

# Raspberry pi

Sujets à propos du nano-ordinateur Raspberry pi

- [Créer une Music Box sur Raspberry Pi avec MPD](#)
- [Créer une Seedbox sur Raspberry Pi avec Transmission](#)
- [Installer un Raspberry Pi sans écran et en Wi-Fi](#)
- [Sauvegarde et restauration carte SD](#)
- [Transformer un Raspberry Pi en enceinte Bluetooth avec PipeWire](#)

# Créer une Music Box sur Raspberry Pi avec MPD

MPD ou Music Player Daemon est un lecteur audio libre qui permet un accès distant depuis un autre ordinateur.

Possibilité par exemple de contrôler le serveur via Jeeedom avec MPC ou encore via un client comme [M.A.L.P](#) sur Android.

---

## Installation et configuration de MPD

```
sudo apt-get install mpd
```

Une fois MPD installé il faut configurer plusieurs répertoires :

- Répertoire de stockage de la musique
- Répertoire pour l'enregistrement des playlists
- Fichier de base de données

Par sécurité, stockage de ces répertoires et fichiers sur un serveur NAS.

## Montage permanent des répertoires

**Pré-requis :** le répertoire de stockage de la musique doit être partagé sur le réseau.

Dans cet exemple le dossier /media/disqueUSB/Telechargement/Musique sur le NAS est partagé via samba avec les droits en lecture/écriture.

Exemple : [Voir § Partage réseau avec Samba](#)

Créer le répertoire pour le point de montage

```
sudo mkdir /media/musique
```

Editer le fichier fstab

```
sudo nano /etc/fstab
```

Ajouter la ligne suivante pour monter le répertoire réseau dans **/media/musique** :

```
//192.168.1.50/disqueUSB/Telechargement/Musique /media/musique cifs  
rw,username=xxx,password=xxx,uid=1000,gid=1000,_netdev 0 0
```

Monter automatiquement le répertoire :

```
sudo mount -a
```

Création des répertoires pour les playlists et la base de données :

```
mkdir /media/musique/Playlists_MPD  
mkdir /media/musique/DB_MPD/
```

## Configuration de MPD

```
sudo service mpd stop
sudo nano /etc/mpd.conf
```

Modifier les lignes suivantes :

```
music_directory "" /media/musique"
...
playlist_directory "" /media/musique/Playlists_MPD"
...
db_file "" "" /media/musique/DB_MPD/tag_cache"
...
#user "" "" "mpd" (ligne à commenter)
...
bind_to_address "" "192.168.1.89"
...
port "" "" "6600"
...
audio_output {
  type "" "alsa"
  name "" "MusicBox"
  #device "" "hw:0,0" # optional
  mixer_type "" "software"
  ...
}
```

Enregistrer et redémarrer MPD

```
sudo service mpd restart
```

Mettre le volume du alsamixer à 100%

alsamixer

MPD est accessible avec n'importe quel client à l'adresse IP 192.168.1.89 sur le port 6600.  
Si nécessaire, mettre à jour la base de données avec le client.

# Créer une Seedbox sur Raspberry Pi avec Transmission

Procédure pas à pas pour créer une Seedbox avec un Raspberry Pi, un disque dur USB et le logiciel Transmission.

---

## Formater et monter le disque dur USB

Une fois le disque dur USB connecté au Raspberry, utiliser la commande suivante pour lister tous les disques et repérer le bon volume.

```
sudo fdisk -l
```

Exemple de sortie de la commande **fdisk -l**

```
Disque /dev/sdb : 298,1 GiB, 320072933376 octets, 625142448 secteurs
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
Type d'étiquette de disque : dos
Identifiant de disque : 0xf381da99
Périphérique Amorçage Début Fin Secteurs Taille Id Type
/dev/sdb1 2048 625119231 625117184 298,1G 7 HPFS/NTFS/exFAT
```

Dans ce cas le disque USB est **/dev/sdb**, sa première partition est **/dev/sdb1** et son système de fichier NTFS.

Pour créer ou recréer une partition sur le disque **/dev/sdb** on utilise l'outil fdisk :

```
sudo fdisk /dev/sdb
```

Une fois fdisk lancé, commande '**p**' pour lister les partitions du disque sélectionné et être sur qu'il s'agit du bon disque.

Exemple :

```
Commande (m pour l'aide) : p
Disque /dev/sdb : 298,1 GiB, 320072933376 octets, 625142448 secteurs
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
Type d'étiquette de disque : dos
Identifiant de disque : 0xf381da99
Périphérique Amorçage Début Fin Secteurs Taille Id Type
/dev/sdb1 2048 625119231 625117184 298,1G 7 HPFS/NTFS/exFAT
```

Pour supprimer la partition, commande '**d**' :

```
Commande (m pour l'aide) : d
Partition 1 sélectionnée
La partition 1 a été supprimée.
```

Créer une nouvelle partition avec la commande '**n**'. Choisir partition primaire '**p**', puis numéro de partition '**1**' et valider deux fois avec Entrer pour sélectionner le début et la fin par défaut de la partition.

```
Commande (m pour l'aide) : n
Type de partition
  p  primaire (0 primaire, 0 étendue, 4 libre)
  e  étendue (conteneur pour partitions logiques)
Sélectionnez (p par défaut) : p
Numéro de partition (1-4, 1 par défaut) : 1
Premier secteur (2048-625142447, 2048 par défaut) :
Dernier secteur, +secteurs ou +taille{K,M,G,T,P} (2048-625142447, 625142447 par défaut) :

Une nouvelle partition 1 de type « Linux » et de taille 298,1 GiB a été créée.
```

Pour écrire les modifications sur le disque et quitter l'utilitaire **fdisk**, il faut utiliser la commande '**w**'

Pour finir, il faut formater la partition en **EXT4** :

```
sudo mkfs.ext4 -b 4096 /dev/sdb1
```

Pour monter la partition sur le Raspberry et pouvoir l'utiliser, il faut commencer par créer un point de montage dans **/media**.

```
sudo mkdir /media/disqueUSB
```

Pour monter la partition automatiquement au démarrage grâce à **fstab**, il faut connaître l'**UUID** de la partition. Lancer la commande :

```
sudo blkid /dev/sdb1
```

On obtient quelque chose comme

```
/dev/sdb1: LABEL="FreeAgent" UUID="8E7C36397C361C81" TYPE="ext4" PARTUUID="f381da99-01"
```

Ensuite éditer le fichier **/etc/fstab**

```
sudo nano /etc/fstab
```

Ajouter les lignes ci-dessous à la fin du fichier, renseigner l'**UUID** sans les guillemets.

```
# disque usb /dev/sdb1  
UUID=8E7C36397C361C81 /media/disqueUSB/ ext4 defaults 0 2
```

Le répertoire **/dev/disk/by-uuid/** qui recense les disques par UUID n'est mis à jour que pendant le démarrage. Pour le mettre à jour manuellement :

```
sudo udevadm trigger
```

Monter la partition :

```
sudo mount /media/disqueUSB
```

Pour vérifier si la partition est bien montée :

```
mount | grep /media/
```

Retour de la commande :

```
/dev/sdb1 on /media/disqueUSB type fuseblk  
(rw,relatime,user_id=0,group_id=0,default_permissions,allow_other,blksize=4096)
```

La partition **sdb1** est bien montée dans **/media/disqueUSB**. Elle sera montée automatiquement à chaque redémarrage.

Changer le propriétaire du disque par l'utilisateur **pi** et régler les autorisations de lecture / écriture :

```
sudo chown pi:pi /media/disqueUSB  
sudo chmod 755 /media/disqueUSB
```

## Installer et configurer Transmission

Installer le daemon Transmission :

```
sudo apt-get install transmission-daemon
```

Créer un répertoire **Telechargements** dans **/media/disqueUSB**

```
mkdir /media/disqueUSB/Telechargements  
sudo chown debian-transmission:debian-transmission /media/disqueUSB/Telechargements  
sudo chmod 755 /media/disqueUSB/Telechargements
```

Stopper le daemon avant de modifier le fichier de configuration.

```
sudo service transmission-daemon stop
```

Le fichier de configuration se trouve dans **/etc/transmission-daemon**, à modifier pour paramétrer le dossier des téléchargements et l'accès à l'interface web.

```
sudo nano /etc/transmission-daemon/settings.json
```

Chercher et modifier les lignes suivantes :

```
"download-dir": "/media/disqueUSB/Telechargements",  
"port-forwarding-enabled": true,  
"rpc-password": "mot_de_passe",  
"rpc-username": "nom_d_utilisateur",  
"rpc-whitelist-enabled": false,  
"umask": 0,
```

Démarrer le daemon Transmission

```
sudo service transmission-daemon start
```

L'interface Web de Transmission devrait être accessible dans un navigateur sur le réseau local à l'adresse [http://IP\\_RASPBERRY:9091](http://IP_RASPBERRY:9091).

Se connecter avec le **username** et le **password** définis dans le fichier de configuration.

## Partage réseau avec Samba

Le logiciel Samba est un outil permettant de partager des dossiers et des imprimantes à travers un réseau local. Il permet de partager et d'accéder aux ressources d'autres ordinateurs fonctionnant sous Windows et GNU/Linux. Le but ici est de partager sur le réseau local le contenu du disque dur USB.

Installer samba :

```
sudo apt-get install samba
```

Le fichier de configuration de samba se trouve dans **/etc/samba/**. Ouvrir le fichier de configuration avec nano :

```
sudo nano /etc/samba/smb.conf
```

Ajouter la définition de partage suivante à la fin du fichier :

```
[disqueUSB]
comment = Disque dur USB
path = /media/disqueUSB
browseable = yes
guest ok = yes
writeable = yes
public = yes
security = share
create mask = 0777
directory mask = 0777
```

Redémarrer le service samba :

```
sudo service smb restart
```

Le disque dur monté dans **/media/disqueUSB** devrait être accessible sur le réseau.

## Ouverture et redirection des ports du routeur

Afin que Transmission fonctionne correctement et que l'interface Web soit accessible via internet, les port **51413** et **9091** doivent être ouverts et redirigés dans le routeur.

Sur une Livebox, ouvrir l'interface Web et se connecter, aller dans l'onglet "**configuration avancée**" puis dans l'onglet "**DHCP**". Dans la section "**Baux DHCP statiques**", choisir le nom du Raspberry et l'ajouter dans la liste pour fixer son adresse ip locale.

Toujours dans l'onglet "**configuration avancée**" puis dans l'onglet "**NAT/PAT**", ajouter une règle de redirection, nom du service "**Transmission**", port interne et externe **51413**, protocole "**les**

**deux**" et choisir le Raspberry dans la liste des appareils.

Ajouter une seconde règle, nom du service "**Client Transmission**", port interne et externe **9091**, protocole "**les deux**" et choisir le Raspberry dans la liste des appareils.

L'interface Web de Transmission devrait être accessible à partir d'une connexion 4G à l'adresse [http://IP\\_BOX\\_INTERNET:9091](http://IP_BOX_INTERNET:9091).

## Mise à jour DNS dynamique avec ddclient

**ddclient** permet d'automatiser la mise à jour d'une adresse IP dynamique reliée à un nom de domaine.

S'inscrire sur le service <https://www.noip.com>. Dans le dashboard, ajouter un hostname en choisissant l'un des domaines proposés. Exemple : **choukahost.ddns.net**

Installer **ddclient** et passer la configuration rapidement en choisissant n'importe quoi.

```
sudo apt-get install ddclient
```

Éditer le fichier de configuration avec nano.

```
sudo nano /etc/ddclient.conf
```

Exemple de fichier de configuration :

```
daemon=600          # check every 600 seconds
protocol=noip
use=web, web=checkip.dyndns.com/, web-skip='IP Address'
server=dynupdate.no-ip.com
login=username
password='motdepasse'
choukahost.ddns.net
```

Redémarrer le daemon ddclient.

```
sudo service ddclient restart
```

Pour faire un essai, modifier l'IP du hostname sur noip.com et mettre 0.0.0.0 puis redémarrer le daemon ddclient. L'adresse IP devrait être mise à jour automatiquement.

L'interface Web de Transmission devrait être accessible à l'adresse suivante **<http://choukahost.ddns.net:9091>** depuis n'importe quelle connexion internet. Avec une Livebox, l'adresse ne fonctionne pas en local.

## Clients Linux, Windows et Android

### **TransGUI - Transmission Remote GUI :**

Pour Windows et Linux

<https://github.com/transmission-remote-gui/transgui/releases>

### **Gearshift :**

Pour Android

<https://github.com/urandom/gearshift/releases>

**Transmission Remote :**

Pour Android

<https://play.google.com/store/apps/details?id=net.yupol.transmissionremote.app>

**Transmission easy client :**

Extension Chrome

<https://chrome.google.com/webstore/detail/transmission-easy-client/cmiphbjkffbcbnjjaidnjhahned>

**Transmission easy client :**

Extension Firefox

<https://addons.mozilla.org/fr/firefox/addon/transmission-easy-client/>

# Installer un Raspberry Pi sans écran et en Wi-Fi

Guide pour installer Raspbian sur un Raspberry Pi, sans écran ni clavier et avec une connexion SSH en Wi-Fi.

Basé sur le guide de [Sylvain Leroux](#).

## Télécharger l'image de Raspbian

Les images de Raspbian sont disponibles sur <https://www.raspberrypi.org/downloads/raspbian/>.



Pour télécharger directement les dernières versions :

Raspbian Desktop : [https://downloads.raspberrypi.org/raspbian\\_latest](https://downloads.raspberrypi.org/raspbian_latest)

Raspbian Lite : [https://downloads.raspberrypi.org/raspbian\\_lite\\_latest](https://downloads.raspberrypi.org/raspbian_lite_latest)

## Flasher l'image sur la SD

Décompresser l'image de Raspbian.

Télécharger et installer Etcher sur <https://etcher.io/>

Sélectionner l'image de Raspbian, Etcher sélectionne automatiquement la carte SD et cliquer sur Flash!



Etcher 1.4.4 - Ubuntu 18.04

## Activer le SSH au démarrage

Une fois l'image flashée sur la carte SD, celle-ci disposera de deux partitions : boot & rootfs. Pour activer le SSH au démarrage du Raspberry, il suffit de créer un fichier vide nommé "ssh" à la racine de la partition boot.

```
cd /media/cartesd/boot  
touch ssh
```

## Activer le Wi-Fi au démarrage

wpa\_supplicant est le logiciel utilisé pour la connexion aux réseaux Wi-Fi. Il est pré-installé sur Raspbian.

Pour que le Raspberry se connecte au réseau Wi-Fi au démarrage, il faut créer un fichier wpa\_supplicant.conf de base à la racine de la partition boot et y ajouter les informations de connexion.

```
nano wpa_supplicant.conf
```

Copier et coller

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev  
country=FR
```

Ajouter le SSID et le mot de passe du réseau wi-Fi avec wpa\_passphrase :

```
wpa_passphrase Livebox-6052 | sed '/#/d' >> wpa_supplicant.conf
```

Entrer le mot de passe du réseau et valider.

Afficher le contenu de wpa\_supplicant.conf pour vérifier :

```
cat wpa_supplicant.conf
```

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
country=FR
network={
ssid="Livebox-6052"
psk=fde58ecfe3934542cb71dfea8b9076e1ad93869504f1d2
}
```

## Démarrer le Raspberry et se connecter en SSH

Pour se connecter en SSH il faut d'abord trouver l'adresse IP du Raspberry grâce à tcpdump en écoutant le trafic sur DHCP et ARP.

```
sudo tcpdump -lenqti any port 68 or arp | grep "^ B"
```

Avec l'outil de capture lancé, démarrer le Raspberry et après quelques secondes tcpdump devrait afficher du trafic provenant de la carte.

Une fois l'adresse IP identifiée il est possible de se connecter au Raspberry en SSH :

```
ssh pi@IP_RASPBERRY
```

Une fois connecté, la première chose à faire est de changer le mot de passe par défaut :

```
passwd  
sudo raspi-config
```

# Sauvegarde et restauration carte SD

Mise en place d'un script de sauvegarde de la carte SD d'un Raspberry vers un disque dur ou un emplacement réseau et mode opératoire pour la restauration sur une nouvelle carte SD.

---

## Emplacement de la sauvegarde

Commencer par créer un emplacement pour la sauvegarde de la carte SD.

L'emplacement peut être sur un disque dur local ou sur un disque réseau monté sur le Raspberry à sauvegarder.

Sur la machine hébergeant le disque dur de sauvegarde :

```
mkdir /home/partage/Sauvegarde_SD
```

Si le disque dur est en local sur la machine à sauvegarder, passer directement au chapitre suivant "script de sauvegarde". Sinon réaliser les étapes ci-dessous pour monter le disque dur en réseau sur la machine à sauvegarder.

**Pré-requis :** le disque dur de sauvegarde doit être partagé sur le réseau local.

Dans cet exemple le dossier **/home/partage** est partagé via samba avec les droits en lecture/écriture.

Exemple : [Voir § Partage réseau avec Samba](#)

Sur le Raspberry à sauvegarder, créer le répertoire pour le montage du disque réseau :

```
sudo mkdir /mnt/Sauvegarde_SD
```

Editer le fichier **/etc/fstab** pour monter le répertoire au démarrage :

```
sudo nano /etc/fstab
```

Ajouter la ligne suivante, penser à renseigner le username et password :

```
//192.168.1.50/partage/Sauvegarde_SD /mnt/Sauvegarde_SD cifs  
rw,username=xxx,password=xxx,uid=1000,gid=1000,_netdev 0 0
```

Avec **192.168.1.50** = ip locale de la machine hébergeant le disque dur et **partage** = [nom du partage] configuré dans **/etc/samba/smb.conf**.

Monter le disque :

```
sudo mount /mnt/Sauvegarde_SD
```

Vérifier si le montage a fonctionné :

```
mount | grep /mnt/Sauvegarde_SD
```

## Script de sauvegarde

Créer le script **Save\_SD.sh** dans le répertoire **/home/pi** du Raspberry à sauvegarder.

```
nano Save_SD.sh
```

Copier - coller le script ci-dessous et enregistrer. Penser à ajuster la variable **DOSSIER**. Si nécessaire, dé-commenter les lignes pour l'arrêt et le redémarrage des services Nginx, PHP et MySQL. Le log des sauvegardes se fait dans **/home/pi/Save\_SD.log**.

```
#!/bin/bash
exec 3>&1 1>>/home/pi/Save_SD.log 2>&1
DATE=$(date +"%Y-%m-%d %H%M")
DOSSIER=/mnt/Sauvegarde_SD
BOOT=Partition_1.tar.gz
SYSTEME=Partition_2.tar.gz

if grep -qs $DOSSIER /proc/mounts; then
    echo "==== Début de la sauvegarde $(date +"%Y-%m-%d %H%M") =====" | tee /dev/fd/3
    #[]echo "Arrêt des services nginx, php, mysql" | tee /dev/fd/3
    #[]sudo service nginx stop
    #[]sudo service php5-fpm stop
    #[]sudo service mysql stop
    []echo "$(date +"%Y-%m-%d %H%M") : Début de la sauvegarde de /boot/" | tee /dev/fd/3
        sudo tar -cpf $DOSSIER/$BOOT /boot/ --one-file-system
    []RESULT=$?
    []if [ $RESULT = 0 ]; then
        echo "$(date +"%Y-%m-%d %H%M") : Sauvegarde de /boot/ effectuée dans $BOOT" | tee
/dev/fd/3
    []elif [ $RESULT = 1 ]; then
        echo "$(date +"%Y-%m-%d %H%M") : Sauvegarde de /boot/ effectuée dans $BOOT mais
certains fichiers peuvent différer" | tee /dev/fd/3
    []else
    []    echo "$(date +"%Y-%m-%d %H%M") : Erreur lors de la sauvegarde de /boot/" | tee /dev/fd/3
    []fi
    []echo "$(date +"%Y-%m-%d %H%M") : Début de la sauvegarde de /" | tee /dev/fd/3
```

```
□sudo tar -cpf $DOSSIER/$SYSTEME / --one-file-system
□RESULT=$?
□if [ $RESULT = 0 ]; then
    echo "$(date +"%Y-%m-%d %Hh%M") : Sauvegarde de / effectuée dans $SYSTEME" | tee
/dev/fd/3
□elif [ $RESULT = 1 ]; then
    echo "$(date +"%Y-%m-%d %Hh%M") : Sauvegarde de / effectuée dans $SYSTEME mais
certains fichiers peuvent différer" | tee /dev/fd/3
□else
□    echo "$(date +"%Y-%m-%d %Hh%M") : Erreur lors de la sauvegarde de /" | tee /dev/fd/3
□fi
#□echo "Redémarrage des services nginx, php, mysql" | tee /dev/fd/3
#□sudo service mysql start
#□sudo service nginx start
#□sudo service php5-fpm start
    echo "==== Fin de la sauvegarde $(date +"%Y-%m-%d %Hh%M") =====" | tee /dev/fd/3
□exit 0
else
    echo "$(date +"%Y-%m-%d %Hh%M") : Erreur, le repertoire de sauvegarde n'est pas monté" |
tee /dev/fd/3
□exit 0
fi
```

Rendre le script exécutable :

```
chmod +x Save_SD.sh
```

Lancer la sauvegarde avec :

```
sudo ./Save_SD.sh
```

Une fois la sauvegarde terminée, vérifier la présence des archives **Partition\_1.tar.gz** (~20 Mo) et **Partition\_2.tar.gz** (>1,80 Go).

```
ls -l /mnt/Sauvegarde_SD/
```

## Programmer la sauvegarde

Programmer la sauvegarde automatique de la carte SD tous les 6 mois avec crontab :

```
sudo crontab -e
```

Ajouter la ligne ci-dessous à la fin et enregistrer.

```
0 4 1 6,12 * sudo sh /home/pi/Save_SD.sh
```

La sauvegarde se lancera les 1er juin et 1er décembre à 4h00.

Pour personnaliser la programmation : <https://crontab.guru/>

## Restauration de la carte SD

Utiliser **gparted** ou **fdisk** pour créer 2 partitions, la première de 50 Mo qui sera de type W95 et formatée en FAT32 et la seconde du reste de l'espace disponible de type Linux et formatée en ext4.

Repérer la carte SD et lancer fdisk (penser à démonter les partitions)

```
sudo fdisk -l
sudo fdisk /dev/sdd
```

- Lister les partitions : commande "p"
- Supprimer les partitions existantes : commande "d"
- Créer une nouvelle table vide de partitions DOS : commande "o"
- Ajouter une première partition : commande "n", p (primaire), numéro 1, premier secteur par défaut, +50M (taille)
- Ajouter une seconde partition : commande "n", p (primaire), numéro 2, premier secteur par défaut, deuxième secteur par défaut
- Modifier le type de la partition 1 : commande "t", numéro 1, code hexa "c"
- Sauvegarder et quitter : commande "w"

Formater et nommer les partitions.

Label première partition : **boot**

Label seconde partition : **rootfs**

```
sudo mkfs.vfat -F 32 -n boot /dev/sdd1
sudo mkfs.ext4 -b 4096 -L rootfs /dev/sdd2
```

Monter les partitions avec l'explorateur de fichier ou manuellement.

```
sudo mkdir /media/sdcard/boot
sudo mkdir /media/sdcard/rootfs
sudo mount /dev/sdd1 /media/sdcard/boot
sudo mount /dev/sdd2 /media/sdcard/rootfs
```

Extraire les archives de sauvegarde **Partition\_1.tar.gz** et **Partition\_2.tar.gz** respectivement sur les partitions **boot** et **rootfs**.

```
sudo tar -xvpf /chemin/vers/Partition_1.tar.gz -C /media/sdcard/boot/ --numeric-owner
sudo tar -xvpf /chemin/vers/Partition_2.tar.gz -C /media/sdcard/rootfs/ --numeric-owner
```

Une fois l'extraction terminée, déplacer tous les fichiers dans **/media/sdcard/boot/boot** vers **/media/sdcard/boot**.

```
mv /media/sdcard/boot/boot/* /media/sdcard/boot/
rm -R /media/sdcard/boot/boot
```

Démonter les partitions et retirer la carte SD.

```
sudo umount /media/sdcard/*
```

Il ne reste plus qu'à booter sur la nouvelle carte SD depuis le Raspberry.

# Transformer un Raspberry Pi en enceinte Bluetooth avec PipeWire

Configuration d'un Raspberry Pi pour qu'il fonctionne comme une enceinte Bluetooth, permettant de diffuser de la musique depuis un téléphone, une tablette ou un ordinateur vers des enceintes connectées au Pi.

## Prérequis matériels

- Raspberry Pi (modèle 3 ou ultérieur avec Bluetooth intégré)
- Carte SD avec Raspberry Pi OS Lite déjà installé : [voir "Installer un Raspberry Pi sans écran et en Wi-Fi"](#)
- Câble audio jack 3.5mm vers RCA
- Enceintes ou système audio avec entrée analogique RCA.

## Configuration initiale du système

### Configuration de la connexion automatique

Pour permettre une connexion automatique sans interface utilisateur :

```
sudo raspi-config
```

Naviguez vers :

- Option 1 : System Options
- Option S5 : Boot / Auto Login
- Sélectionnez Console Autologin

Raspberry Pi OS se connectera automatiquement à l'utilisateur créé lors de l'installation, ce qui lancera automatiquement PipeWire et WirePlumber.

## Installation des paquets nécessaires

Mettez à jour les dépôts et installez PipeWire, WirePlumber avec les dépendances Bluetooth et Python :

```
sudo apt update
sudo apt install pipewire wireplumber libspa-0.2-bluetooth python3-dbus
```

# Configuration Bluetooth

## Téléchargement des scripts de gestion

Téléchargez le script de gestion Bluetooth et son fichier de service systemd directement depuis le dépôt GitHub :

```
cd ~
wget https://raw.githubusercontent.com/fdanis-oss/pw_wp_bluetooth_rpi_speaker/master/speaker-agent.py
wget https://raw.githubusercontent.com/fdanis-oss/pw_wp_bluetooth_rpi_speaker/master/.config/systemd/user/speaker-agent.service
```

Rendre le script exécutable :

```
chmod +x speaker-agent.py
```

## Configuration du script Bluetooth

Pour autoriser plusieurs connexions simultanées (mixage audio), utilisez la configuration par défaut qui permet à plusieurs appareils de se connecter et de mixer leur audio.

Pour n'autoriser qu'une seule connexion à la fois, modifiez le fichier de service pour ajouter l'option `--single-connection` :

```
sed -i 's|ExecStart=python speaker-agent.py|ExecStart=python speaker-agent.py --single-connection|' speaker-agent.service
```

## Installation du service systemd

Créez le répertoire utilisateur pour systemd si nécessaire :

```
mkdir -p ~/.config/systemd/user
```

Déplacez le fichier de service au bon emplacement :

```
mv speaker-agent.service ~/.config/systemd/user/
```

Activez et démarrez le service :

```
systemctl --user daemon-reload
systemctl --user enable speaker-agent.service
systemctl --user start speaker-agent.service
```

## Configuration de BlueZ pour appairage automatique

Modifiez la configuration Bluetooth pour permettre l'appairage sans interaction :

```
sudo sed -i 's/#JustWorksRepairing.*/JustWorksRepairing = always/' /etc/bluetooth/main.conf
```

Redémarrez le service Bluetooth :

```
sudo systemctl restart bluetooth
```

## Utilisation

1. Connectez les enceintes : branchez les enceintes ou système Hi-Fi sur la prise jack 3.5mm du Raspberry Pi
2. Redémarrez le système :

```
sudo reboot
```

3. Appairez un appareil :

- Sur le téléphone/tablette/ordinateur, activez le Bluetooth
- Recherchez les nouveaux appareils
- Sélectionnez raspberrypi (ou un nom similaire)
- L'appairage se fera normalement automatiquement sans code PIN requis

4. Diffusez de l'audio : une fois connecté, l'audio devrait être automatiquement redirigé vers les enceintes. Le volume peut être contrôlé depuis l'appareil source

## Dépannage

### Vérifier l'état des services

```
systemctl --user status speaker-agent.service
systemctl --user status pipewire
systemctl --user status wireplumber
```

## Vérifier les connexions Bluetooth

```
bluetoothctl
# Dans l'interface bluetoothctl :
devices
info <adresse_MAC>
```

## Journal des services

Pour voir les logs de l'agent Bluetooth :

```
journalctl --user -u speaker-agent.service -f
```

## Réinitialiser les appareils appairés

Si vous rencontrez des problèmes de connexion :

```
bluetoothctl
# Dans l'interface bluetoothctl :
remove <adresse_MAC>
# Puis réappairez l'appareil
```

## Résoudre l'erreur "appairage non accepté"

L'erreur "appairage non accepté" lors de la tentative de connexion d'un appareil au Raspberry Pi est un problème courant lié aux profils et services Bluetooth. Voici comment le résoudre, étape par étape.

### 1. Diagnostic initial

Vérification de l'état de l'interface Bluetooth et du service. Ouvrez un terminal sur le Pi et exécutez ces commandes :

Vérifiez que l'adaptateur est listé et qu'il est UP RUNNING.

```
hciconfig -a
```

S'il est marqué DOWN, il faut le réactiver.

Vérifiez que le service Bluetooth est bien actif (active (running)).

```
sudo systemctl status bluetooth
```

Si l'adaptateur est désactivé (DOWN), vous pouvez essayer de le réactiver :

```
sudo rfkill unblock bluetooth
```

## 2. Configuration du profil A2DP

L'erreur "Pas de services utilisables" ou "appairage non accepté" indique souvent que le profil A2DP (Audio Sink) n'est pas correctement publié.

Le gestionnaire audio PulseAudio gère souvent les connexions Bluetooth. Vérifiez si le module nécessaire est installé :

```
sudo apt install pulseaudio-module-bluetooth
```

Après l'installation, rechargez PulseAudio :

```
pulseaudio -k  
pulseaudio --start
```

Parfois, il faut configurer l'agent Bluetooth (bluetoothctl) pour accepter les connexions sur le bon profil. Ouvrez l'outil en ligne de commande :

```
sudo bluetoothctl
```

Dans l'interface bluetoothctl, exécutez les commandes suivantes dans l'ordre :

`agent on` : Active l'agent de gestion des associations.

`default-agent` : Définit l'agent par défaut.

`discoverable on` : Rend le Pi visible.

`pairable on` : Autorise l'appairage.

`scan on` (temporairement) : Pour voir les appareils à proximité.

Ensuite, depuis le téléphone, tentez à nouveau l'appairage. Si une clé est demandée, confirmez-la à la fois sur le téléphone et dans le terminal bluetoothctl.

Une fois l'appairage réussi, assurez-vous de faire confiance (trust) à l'appareil depuis bluetoothctl.

Vérifier les services exposés Après l'appairage, dans `bluetoothctl`, utilisez la commande `info` `<adresse MAC du telephone>` pour lister les services du Pi vus par le téléphone.

Vous devriez voir des UUIDs de services audio comme `0000110a-0000-1000-8000-00805f9b34fb` (Audio Source) ou `0000110b-0000-1000-8000-00805f9b34fb` (Audio Sink). Leur absence confirme le problème de profil.

## Crédits et licence

Ce tutoriel est basé sur le travail de Frédéric Danis :

- Article original : [pw wp bluetooth rpi speaker](#)
- Licence : LGPL-2.1-or-later

Les scripts fournis sont sous licence LGPL-2.1-or-later. Vous êtes libre de les utiliser, modifier et distribuer selon les termes de cette licence.