

# Résumé de la solution de Connexion Bluetooth A2DP Persistante

**Objectif :** Établir une connexion Bluetooth Audio (A2DP-Sink) automatique et persistante entre un Orange Pi Zero et une tablette spécifique (MAC : 9C:39:28:2E:FF:F9 ), en contournant les erreurs de profil indisponible ( `br-connection-profile-unavailable` ) et l'absence des binaires BlueAlsa.

**Problème Racine :** L'exécutable `bluealsa` et ses outils ( `bluealsa-aplay` ) étaient absents du système (distribution Orange Pi/Armbian spécifique), nécessitant une compilation manuelle. La compilation a révélé que les binaires sont installés sous les noms `bluealsad` et `bluealsa-aplay` dans le répertoire `/usr/bin` .

## I. Fichiers de Configuration Finaux

Ces trois fichiers représentent l'état stable et fonctionnel de la solution, avec les chemins corrigés ( `/usr/bin/...` ).

### 1. Service BlueAlsa Corrigé ( `/etc/systemd/system/bluealsa.service` )

(Assure le bon démarrage du démon BlueAlsa, en utilisant le nom et le chemin corrects après compilation : `/usr/bin/bluealsad` )

```
[Unit]
Description=BlueALSA Agent
After=bluetooth.service
Wants=bluetooth.service

[Service]
Type=simple
# CHEMIN ET NOM CORRIGÉS du démon BlueAlsa
ExecStart=/usr/bin/bluealsad -p a2dp-source -p a2dp-sink
StandardOutput=journal
StandardError=journal
SyslogIdentifier=BlueALSA-Agent

[Install]
WantedBy=bluetooth.service
```

### 2. Périphérique ALSA Défini ( `/etc/asound.conf` )

(Permet au système ALSA de reconnaître le périphérique virtuel `bluealsa` , résolvant l'erreur `Unknown PCM bluealsa` )

```
# Configuration ALSA pour définir le périphérique 'bluealsa'.
# C'est essentiel pour que 'bluealsa-aplay' fonctionne.
pcm.bluealsa {
    type plug
    slave.pcm {
        type bluealsa
        device "hci0"
        profile "a2dp"
    }
}
ctl.bluealsa {
```

```
    type bluealsa
}
```

### 3. Service de Connexion Automatique A2DP ( /etc/systemd/system/a2dp-playback.service )

(Utilise *bluealsa-aptlay* pour simuler une lecture audio, forçant le profil A2DP à rester disponible et déclenchant la reconnexion)

```
[Unit]
Description=A2DP Playback (Connexion automatique à la tablette)
After=bluealsa.service syslog.service bluetooth_dbus_agent.service
Requires=bluealsa.service
Wants=bluetooth_dbus_agent.service

[Service]
Type=simple
User=root
# Attendre que les services de base soient prêts
ExecStartPre=/bin/sleep 15
# CHEMIN CORRIGÉ de l'outil de lecture
ExecStart=/usr/bin/bluealsa-aptlay --profile-a2dp 9C:39:28:2E:FF:F9
Restart=always
RestartSec=5
StandardOutput=journal
StandardError=journal
SyslogIdentifier=A2DP-Playback-Final

[Install]
WantedBy=multi-user.target
```

## II. Séquence des Commandes Essentielles

Voici la liste ordonnée des commandes pour installer et déployer la solution.

### A. Préparation et Installation des Dépendances

```
# Mise à jour des listes de paquets
sudo apt update

# Installation des dépendances de compilation (Build Essentials)
sudo apt install -y build-essential automake libtool git
sudo apt install -y libasound2-dev libbluetooth-dev libglib2.0-dev
```

### B. Compilation Manuelle de BlueAlsa

```
# Aller dans le répertoire temporaire
cd /tmp

# Cloner le dépôt officiel BlueAlsa
git clone [https://github.com/Arkq/bluez-alsa.git](https://github.com/Arkq/bluez-alsa.git)

# Entrer dans le répertoire source
cd bluez-alsa
```

```
# Préparer les fichiers de construction
autoreconf --install

# Configurer la construction (activer le support A2DP-Sink)
./configure --enable-a2dp-sink

# Compiler le logiciel
make

# Installer le logiciel (installe bluealsad et bluealsa-aplay dans /usr/bin)
sudo make install
```

### C. Déploiement des Fichiers de Configuration (Utilisation de `tee` pour la simplicité)

*(Cette étape est conceptuelle, elle représente la création des fichiers de la section I.)*

```
# Création du fichier de configuration ALSA (/etc/asound.conf)
# Création/Mise à jour du service BlueAlsa (/etc/systemd/system/bluealsa.service)
# Création/Mise à jour du service de connexion A2DP (/etc/systemd/system/a2dp-playback.
```

### D. Activation et Nettoyage Final

*(Cette séquence assure que seuls les services nécessaires tournent et que les anciens services en échec sont désactivés.)*

```
# 1. Désactivation des anciens services en échec ou de reconnexion
sudo systemctl disable dbus_connector.service
sudo systemctl disable bluetooth_reconnect.service
sudo systemctl disable bluetooth-autoconnect.service
sudo systemctl disable bt-agent-a2dp.service
sudo systemctl disable bluetooth_dbus_agent.service
sudo systemctl disable metadata_listener.service
sudo systemctl disable systemd-networkd-wait-online.service

# 2. Arrêt des services pour éviter les conflits au redémarrage
sudo systemctl stop bluetooth-autoconnect.service bt-agent-a2dp.service bluetooth_dbus_

# 3. Rechargement des définitions de service par Systemd
sudo systemctl daemon-reload

# 4. Activation des services définitifs (pour le démarrage au boot)
sudo systemctl enable bluealsa.service
sudo systemctl enable a2dp-playback.service

# 5. Démarrage des services et redémarrage pour tester la séquence complète
sudo systemctl restart bluetooth.service
sudo systemctl restart bluealsa.service
sudo systemctl restart a2dp-playback.service
sudo reboot
```

Le succès de ce processus repose sur l'installation et l'exécution sans erreur de l'exécutable `bluealsad` , qui maintient le profil A2DP nécessaire pour la connexion.

