



pHat DAC & Bluetooth

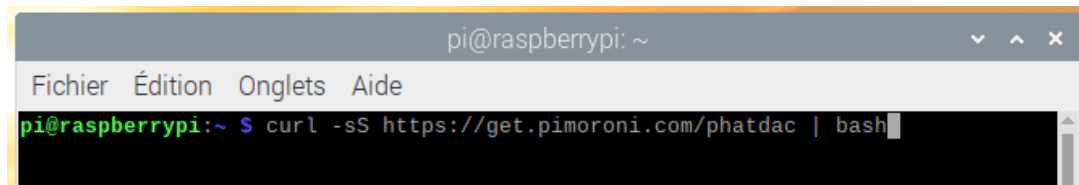
STREAMING AUDIO VIA SMARTPHONE

BOTTIN | Projets Raspberry | Septembre 2019

INSTALLATION D'UN DAC AUDIO EN I2S

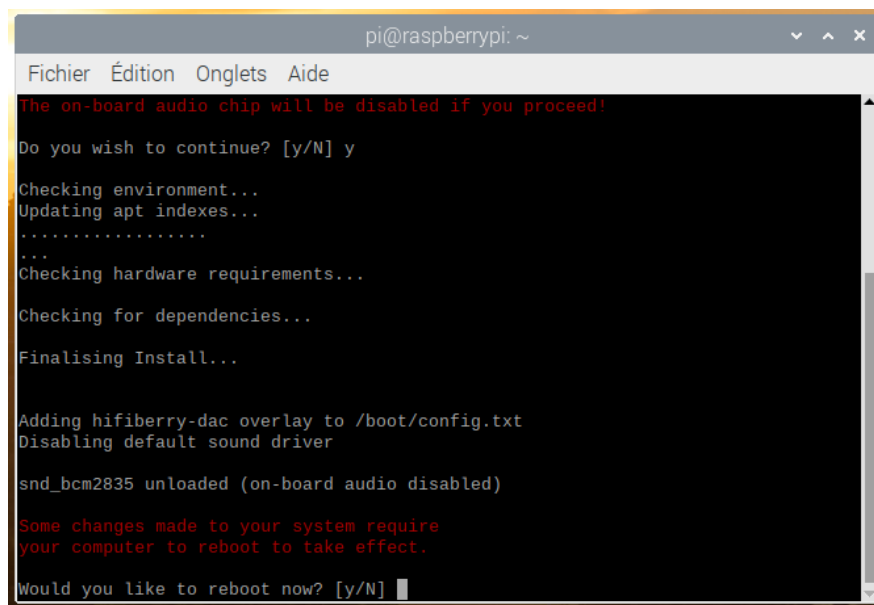
Le DAC utilisé est celui de Pimoroni (pHat DAC) avec un CNA utilisant le protocole I2S. Il utilise les mêmes puces que les DACs HifiBerry et reste donc compatible avec cette large gamme de cartes son pour Raspberry.

La 1^{ère} étape consiste à télécharger et exécuter le fichier BASH fourni par Pimoroni :



```
pi@raspberrypi: ~  
Fichier Édition Onglets Aide  
pi@raspberrypi:~ $ curl -sS https://get.pimoroni.com/phatdac | bash
```

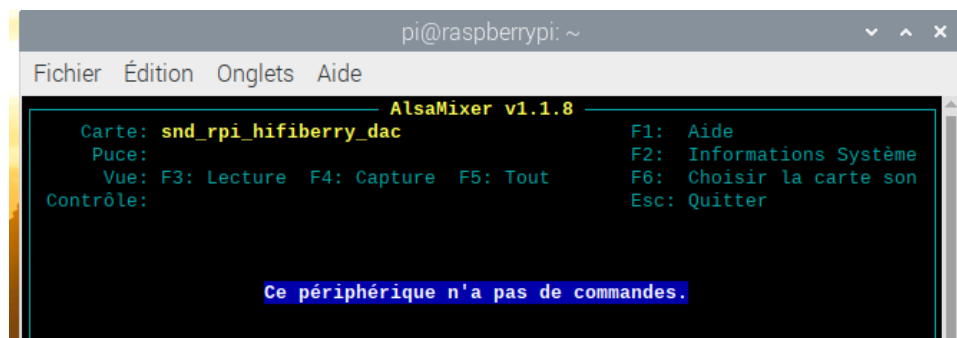
Il nous prévient avant que l'on autorise l'exécution que la puce audio embarquée sur la Raspberry sera désactivée.



```
pi@raspberrypi: ~  
Fichier Édition Onglets Aide  
The on-board audio chip will be disabled if you proceed!  
Do you wish to continue? [y/N] y  
Checking environment...  
Updating apt indexes...  
.....  
...  
Checking hardware requirements...  
Checking for dependencies...  
Finalising Install...  
  
Adding hifiberry-dac overlay to /boot/config.txt  
Disabling default sound driver  
  
snd_bcm2835 unloaded (on-board audio disabled)  
  
Some changes made to your system require  
your computer to reboot to take effect.  
Would you like to reboot now? [y/N]
```

Il vous demande alors de rebooter votre système.

Au retour sur le bureau, vous aurez perdu l'icône de contrôle de volume en haut à droite du bureau. En effet, un lancement de la commande 'alsamixer' dans le terminal nous renseigne que ce driver audio n'a pas de contrôle de volume :

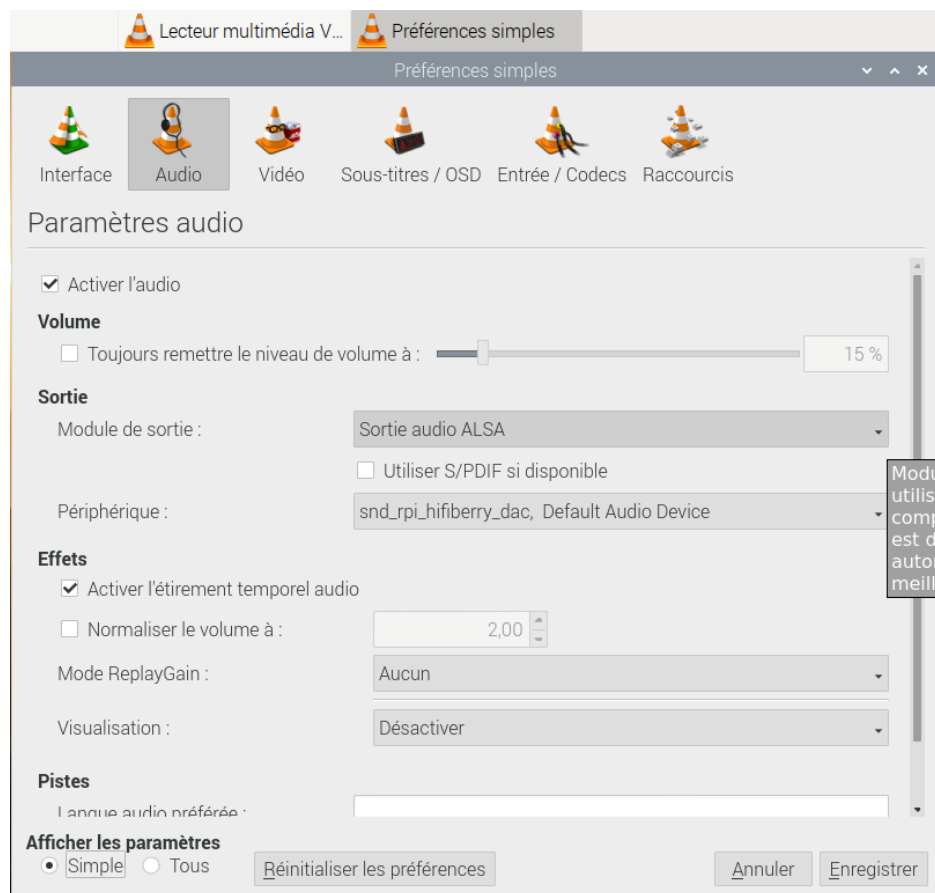


```
pi@raspberrypi: ~  
Fichier Édition Onglets Aide  
AlsaMixer v1.1.8  
Carte: snd_rpi_hifiberry_dac  
Puce:  
Vue: F3: Lecture F4: Capture F5: Tout F1: Aide  
Contrôle: F6: Choisir la carte son F2: Informations Système  
Esc: Quitter  
  
Ce périphérique n'a pas de commandes.
```

Pour vérifier toutefois que le son sort bien par la carte pHat DAC, il vous suffit de lancer un fichier sonore. On peut utiliser VLC pour cela, mais il faut configurer ses paramètres audio.

Lancer le player VLC, puis dans le menu 'outils', ouvrir les préférences.

On change alors le module de sortie et le périphérique :



Une fois, ces paramètres enregistrés, la lecture d'un fichier audio devrait donner un son dans le haut-parleur (à condition bien sûr qu'il soit connecté au jack de la carte pHat DAC et pas au jack de la Raspberry) :



CONFIGURATION DU BLUETOOTH POUR LE STREAMING AUDIO

Avant toute chose, il est nécessaire de mettre à jour le système Raspbian par les commandes suivantes :

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

 (cette manipulation peut prendre environ 30 minutes !!)

Un redémarrage sera nécessaire pour terminer la mise à jour.

Pensez à vérifier que la lecture d'un fichier audio via VLC fonctionne toujours avant de poursuivre.

Remarque : Si vous désirez utiliser un dongle Bluetooth USB plutôt que le Bluetooth intégré, il faut faire les étapes suivantes avant de passer à la suite :

Disabling Integrated Bluetooth

If you are using a separate USB Bluetooth dongle, disable the integrated Bluetooth to prevent conflicts.

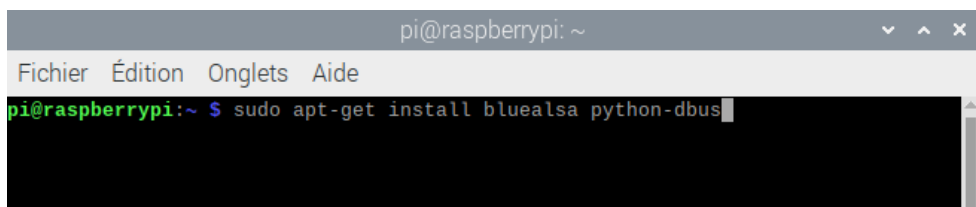
To disable the integrated Bluetooth add the following

```
# Disable onboard Bluetooth
dtoverlay=pi3-disable-bt
```

to `/boot/config.txt` and execute the following command

```
sudo systemctl disable hciuart.service
```

Pour l'utilisation du Bluetooth (interne ou dongle) de la Raspberry, on a besoin d'installer des paquets supplémentaires :

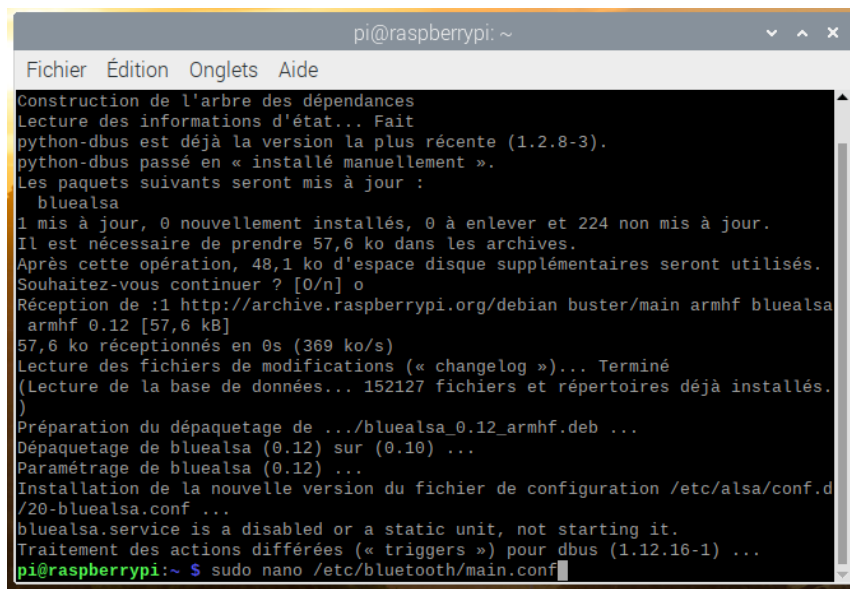


```
pi@raspberrypi: ~
Fichier  Édition  Onglets  Aide
pi@raspberrypi:~ $ sudo apt-get install bluealsa python-dbus
```

Il faut ensuite rendre la raspberry découvrable (discoverable) via le Bluetooth (vérifiez au préalable que le Bluetooth est bien activé via l'icône en haut à droite du bureau → l'icône doit être bleu et non grisé).

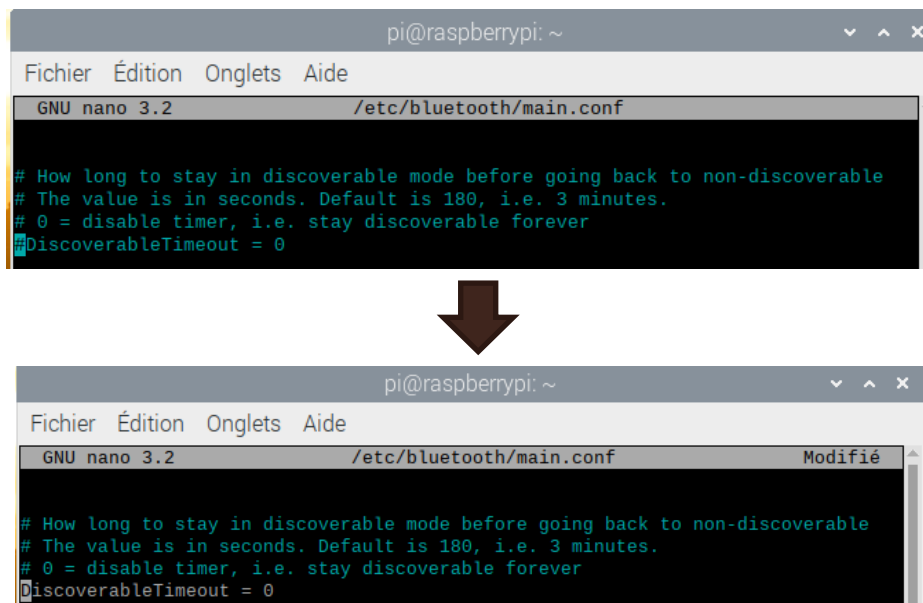
Normalement, un périphérique Bluetooth peut être découvert que sur un temps limité, après il n'est plus identifiable dans l'environnement proche.

Ici, on va faire en sorte qu'il soit décelable tout le temps. Pour cela, vous devez éditer le fichier '/etc/bluetooth/main.conf' :



```
pi@raspberrypi: ~  
Fichier Édition Onglets Aide  
Construction de l'arbre des dépendances  
Lecture des informations d'état... Fait  
python-dbus est déjà la version la plus récente (1.2.8-3).  
python-dbus passé en « installé manuellement ».  
Les paquets suivants seront mis à jour :  
  bluealsa  
1 mis à jour, 0 nouvellement installés, 0 à enlever et 224 non mis à jour.  
Il est nécessaire de prendre 57,6 ko dans les archives.  
Après cette opération, 48,1 ko d'espace disque supplémentaires seront utilisés.  
Souhaitez-vous continuer ? [O/n] o  
Réception de :1 http://archive.raspberrypi.org/debian buster/main armhf bluealsa  
  armhf 0.12 [57,6 kB]  
57,6 ko réceptionnés en 0s (369 ko/s)  
Lecture des fichiers de modifications (« changelog »)... Terminé  
(Lecture de la base de données... 152127 fichiers et répertoires déjà installés.)  
Préparation du dépaquetage de .../bluealsa_0.12_armhf.deb ...  
Dépaquetage de bluealsa (0.12) sur (0.10) ...  
Paramétrage de bluealsa (0.12) ...  
Installation de la nouvelle version du fichier de configuration /etc/alsa/conf.d  
/20-bluealsa.conf ...  
bluealsa.service is a disabled or a static unit, not starting it.  
Traitement des actions différées (« triggers ») pour dbus (1.12.16-1) ...  
pi@raspberrypi:~$ sudo nano /etc/bluetooth/main.conf
```

Dans ce fichier, vous devez chercher la ligne '#DiscoverableTimeout = 0' et la décommenter :

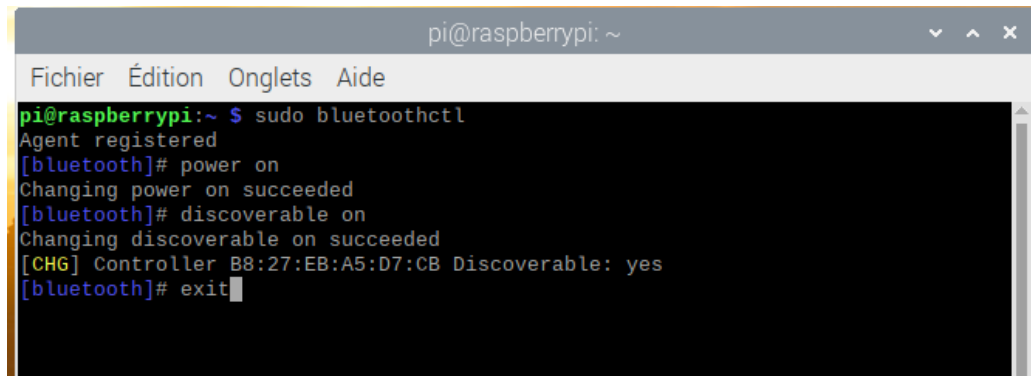


```
pi@raspberrypi: ~  
Fichier Édition Onglets Aide  
GNU nano 3.2 /etc/bluetooth/main.conf  
# How long to stay in discoverable mode before going back to non-discoverable  
# The value is in seconds. Default is 180, i.e. 3 minutes.  
# 0 = disable timer, i.e. stay discoverable forever  
#DiscoverableTimeout = 0
```

↓

```
pi@raspberrypi: ~  
Fichier Édition Onglets Aide  
GNU nano 3.2 /etc/bluetooth/main.conf Modifié  
# How long to stay in discoverable mode before going back to non-discoverable  
# The value is in seconds. Default is 180, i.e. 3 minutes.  
# 0 = disable timer, i.e. stay discoverable forever  
DiscoverableTimeout = 0
```

On doit ensuite rendre la Raspberry découvrable par le téléphone par le biais des 3 commandes suivantes :



```
pi@raspberrypi: ~  
Fichier  Édition  Onglets  Aide  
pi@raspberrypi:~$ sudo bluetoothctl  
Agent registered  
[bluetooth]# power on  
Changing power on succeeded  
[bluetooth]# discoverable on  
Changing discoverable on succeeded  
[CHG] Controller B8:27:EB:A5:D7:CB Discoverable: yes  
[bluetooth]# exit
```

Remarque : vous pouvez aussi passer directement par l'icône Bluetooth et sélectionner 'make discoverable'.

Dans tout les cas, l'icône Bluetooth doit changer continuellement de couleur (bleu/vert).

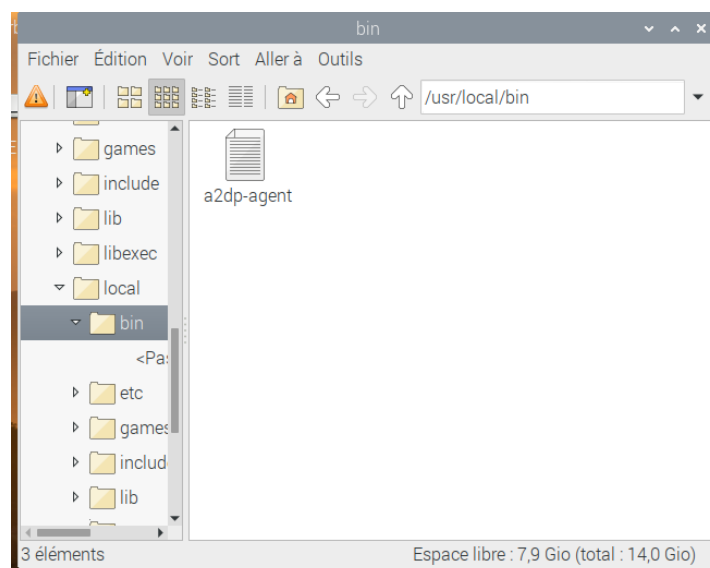
On doit maintenant s'occuper de l'agent Bluetooth. Il s'agit d'un service qui s'occuper de l'appairage et des autorisations pour la connexion Bluetooth entre deux appareils.

Comme on désire faire du streaming audio, il faut s'intéresser à l'agent du protocole A2DP uniquement (service bluetooth pour tout ce qui est audio haute qualité en stéréo).

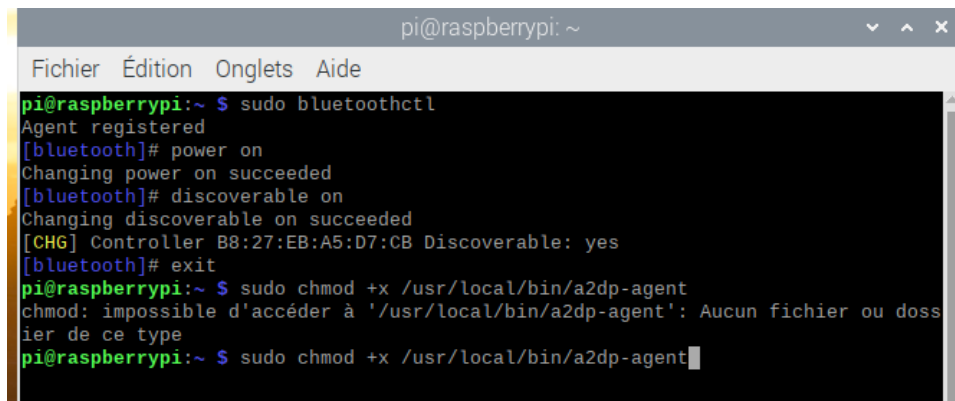
Le fichier fourni '**a2dp-agent**' permettra à la Raspberry de s'appairer correctement avec un smartphone pour une connexion de type 'média' (typiquement comme une enceinte bluetooth). Les autres services seront rejetés (ex : transfert de fichiers, HID,...)

Vous devez copier ce fichier à l'emplacement '**/usr/local/bin**'.

Attention, ce répertoire est un répertoire système, il vous faut donc les droits administrateurs '**sudo**' pour copier ces fichiers. Si vous voulez éviter d'utiliser la ligne de commande, vous pouvez utiliser un explorateur, mais il faut le lancer avec les droits administrateur via un terminal ('**sudo pcmanfm**') :

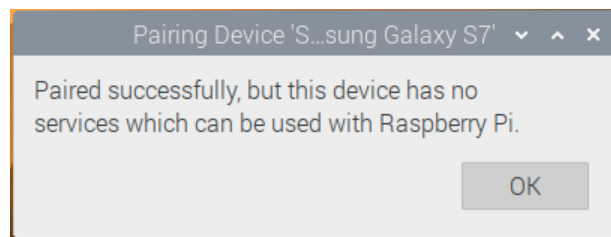


Vous devez ensuite rendre ce fichier exécutable via un terminal :

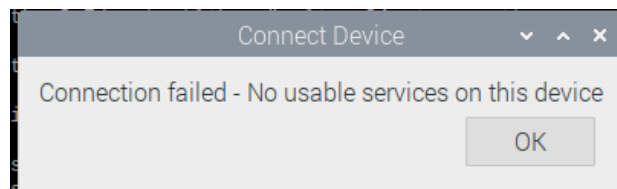


```
pi@raspberrypi: ~  
Fichier Édition Onglets Aide  
pi@raspberrypi:~$ sudo bluetoothctl  
Agent registered  
[bluetooth]# power on  
Changing power on succeeded  
[bluetooth]# discoverable on  
Changing discoverable on succeeded  
[CHG] Controller B8:27:EB:A5:D7:CB Discoverable: yes  
[bluetooth]# exit  
pi@raspberrypi:~$ sudo chmod +x /usr/local/bin/a2dp-agent  
chmod: impossible d'accéder à '/usr/local/bin/a2dp-agent': Aucun fichier ou dossier de ce type  
pi@raspberrypi:~$ sudo chmod +x /usr/local/bin/a2dp-agent
```

A ce stade, vous pourriez vous appairer, mais la connexion serait rejetée car la Raspberry ne serait pas vue comme un périphérique audio (ce que nécessite AzDP). Vous obtiendriez le message suivant :

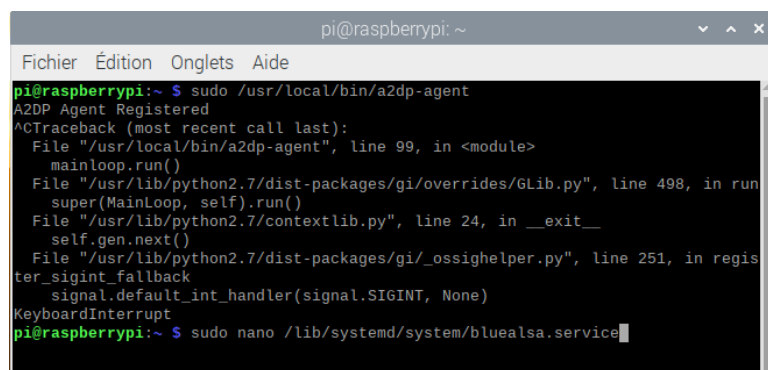


Ou celui-ci :

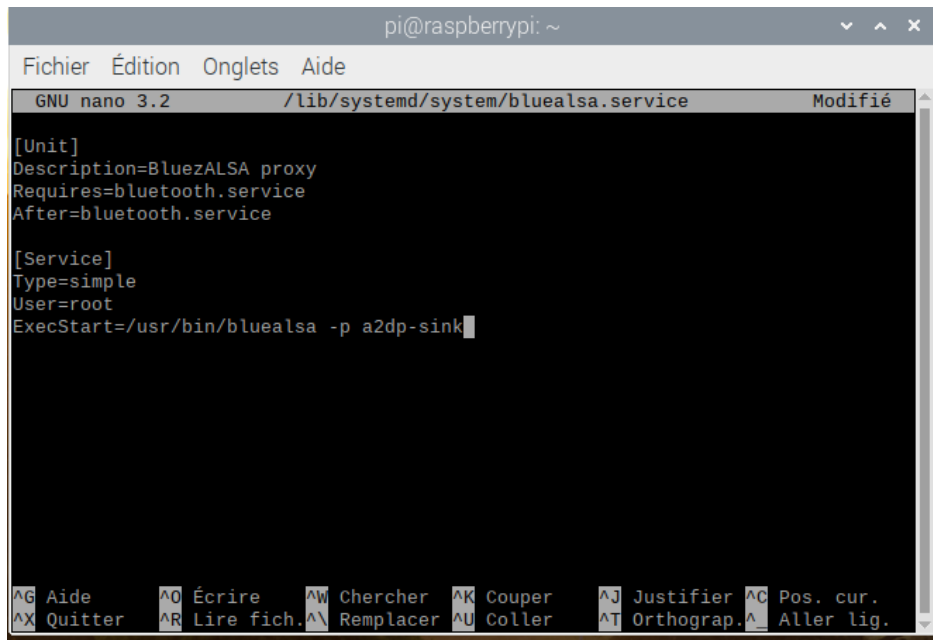


Avant de tester cet agent AzDP Bluetooth pour vérifier que l'on arrive bien à s'appairer, il faut donc que la Raspberry soit bien perçue comme un périphérique audio par le téléphone.

Pour cela, il faut éditer le fichier '/lib/systemd/system/bluealsa.service' et modifier la ligne 'execstart' :



```
pi@raspberrypi: ~  
Fichier Édition Onglets Aide  
pi@raspberrypi:~$ sudo /usr/local/bin/a2dp-agent  
A2DP Agent Registered  
^CTraceback (most recent call last):  
  File "/usr/local/bin/a2dp-agent", line 99, in <module>  
    mainloop.run()  
  File "/usr/lib/python2.7/dist-packages/gi/overrides/GLib.py", line 498, in run  
    super(MainLoop, self).run()  
  File "/usr/lib/python2.7/contextlib.py", line 24, in __exit__  
    self.gen.next()  
  File "/usr/lib/python2.7/dist-packages/gi/_ossighelper.py", line 251, in register_sigint_fallback  
    signal.default_int_handler(signal.SIGINT, None)  
KeyboardInterrupt  
pi@raspberrypi:~$ sudo nano /lib/systemd/system/bluealsa.service
```



```
pi@raspberrypi: ~
Fichier Édition Onglets Aide
GNU nano 3.2 /lib/systemd/system/bluealsa.service Modifié

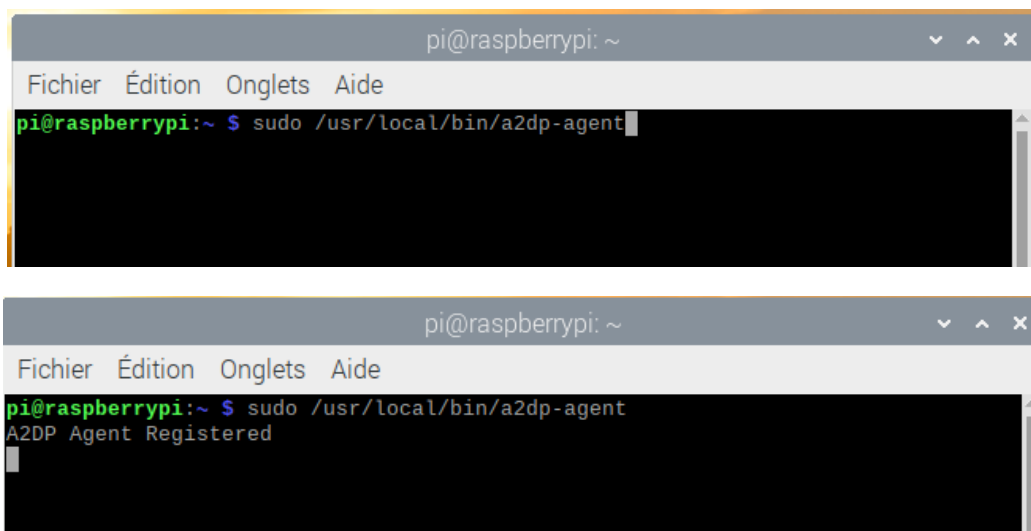
[Unit]
Description=BluezALSA proxy
Requires=bluetooth.service
After=bluetooth.service

[Service]
Type=simple
User=root
ExecStart=/usr/bin/bluealsa -p a2dp-sink

^G Aide ^O Écrire ^W Chercher ^K Couper ^J Justifier ^C Pos. cur.
^X Quitter ^R Lire fich. ^L Remplacer ^U Coller ^T Orthograp. ^_ Aller lig.
```

Normalement, à ce stade la raspberry est visible via le Bluetooth, peut être appairée et reconnue comme un appareil audio.

Pour le tester, il suffit de lancer notre agent Bluetooth A2DP comme ceci :



```
pi@raspberrypi: ~
Fichier Édition Onglets Aide
pi@raspberrypi:~ $ sudo /usr/local/bin/a2dp-agent

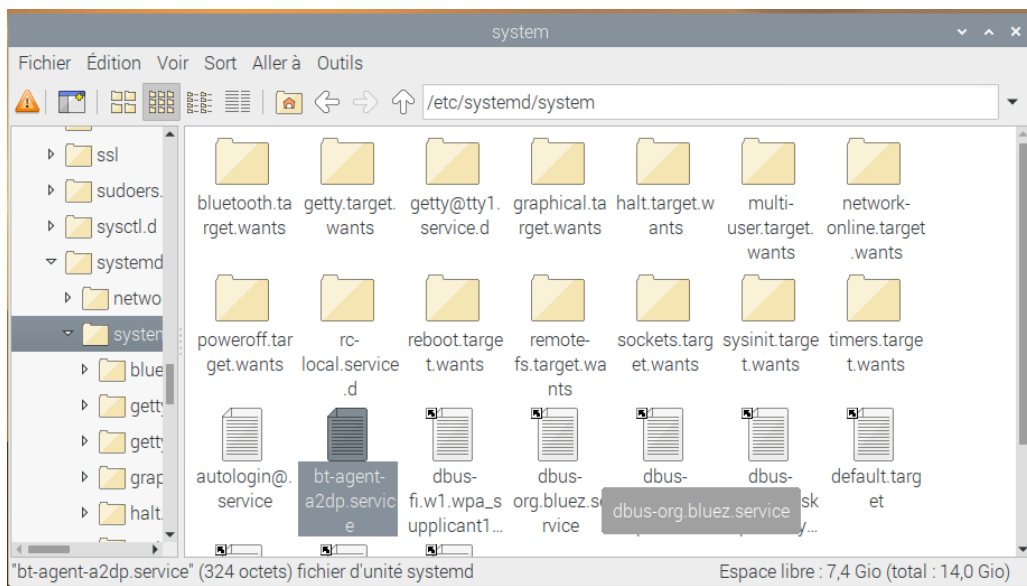
pi@raspberrypi:~ $ sudo /usr/local/bin/a2dp-agent
A2DP Agent Registered
```

Tentez alors via votre téléphone de vous appairer à la Raspberry.

De nouveaux messages doivent apparaître (pas de copie d'écran sous le coude ☹) indiquant l'acceptation du service A2DP et le rejet des services 'non-A2DP'.

Pour que l'on n'ait pas à lancer manuellement cet agent Bluetooth, on va s'arranger pour qu'il démarre à chaque boot de la Raspberry.

Pour installer l'agent Bluetooth en tant que service, il faut copier le fichier '**bt-agent-a2dp.service**' dans le dossier '**/etc/systemd/system**' via l'explorateur (en mode '**sudo**' !!) :



Il faut maintenant lancer ce service via un terminal :

```
pi@raspberrypi: ~
Fichier  Édition  Onglets  Aide

pi@raspberrypi:~$ sudo /usr/local/bin/a2dp-agent
A2DP Agent Registered
^CTraceback (most recent call last):
  File "/usr/local/bin/a2dp-agent", line 99, in <module>
    mainloop.run()
  File "/usr/lib/python2.7/dist-packages/gi/overrides/GLib.py", line 498, in run
    super(MainLoop, self).run()
  File "/usr/lib/python2.7/contextlib.py", line 24, in __exit__
    self.gen.next()
  File "/usr/lib/python2.7/dist-packages/gi/_ossighelper.py", line 251, in register_sigint_fallback
    signal.default_int_handler(signal.SIGINT, None)
KeyboardInterrupt
pi@raspberrypi:~$ sudo nano /lib/systemd/system/bluealsa.service
pi@raspberrypi:~$ sudo systemctl enable bt-agent-a2dp.service
Failed to enable unit: Unit file bt-agent-a2dp.service does not exist.
pi@raspberrypi:~$ sudo systemctl enable bt-agent-a2dp.service
```

Dès lors, le service se lancera à chaque démarrage de la Raspberry et permettra la découverte, l'appairage et la connexion de la Raspberry sans intervention de l'utilisateur du téléphone (sauf bien sûr s'il a désactivé son Bluetooth).

On peut maintenant tester la lecture audio en streaming via la commande suivante :

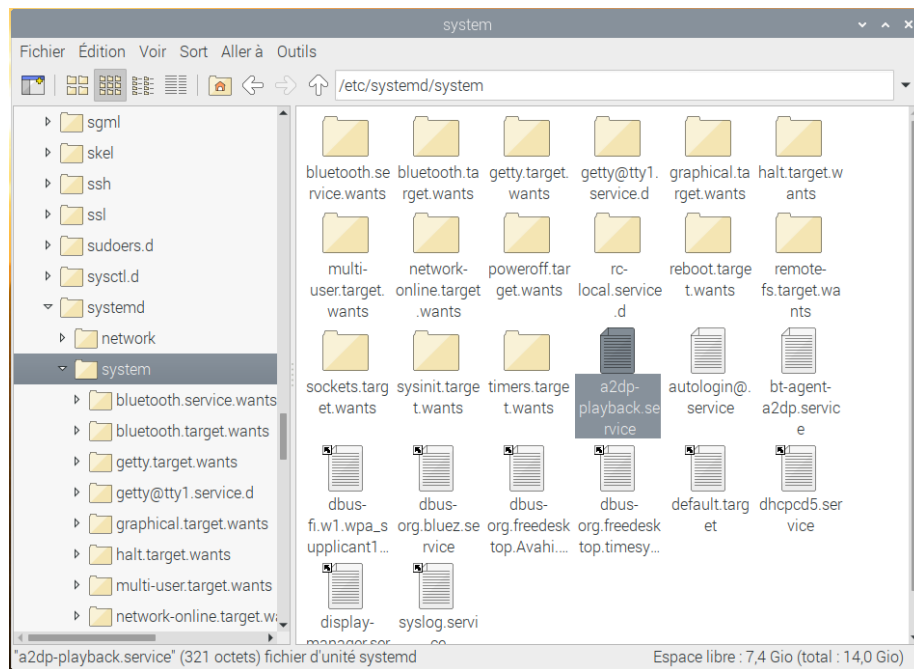
```
pi@raspberrypi: ~
Fichier  Édition  Onglets  Aide

pi@raspberrypi:~$ bluealsa-aptplay -vv 00:00:00:00:00:00
Selected configuration:
BlueALSA service: org.bluealsa
PCM device: default
PCM buffer time: 500000 us
PCM period time: 100000 us
Bluetooth device(s): ANY
Profile: A2DP
```

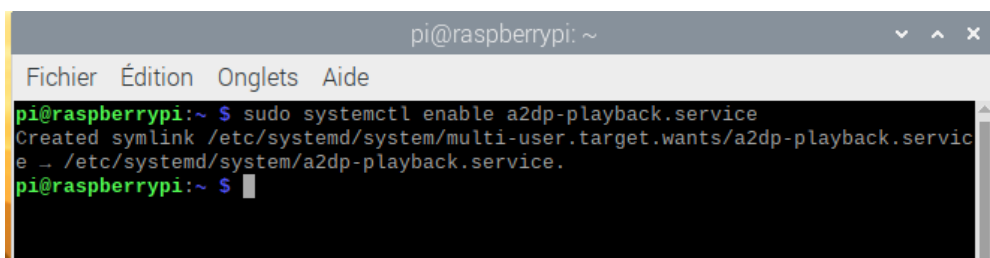
Cette commande transfère juste le flux audio entre la puce Bluetooth de la Raspberry et le driver audio ALSA de la Raspberry (soit notre carte pHat DAC ici).

Il suffit donc de lancer un morceau audio sur le téléphone pour vérifier qu'il est bien entendu dans le haut-parleur branché à la carte pHat DAC.

Il nous reste une dernière étape, c'est faire en sorte que ce streaming audio soit automatique au démarrage. Pour cela, il suffit de copier le fichier '**a2dp-playback.service**' dans le dossier '**/etc/systemd/system**' (toujours via un explorateur en mode '**sudo**') :



Puis d'autoriser ce service au démarrage comme on l'a fait pour l'agent Bluetooth :



C'EST FINI !! Au redémarrage de la Raspberry, tout doit fonctionner correctement !!

L'icône Bluetooth doit clignoter. Vous activez alors le Bluetooth du smartphone.

Si la raspberry n'a jamais été appairée avec ce smartphone, il convient de le faire.

Sinon, vous la sélectionnez dans la liste et après connexion, vous devriez voir « connectée pour son contenu multimédia ». Il ne reste plus qu'à lancer l'appli 'lecteur audio' pour streamer le flux audio vers la Raspberry.