

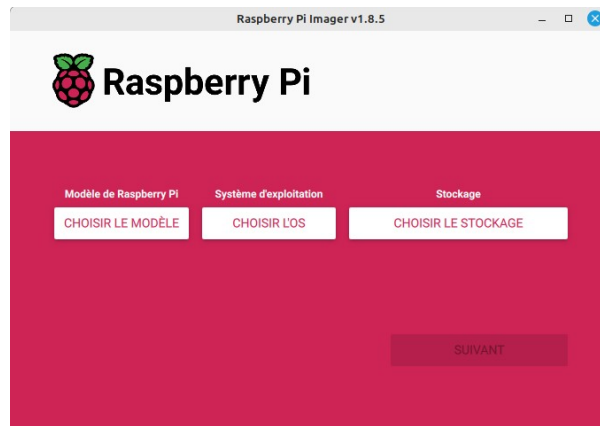
Installation OS sur Raspberry pi zéro 2-W

Table des matières

Installation OS sur Raspberry pi zéro 2-W.....	1
1. Installer Pi Imager.....	2
2. Connexion en SSH.....	3
3. Mise à jour de l'OS.....	4
Vérifier la version de l'OS.....	4
4. Installation de git.....	4
5. Installation de wiringPi.....	5
6. Bus I2C et SPI.....	5
Installation des outils I2C.....	6
7. Utiliser la camera.....	6
Vérifier la présence de la caméra (en tant que root).....	6
Prendre une photo.....	7
Transférer la photo sur le PC.....	7
8. Incruster du texte dans une image.....	8
9. Installer rpitx.....	9
Test de l'installation.....	9
Test de la transmission SSTV.....	10
10. Optimisation des paramètres pour minimiser la consommation.....	10
11. Test de la compilation en C++.....	12
12. Communication file IPC.....	12
13. Monter un système de fichiers en mémoire RAM (ramfs).....	13
14. Lancer le script start_ballon.sh au démarrage.....	13
15. Paramétrer un deuxième PA Wifi.....	14
Obtenir la liste des PA disponibles.....	14
Installer la commande uuidgen.....	14
Créer un fichier pour un deuxième PA.....	14
Créer un uuid.....	14
Vérifier les droits :.....	15
Recharger NetworkManager :.....	15
Vérifier que la connexion est bien enregistrée :.....	15
16. Paramétrer l'horloge matériel.....	16
Vérifier la détection du module RTC DS3231.....	16
Déclarer l'horloge.....	16
Supprimer le faux pilote RTC.....	16
Vérifier la détection au démarrage.....	16
Initialiser l'heure.....	16
17. Désactiver le Bluetooth.....	17

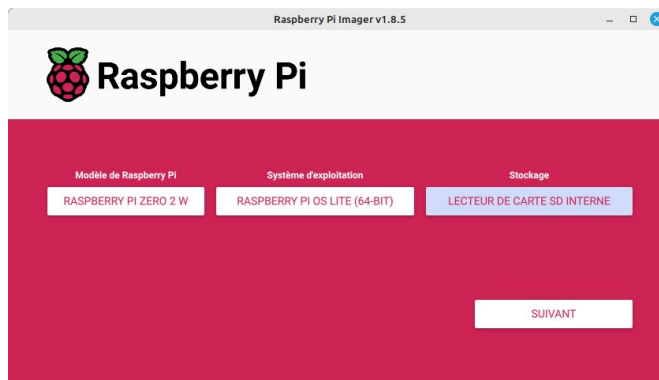
1. Installer Pi Imager

Sur un PC



Configurer

1. le Modèle de Raspberry Pi,
2. le système d'exploitation **Raspberry Pi OS Lite (64-bit)**
3. le stockage choisir le lecteur de carte SD



Utiliser la personnalisation de l'OS

Personnalisation de l'OS

GÉNÉRAL SERVICES OPTIONS

☒ Nom d'hôte : Ballon .local

☒ Définir nom d'utilisateur et mot de passe

Nom d'utilisateur : ballon

Mot de passe : ●●●●

☒ Configurer le Wi-Fi

SSID : Livebox-5648

Mot de passe : ●●●●●●●●●●●●●●●●

☐ Afficher le mot de passe ☐ SSID caché

Pays Wi-Fi : FR

☒ Définir les réglages locaux

Fuseau horaire : Europe/Paris

Type de clavier : fr

ENREGISTRER

Personnalisation de l'OS

GÉNÉRAL SERVICES OPTIONS

☒ Activer SSH

☒ Utiliser un mot de passe pour l'authentification

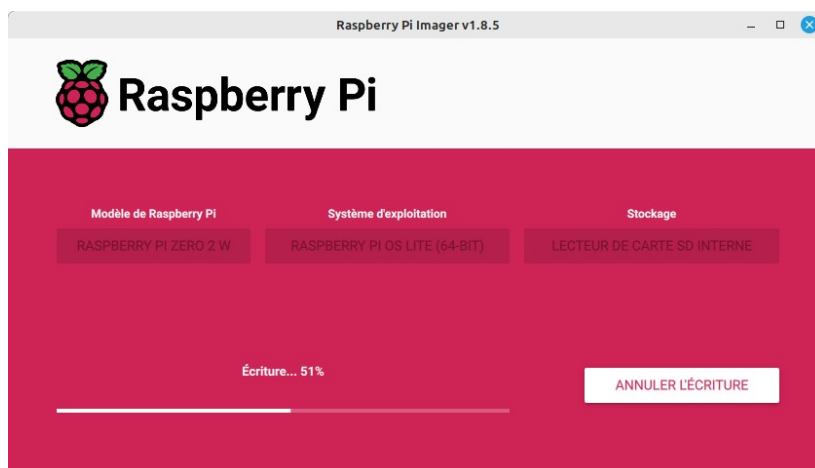
☐ Authentification via clef publique

Définir authorized_keys pour 'ballon' :

LANCER SSH-KEYGEN

ENREGISTRER

puis lancer l'écriture



2. Connexion en SSH

Installer la carte SD, mettre sous tension, retrouver l'adresse IP sur la box

```
philippe@philippe-PC:~$ ssh ballon@192.168.1.47
The authenticity of host '192.168.1.47 (192.168.1.47)' can't be established.
ED25519 key fingerprint is
SHA256:SqFEsR5mTSXqJ9eqCS7cuBB66H5YHgQ3bqUHwg6pJ8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? Yes
Warning: Permanently added '192.168.1.47' (ED25519) to the list of known hosts.
ballon@192.168.1.47's password:
```

Linux Ballon 6.12.25+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.12.25-1+rpt1 (2025-04-30) 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.

3. Mise à jour de l'OS

- `apt update` → rafraîchit la liste des paquets.
- `apt upgrade` → met à jour les paquets **sans suppression**.
- `apt full-upgrade` → met à jour **tout le système**, quitte à installer/supprimer des paquets.

```
apt update
apt full-upgrade
```

Fin Aout 2025 un problème est apparu, depuis les dernières mises à jour de Raspberry Pi OS Sur un **Raspberry Pi**, après un `apt upgrade` ou `full-upgrade`, la commande `libcamera-hello --list-cameras` n'existe plus, elle est remplacée par `rplicam-hello --list-camera`

Vérifier la version de l'OS

```
ballon@ballon:~$ cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 12 (bookworm)"
NAME="Debian GNU/Linux"
VERSION_ID="12"
VERSION="12 (bookworm)"
VERSION_CODENAME=bookworm
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
```

4. Installation de git

```
root@ballon:/home/ballon# apt install git
root@ballon:/home/ballon# git --version
git version 2.39.5
```

5. Installation de wiringPi

```
git clone https://github.com/WiringPi/WiringPi.git
cd WiringPi/
~/WiringPi $ ./build debian
~/WiringPi $ mv debian-template/wiringpi_3.16_arm64.deb .
~/WiringPi $ sudo apt install ./wiringpi_3.16_arm64.deb
```

test de l'installation

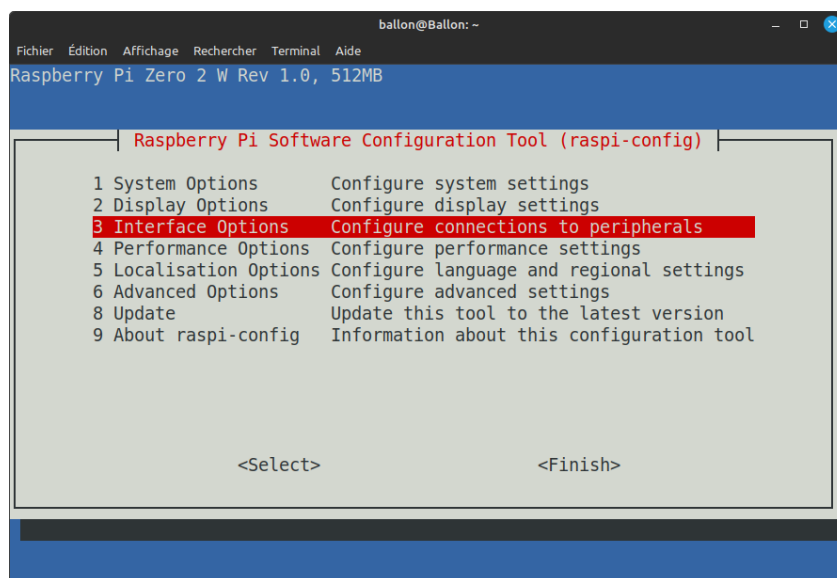
```
ballon@ballon:~/WiringPi $ gpio -v
gpio version: 3.16
Copyright (c) 2012-2025 Gordon Henderson and contributors
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Hardware details:
Type: Pi Zero2-W, Revision: 00, Memory: 512MB, Maker: Sony UK

System details:
* Device tree present.
  Model: Raspberry Pi Zero 2 W Rev 1.0
* Supports full user-level GPIO access via memory.
* Supports basic user-level GPIO access via /dev/gpiomem.
* Supports basic user-level GPIO access via /dev/gpiochip (slow).
```

6. Bus I2C et SPI

Activer les bus i2c et spi avec raspi-config.



Installation des outils I2C

```
apt install i2c-tools
root@Ballon:/home/ballon# i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40: 40  --  --  --  --  --  --  -- 48  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  -- 68 69  --  --  --  --  --  --
70:  --  --  --  --  --  --  -- 77
```

l'adresse 0x40 (ina219) est présente si la carte batterie est connectée au bus I2C

7. Utiliser la camera

rpicam est la bibliothèque qui permet d'accéder et de contrôler la caméra. Elle offre une API C++ aux applications qui permet la configuration de la caméra et de la récupération des images. **Elle est installée par défaut sur PI OS version Bookworm.**

Vérifier la présence de la caméra (en tant que root)

```
root@Ballon:/home/ballon# rpicam-hello --list-camera
Available cameras
-----
0 : imx708 [4608x2592 10-bit RGGB] (/base/soc/i2c0mux/i2c@1/imx708@1a)
  Modes: 'SRGGB10_CSI2P' : 1536x864 [120.13 fps - (768, 432)/3072x1728 crop]
        2304x1296 [56.03 fps - (0, 0)/4608x2592 crop]
        4608x2592 [14.35 fps - (0, 0)/4608x2592 crop]
```



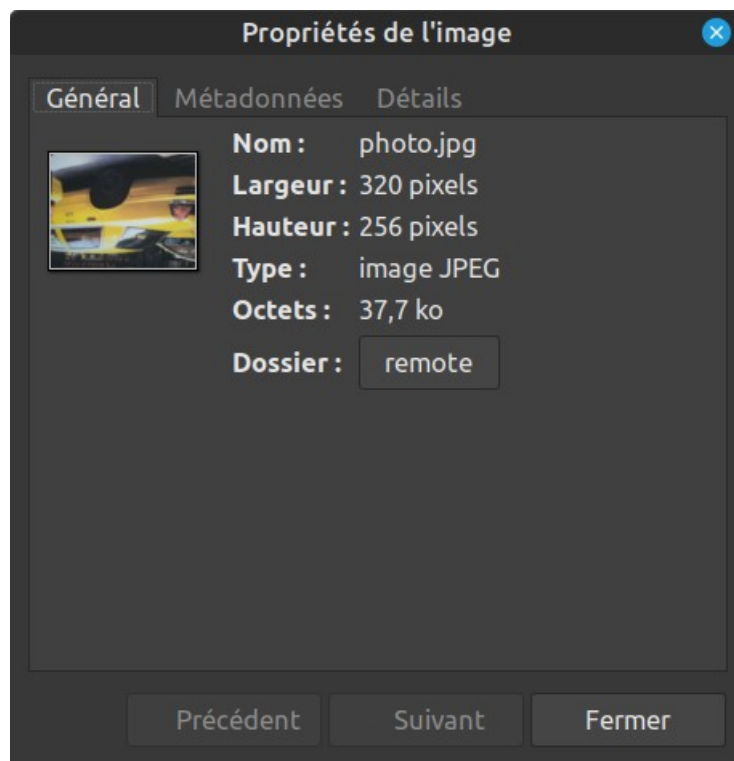
Prendre une photo

```
rpicam-still --width 320 --height 256 -o photo.jpg
```

Transférer la photo sur le PC

```
sshfs ballon@192.168.1.47:/home/ballon remote/
```

Cette commande **permet d'accéder en local (dans remote/) au contenu du répertoire /home/ballon de la machine 192.168.1.47**, comme s'il faisait partie du système de fichiers local.



La suite `sudo apt install libcamera-dev`

```
sudo apt install libcamera-dev
cd /usr/include/libcamera
cd libcamera
sudo cp -r * ../
ballon@Ballon:/usr/include/libcamera $ ls
base          control_ids.h  framebuffer_allocator.h  libcamera      pixel_format.h  transform.h
camera.h      controls.h    framebuffer.h           libcamera.h    property_ids.h  version.h
camera_manager.h  fence.h      geometry.h             logging.h      request.h
color_space.h  formats.h    ipa
```

8. Incruster du texte dans une image

```
apt install imagemagick
```

vérification de la version

```
root@Ballon:/home/ballon# convert --version
```

```
Version: ImageMagick 6.9.11-60 Q16 aarch64 2021-01-25 https://imagemagick.org
```

```
Copyright: (C) 1999-2021 ImageMagick Studio LLC
```

```
License: https://imagemagick.org/script/license.php
```

```
Features: Cipher DPC Modules OpenMP(4.5)
```

```
Delegates (built-in): bzlib djvu fftw fontconfig freetype heic jbig jng jp2 jpeg lcms lqr ltdl lzma  
openexr pangocairo png tiff webp wmf x xml zlib
```

commande pour incruster du texte

```
convert -pointsize 20 -draw "text 10,20 'F4KMN'" photo.jpg photo_texte.jpg
```


9. Installer rpitx

rpitx est un logiciel open-source qui permet à un Raspberry Pi de fonctionner comme un émetteur radio. Le nom "rpitx" est dérivé de "Raspberry Pi" et "transmitter" (émetteur en anglais). Il permet au Raspberry Pi de générer des signaux RF (Radio Fréquence) sur différentes gammes de fréquences.

```
git clone https://github.com/F5OEO/rpitx
cd rpitx
./install.sh
```

A la fin de l'installation un message demande une autorisation pour modifier le fichier /boot/firmware/config.txt

```
In order to run properly, rpitx need to modify /boot/config.txt. Are you sure (y/n) y
Set GPU to 250Mhz in order to be stable
Raspbian 12 detected using /boot/firmware/config.txt
Installation completed !
```

Si on édite ce fichier on retrouve à la fin :

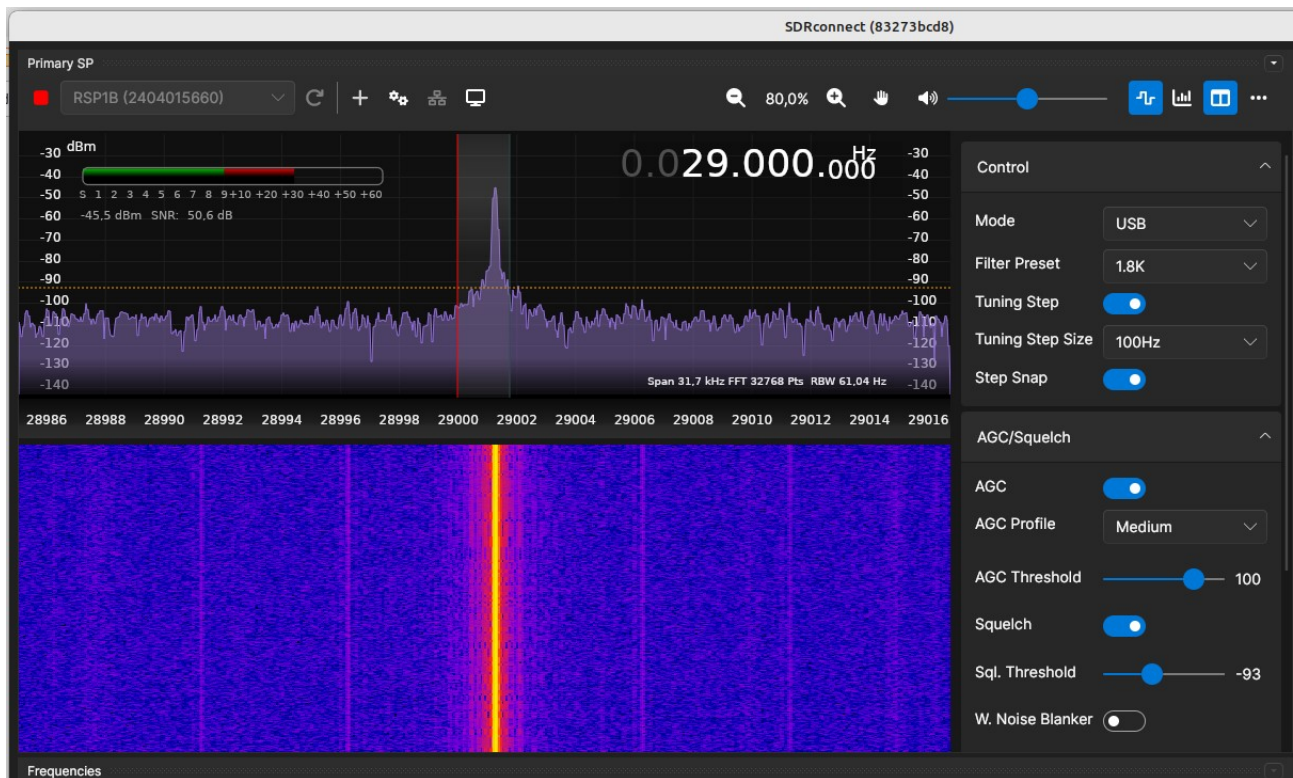
```
[all]
gpu_freq=250
force_turbo=1
```

Définit la fréquence du GPU (processeur graphique VideoCore). Ici, elle est forcée à **250 MHz**. Cela force le GPU à tourner **toujours à la fréquence définie**.

Test de l'installation

Le premier test est d'émettre une porteuse afin de vérifier avec une clé SDR et un récepteur logiciel la bonne réception du signal.

```
root@Ballon:/home/ballon/rpitx# ./tune -f 29001200
```



Comme on peut le voir sur la capture d'écran on obtient un trait pour la fréquence 29,001200 Mhz

Test de la transmission SSTV

/home/ballon/rpitx/pisstv /home/ballon/photo_date.rgb 29000000

La transmission de l'image se fait au format martin 1 sur une durée de **1 minute et 57 secondes**.

10. Optimisation des paramètres pour minimiser la consommation

Sur un Raspberry Pi, le fichier **/boot/firmware/config.txt** sert à définir les paramètres du firmware au démarrage.

```
[all]

# CPU modéré pour économiser la batterie
arm_freq=1000      # CPU à 1 GHz, suffisant pour APRS
arm_freq_min=1000  # Pas de baisse de fréquence (stabilité)

# GPU minimal
gpu_freq=250       # GPU fixé à 250 MHz
gpu_freq_min=250   # Reste toujours à 250 MHz

# Bus interne
core_freq=250      # Bus VPU et USB au minimum stable
sdram_freq=400     # Mémoire en mode économie
```

```
# Pas de survoltage (économie & stabilité)
```

```
over_voltage=0
```

```
# Turbo forcé pour avoir des fréquences fixes et éviter les variations
```

```
force_turbo=1
```

11. Test de la compilation en C++

Copier le répertoire **stop** qui se trouve sur mon dépôt github.

```
root@Ballon:/home/ballon# cd stop/  
root@Ballon:/home/ballon/stop# make all  
gcc -Wall -Wextra -Werror -O2 -c stop.c -o stop.o  
gcc -Wall -Wextra -Werror -O2 -o stop stop.o -lwiringPi
```

Tester l'exécution du programme

```
root@Ballon:/home/ballon/stop# ./stop &  
[1] 942  
root@Ballon:/home/ballon/stop#
```

Appuyer sur le bouton poussoir

```
root@Ballon:/home/ballon/stop# Détection d'un front descendant sur BP GPIO 25  
Broadcast message from root@Ballon on pts/1 (Thu 2025-07-03 11:25:03 CEST):  
The system will power off now!  
Connection to 192.168.1.47 closed by remote host.  
Connection to 192.168.1.47 closed.  
philippe@philippe-PC:~$
```

12. Communication file IPC

Copier le répertoire **lora-files** qui se trouve sur mon dépôt github.

Puis compiler avec la commande make all

Vérifier la présence des 2 files avec la commande ipcs -q

```
root@ballon:/home/ballon/lora_files# ipcs -q  
  
----- Message Queues -----  
  
key      msqid    owner    perms    used-bytes  messages  
0x0000162e 0      root     666      1320        5  
0x0000162f 1      root     666      0           0
```

13. Monter un système de fichiers en mémoire RAM (ramfs)

Montage automatique au démarrage (via `/etc/fstab`)

1 Ouvrir le fichier `/etc/fstab`

2 Ajouter cette ligne à la fin

```
tmpfs /ramfs tmpfs defaults,size=100M 0 0
```

3 Crée le dossier si ce n'est pas fait :

```
sudo mkdir /ramfs
```

après un redémarrage vérifier avec `mount`

```
ballon@ballon:~ $ mount | grep /ramfs
tmpfs on /ramfs type tmpfs (rw,relatime,size=102400k)
```

Tout contenu dans `/ramfs` est **perdu au redémarrage**.

14. Lancer le script `start_ballon.sh` au démarrage

En tant que root

```
crontab -e
```

ajouter la ligne suivante à la fin du fichier

```
@reboot /home/ballon/start_ballon.sh > /home/ballon/log/start_ballon.log 2>&1
```

15. Paramétrer un deuxième PA Wifi

Depuis Raspberry Pi OS Bookworm (basé sur Debian 12), la fondation a remplacé `dhcpcd` + `wpa_supplicant` par **NetworkManager** comme gestionnaire réseau par défaut.















Les connexions seront enregistrées dans NetworkManager et la Pi essaiera de s'y connecter automatiquement quand l'un est disponible.

Obtenir la liste des PA disponibles

En tant que **root** saisir

```
nmcli dev wifi list
```

```
root@ballon:/etc/NetworkManager/system-connections# nmcli dev wifi list
```

IN-USE	BSSID	SSID	MODE	CHAN	RATE	SIGNAL	BARS	SECURITY
	16:CB:D9:5F:9C:3C	PA_philippe	Infra	10	117 Mbit/s	100		WPA2
	44:D4:54:7B:AD:D0	Bbox-41F59A7D	Infra	11	130 Mbit/s	99		WPA2
	44:D4:54:7B:AD:D0	--	Infra	11	130 Mbit/s	92		WPA2
*	58:1D:D8:6B:09:C8	Livebox-5648	Infra	11	405 Mbit/s	74		WPA2
	A4:3E:51:3C:CF:E5	Livebox-CFE5	Infra	6	130 Mbit/s	70		WPA1 WPA2
	10:06:45:BE:59:B0	Livebox-8D88	Infra	1	195 Mbit/s	64		WPA1 WPA2
	58:FC:20:E9:DB:00	SFR DAFF	Infra	6	260 Mbit/s	47		WPA1 WPA2
	10:E9:92:57:20:50	Livebox-2050	Infra	11	130 Mbit/s	29		WPA2
	2C:93:FB:B3:3E:22	Livebox-C910	Infra	6	270 Mbit/s	25		WPA2
	54:EC:B0:15:DE:22	Livebox-5648	Infra	11	270 Mbit/s	25		WPA2
	28:9E:FC:FE:73:90	Bbox-487EEB54	Infra	1	130 Mbit/s	24		WPA1 WPA2
	34:27:92:A4:35:8A	FreeWifi_secure	Infra	6	195 Mbit/s	19		WPA2 802.1X
	44:D4:54:74:5E:A0	Bbox-9AEA39B6	Infra	6	130 Mbit/s	19		WPA1 WPA2
	B8:26:6C:0D:DC:20	Livebox-DC20	Infra	6	130 Mbit/s	14		WPA1 WPA2

```
root@ballon:/etc/NetworkManager/system-connections#
```

Installer la commande uuidgen

```
apt install -y uuid-runtime
```

Créer un fichier pour un deuxième PA

avec une priorité plus faible que le premier

```
sudo nano /etc/NetworkManager/system-connections/DeuxiemePA.nmconnection
```

Créer un uuid

Uuidgen

```
13c928fc-e628-47e5-830d-072878afb213
```

```
[connection]
id=DeuxiemePA
uuid=13c928fc-e628-47e5-830d-072878afb213
type=wifi
autoconnect=true
priority=5

[wifi]
mode=infrastructure
ssid=PA_philippe

[wifi-security]
key-mgmt=wpa-psk
psk=tototata

[ipv4]
method=auto

[ipv6]
method=ignore
```

Vérifier les droits :

```
sudo chown root:root /etc/NetworkManager/system-connections/DeuxiemePA.nmconnection
sudo chmod 600 /etc/NetworkManager/system-connections/DeuxiemePA.nmconnection
```

Recharger NetworkManager :

```
nmcli connection reload
```

Vérifier que la connexion est bien enregistrée :

```
nmcli connection show
```

```
root@ballon:/home/ballon# nmcli connection show
NAME                UUID                                  TYPE      DEVICE
preconfigured      008eaa59-b14a-4c85-a75a-22222e2fe805  wifi      wlan0
lo                  263a9116-6a8f-4b23-8ba2-40e4fd8bf81a  loopback  lo
DeuxiemePA          13c928fc-e628-47e5-830d-072878afb213  wifi      --
```

16. Paramétrer l'horloge matériel

On utilise le module **RTC DS3231** comme horloge matérielle pour que la Raspberry Pi Zero ait la bonne heure même sans réseau (ni NTP).

Vérifier la détection du module RTC DS3231

```
i2cdetect -y 1
```

l'adresse **0x68** devrait être présente.

Déclarer l'horloge

Dans le fichier `/boot/firmware/config.txt` ajouter la ligne suivante :

```
dtoverlay=i2c-rtc,ds3231
```

Supprimer le faux pilote RTC

Par défaut, Raspberry Pi OS utilise un module "fake-hwclock" qui peut écraser l'heure. Il faut le désactiver.

```
sudo apt purge fake-hwclock  
sudo systemctl disable fake-hwclock
```

Vérifier la détection au démarrage

Redémarrer la Pi, puis vérifier que l'horloge est bien prise en charge :

```
dmesg | grep rtc-
```

```
root@ballon:/home/ballon# dmesg | grep rtc-  
[ 11.508708] rtc-ds1307 1-0068: SET TIME!  
[ 11.512795] rtc-ds1307 1-0068: registered as rtc0  
[ 11.513957] rtc-ds1307 1-0068: setting system clock to 2025-09-11T08:17:29 UTC (1757578649)  
root@ballon:/home/ballon#
```

Vous devriez voir la ligne **rtc-ds1307 1-0068: registered as rtc0**

Beaucoup de modules RTC chinois basés sur le DS3231 sont **annoncés comme DS1307** par défaut, car ils sont compatibles et le noyau les attache sur le driver `rtc-ds1307`. Ça ne change rien pour l'utilisation (`/dev/rtc0` fonctionne de la même façon). Seuls les registres `00h-07h` sont utilisés (heures, minutes, secondes, etc.).

Initialiser l'heure

Si l'heure de la DS3231 est vierge, régler-la une première fois avec l'heure système :

```
sudo hwclock -w
```

Et pour relire l'heure de la RTC :


```
sudo hwclock -r
```

17. Désactiver le Bluetooth

Sur une **Raspberry Pi Zero W** (qui intègre Wi-Fi et Bluetooth), on peut désactiver le Bluetooth.

ajouter une ligne dans le fichier de configuration du firmware :

```
sudo nano /boot/firmware/config.txt
```

ajouter à la fin

```
# Désactiver le Bluetooth  
dtoverlay=disable-bt
```

désinstallez la pile Bluetooth. Cela rend le Bluetooth indisponible, même si un adaptateur Bluetooth externe est branché.

```
sudo apt-get purge bluez -y  
sudo apt-get autoremove -y
```