictability, buy yourself an Arduino <http://www.arduino.cc> !
- Note that the current release does not support SPI, I2C, hardware PWM or serial functionality on the RPi yet. This is planned for the near future - watch this space! One-wire functionality is also planned.
- Although hardware PWM is not available yet, software PWM is available to use on all channels.

La librairie RPI.GPIO est simple à utiliser et permet rapidement de piloter une sortie TOR ou de lire l'état d'une entrée TOR, par contre, les contraintes temporelles ne sont pas maîtrisées, càd par exemple, lorsque le code met la sortie à 1, l'action réelle de mise à 1 peut prendre un temps variable non maîtrisé.

- Il existe de nombreuses librairies ou méthodes pour piloter les GPIO, des plus simples à celles réservées aux experts. Tout va dépendre de l'application. Ces aspects temporels pourront être développés dans un prochain article.