Nooelec — NESDR SMART

Guide de reconstruction. Il décrit comment faire d'une clé radio logicielle une source permanente pour un serveur de diffusion.

Le fichier `.md` de cette page : ouvrir le `.md`

Ce que cette page couvre : transformer une clé USB de 25 $ en récepteur FM permanent, dont le son est disponible sur tout le réseau domestique. Le montage sert de repli d'alarme : si le flux web que joue le réveil tombe, le lecteur bascule sur cette entrée plutôt que de rester muet.

Contexte : Ubuntu Server 24.04, une clé Nooelec NESDR SMART à puce RTL2832U, `rtl_fm` de `rtl-sdr`, et `ices2` qui pousse le résultat vers un serveur Icecast local. Le serveur lui-même n'est pas décrit ici : il fait l'objet de la page Icecast2 — Ices2, à monter d'abord.

Trois choses valent le détour même sans monter la chaîne. Pourquoi le récepteur affiche une fréquence différente de celle demandée — et pourquoi c'est correct : section *rtl_fm*. Comment déclarer à systemd que quatre services s'excluent physiquement, en n'écrivant la contrainte que dans trois d'entre eux : section *Les quatre stations*. Et pourquoi un pilote de télévision numérique s'empare de la clé au démarrage, alors qu'on ne regarde pas la télévision : section *Écarter les pilotes DVB*.

Convention : tous les blocs de commandes s'exécutent sur la machine où est branchée la clé. Un bloc unique se copie tel quel ; des blocs séparés signifient qu'une édition ou une décision intervient entre eux.

Ce qu'est cette clé, et ce qu'elle sait

faire



Une clé RTL-SDR est à l’origine un récepteur de télévision numérique terrestre. Sa puce, la Realtek RTL2832U, sait livrer à l’ordinateur non pas une image décodée mais l’échantillon brut de ce qu’elle capte — et à partir de ce flux brut, un logiciel peut démoduler à peu près n’importe quoi : radio FM, bandes aériennes, radioamateur, transpondeurs d’avions, télémétrie de stations météo.

C’est ce qu’on appelle une radio logicielle : le matériel se contente de descendre un morceau de spectre en bande de base, tout le reste est du calcul.

Le modèle utilisé ici est une Nooelec NESDR SMART — boîtier d'aluminium, oscillateur à compensation de température, connecteur SMA. L'oscillateur compensé est ce qui la distingue des clés à quelques dollars : sa fréquence ne dérive pas quand elle chauffe, ce qui compte pour un service qui tourne en continu.

Ce guide n'en exploite qu'une fraction : une seule bande, la FM commerciale, et un seul usage, la diffusion continue d'une station.

Le matériel

```
lsusb | grep -i realtek
```

Attendu :

```
Bus 001 Device 003: ID 0bda:2832 Realtek RTL2832U DVB-T
```

Le couple `0bda:2832` est l'identifiant qui compte. C'est lui qu'on écrit dans la règle d'accès plus bas, et c'est lui que le système présente — quelle que soit la marque imprimée sur le boîtier, la puce est la même chez tous les fabricants.

L'antenne fournie suffit pour la FM commerciale, dont les émetteurs sont puissants et proches. Sa position, en revanche, décide de tout : une station peut être parfaite et sa voisine inécoutable selon l'endroit où repose l'antenne. C'est à mesurer avant d'incriminer le logiciel.

Installer

```
sudo apt install rtl-sdr
```

Le paquet apporte `rtl_fm` — la démodulation qui nous intéresse — ainsi que `rtl_test` et `rtl_power`, utiles au diagnostic.

`ices2` et le serveur Icecast sont installés séparément : voir la page Icecast2 — Ices2.

Écarter les pilotes DVB

C’est l’étape qu’on oublie, et elle est la cause du plus grand nombre de démarrages ratés.

Le problème. Le noyau reconnaît la puce comme ce qu’elle est officiellement — un tuner de télévision numérique — et lui attache ses pilotes : `dvb_usb_rtl28xxu`, `rtl2832_sdr`. Ils s’emparent du périphérique au branchement. Quand `rtl_fm` démarre ensuite, la bibliothèque `librtlsdr` les détache de force, et le journal l’annonce :

```
Detached kernel driver
```

⚠ Ça fonctionne, et c’est précisément le piège. Le message ressemble à une information anodine, la radio joue, on passe à autre chose. Mais c’est une course entre le pilote du noyau et le programme : elle est gagnée presque toujours, ce qui veut dire pas toujours.

Le correctif — `/etc/modprobe.d/blacklist-rtl-sdr.conf` :

```
# Le dongle RTL2832U sert au SDR (rtl_fm), jamais à la télévision numérique.
# Sans ce blocage, le pilote DVB s'attache au démarrage et librtlsdr doit
# le détacher à chaque lancement — une course inutile.
blacklist dvb_usb_rtl28xxu
blacklist rtl2832_sdr
```

Le blocage ne prend effet qu’au redémarrage suivant, puisqu’il s’agit d’empêcher un chargement automatique.

Vérification, après redémarrage :

```
lsmod | grep -c dvb
```

Attendu : . Et le message `Detached kernel driver` doit avoir disparu du journal du service, tandis que `Tuned to …` et `Connected to server` demeurent.

⚠ Cette précaution peut figurer dans la documentation sans exister sur la machine. Elle était décrite dans la première version de cet article, en 2021, sous un autre nom de fichier — et ce fichier n’avait jamais été créé. L’écart est resté invisible quatre ans, parce que la course était gagnée ; il n’est apparu qu’à un changement de noyau qui a ramené les modules DVB. Vérifier plutôt que croire : le fichier existe-t-il vraiment ?

```
ls -l /etc/modprobe.d/ | grep -i -e dvb -e rtl
```

L’accès à la clé — la règle udev

Sans règle particulière, le périphérique USB n’appartient qu’à `root` : le service devrait tourner en administrateur, ce qu’on préfère éviter pour un programme qui reçoit des données du dehors.

`/etc/udev/rules.d/20.rtlsdr.rules` :

```
SUBSYSTEM=="usb", ATTRS{idVendor}=="0bda", ATTRS{idProduct}=="2832",
GROUP="plugdev", MODE="0660", SYMLINK+="rtl_sdr"
```

Trois parties, à lire séparément :

- `ATTRS{idVendor}` et `ATTRS{idProduct}` — la règle ne vise que cette puce, et s'applique quel que soit le port où l'on branche la clé.

- `GROUP="plugdev"` et `MODE="0660"` — le périphérique appartient au groupe `plugdev`, en lecture et écriture pour ce groupe seulement. Le compte du service doit en être membre.

- `SYMLINK+="rtl_sdr"` — crée `/dev/rtl_sdr`, un nom stable qui ne dépend pas du numéro de bus ni de l'ordre d'énumération.

⚠ La version qui circule dans la plupart des tutoriels — et dans la première version de cet article — porte `GROUP="adm", MODE="0666"`. C'est-à-dire lecture et écriture pour tout processus de la machine, y compris ceux qui n'ont rien à y faire. `0660` avec un groupe dédié donne le même service en fermant cette porte.

Appliquer sans débrancher :

```
sudo udevadm control --reload-rules && sudo udevadm trigger --subsystem-match=usb
```

```
ls -lL /dev/rtl_sdr
```

Attendu : `crw-rw---- root plugdev`.

Le `-L` n'est pas décoratif. `/dev/rtl_sdr` est un lien vers `/dev/bus/usb/001/003` : les permissions vivent sur la cible, et un `ls -l` sans `-L` montre celles du lien, qui sont toujours `lrwxrwxrwx` et n'apprennent rien.

Vérifier l'appartenance au groupe :

```
id hostadmin | tr ',' '\n' | grep plugdev
```

Si le compte n’y est pas :

```
sudo usermod -aG plugdev hostadmin
```

⚠ Le vrai test reste le redémarrage du service, qui rouvre le périphérique sous la nouvelle identité. Une règle rechargée ne change rien à un processus déjà en cours.

`rtl_fm` — de l’onde au son

La démodulation tient en une commande :

```
rtl_fm -M fm -s 200k -o 4 -A fast -r 48k -l 0 -E deemp -f 98.5M
```

Elle écrit du PCM brut sur la sortie standard : pas d’en-tête, pas de format de fichier, juste des échantillons. C’est ce qui permet de l’enchaîner directement à `ices2`.

Les paramètres, un par un :

Paramètre	Rôle
`-M fm`	Démodulation en modulation de fréquence
`-s 200k`	Largeur de bande échantillonnée — 200 kHz, la largeur d’un canal FM commercial
`-r 48k`	Fréquence de sortie du son, 48 kHz
`-o 4`	Suréchantillonnage d’un facteur 4, qui améliore le rapport signal/bruit
`-A fast`	Calcul rapide de l’arc tangente — allège le processeur, différence inaudible
`-E deemp`	Désaccentuation. Indispensable en FM commerciale : l’émetteur relève les aigus avant émission, le récepteur doit les rabaisser. Sans ce paramètre, le son est criard.

Paramètre	Rôle
`-l 0`	Seuil de silencieux désactivé : diffuser en continu, même le souffle
`-f 98.5M`	La fréquence de la station

⚠ `-l 0` est un choix, pas un défaut. Un silencieux couperait le son entre les émissions faibles — pour une diffusion permanente qui sert de repli, un souffle audible vaut mieux qu'un silence qu'on prendrait pour une panne.

`Tuned to 98900000 Hz` pour une demande à 98,5 n'est pas une erreur, et le journal contient de quoi le vérifier soi-même. `rtl_fm` décale volontairement le récepteur d'un quart de sa fréquence d'échantillonnage, pour éloigner la station du pic parasite qui siège au centre de la bande de tout récepteur à conversion directe, puis ramène le signal par calcul :

```
Sampling at 1600000 S/s.      →  1 600 000 / 4 = 400 000 Hz
Tuned to 98900000 Hz.         →  98,5 MHz + 400 kHz
```

La fréquence affichée est celle du matériel ; celle qu'on écoute est bien la bonne.

Essayer avant d'automatiser :

```
rtl_fm -M fm -s 200k -o 4 -A fast -r 48k -l 0 -E deemp -f 98.5M | aplay -r 48k -f S16_LE -t raw -c 1
```

Sur une machine sans carte son, remplacer `aplay` par un enregistrement borné dans le temps :

```
timeout 5 rtl_fm -M fm -s 200k -o 4 -A fast -r 48k -l 0 -E deemp -f 98.5M > /tmp/essai.raw ; ls -l /tmp/essai.raw
```

Attendu : de l'ordre de 480 ko pour cinq secondes à 48 kHz sur 16 bits en mono.

Le tuyau vers Icecast

`rtl_fm` produit du PCM ; `ices2` sait le lire par son entrée standard grâce au module `stdinpcm`. Les deux se joignent par un tuyau ordinaire :

```
rtl_fm … -f 98.5M | ices2 /etc/ices2_98_5_config.xml
```

Le fichier de configuration suit exactement la charpente décrite dans la page Icecast2 — Ices2 ; seuls diffèrent le bloc d’entrée et les métadonnées.

`/etc/ices2_98_5_config.xml` — commentaires retirés, mot de passe remplacé par `MASQUE` :

```
<?xml version="1.0"?>
<ices>
    <background>0</background>
    <loglevel>3</loglevel>
    <consolelog>1</consolelog>
    <pidfile>/home/hostadmin/ices2/ices_98_5.pid</pidfile>

    <stream>
        <metadata>
            <name>Diffusion Radio FM - 98.5 FM</name>
            <genre>FM</genre>
            <artist>Radio FM - Région de Montréal</artist>
            <description>Diffusion en streaming du FM local</description>
        </metadata>

        <input>
          <module>stdinpcm</module>
          <param name="rate">48000</param>
          <param name="channels">1</param>
          <param name="metadata">1</param>
          <param
name="metadatafilename">/home/hostadmin/ices2/metadata_98_5</param>
        </input>
```

```
        <instance>
            <hostname>localhost</hostname>
            <port>8000</port>
            <password>MASQUE</password>
            <mount>/Radio_FM.ogg</mount>
            <reconnectdelay>2</reconnectdelay>
            <reconnectattempts>5</reconnectattempts>
            <maxqueuelength>80</maxqueuelength>

            <encode>
                <nominal-bitrate>192000</nominal-bitrate>
                <samplerate>48000</samplerate>
                <channels>1</channels>
            </encode>
        </instance>
    </stream>
</ices>
```

Les différences avec une source de carte son :

- `<module>stdinpcm</module>` au lieu d'`alsa` — le son arrive par le tuyau, il n'y a aucun périphérique à ouvrir. Il n'y a pas non plus de `device` à nommer : l'entrée standard n'a pas d'adresse.

- 48 000 Hz, un seul canal. La FM démodulée par `rtl_fm` est monophonique — il ne décode pas la sous-porteuse stéréo. Déclarer deux canaux ne créerait pas de stéréo, seulement le double du débit pour le même son.

- Les fréquences d'entrée et d'encodage sont identiques, 48 000 des deux côtés, exactement le `-r 48k` de `rtl_fm`. ⚠ Une discordance entre ces trois valeurs donne un son trop lent ou trop rapide, sans le moindre message d'erreur — `ices2` ne sait pas rééchantillonner.

Le `<mount>` est le même pour les quatre stations — `/Radio_FM.ogg`, le poste de

radio dont on change la station. Et la fréquence n’est pas dans ce fichier du tout : elle vit dans la ligne `ExecStart` de l’unité systemd, au paramètre `-f` de `rtl_fm`.

⚠ Ce fichier porte le mot de passe source d’Icecast en clair. Il doit être lisible par le compte du service, et par personne d’autre :

```
sudo chown root:hostadmin /etc/ices2_*.xml
```

```
sudo chmod 640 /etc/ices2_*.xml
```

Les trois autres stations

Elles ne diffèrent que par trois lignes — le `<pidfile>`, le `<name>` des métadonnées et le `metadatafilename`. Les voici en entier, pour qu’il n’y ait rien à transposer :

`/etc/ices2_95_1_config.xml`

```
<?xml version="1.0"?>
<ices>
    <background>0</background>
    <loglevel>3</loglevel>
    <consolelog>1</consolelog>
    <pidfile>/home/hostadmin/ices2/ices_95_1.pid</pidfile>

    <stream>
        <metadata>
            <name>Diffusion Radio FM - ICI Première 95.1</name>
            <genre>FM</genre>
            <artist>Radio FM - Région de Montréal</artist>
            <description>Diffusion en streaming du FM local</description>
        </metadata>

        <input>
          <module>stdinpcm</module>
          <param name="rate">48000</param>
```

```
          <param name="channels">1</param>
          <param name="metadata">1</param>
          <param
name="metadatafilename">/home/hostadmin/ices2/metadata_95_1</param>
        </input>

        <instance>
            <hostname>localhost</hostname>
            <port>8000</port>
            <password>MASQUE</password>
            <mount>/Radio_FM.ogg</mount>
            <reconnectdelay>2</reconnectdelay>
            <reconnectattempts>5</reconnectattempts>
            <maxqueuelength>80</maxqueuelength>

            <encode>
                <nominal-bitrate>192000</nominal-bitrate>
                <samplerate>48000</samplerate>
                <channels>1</channels>
            </encode>
        </instance>
    </stream>
</ices>
```

/etc/ices2_96_9_config.xml

```
<?xml version="1.0"?>
<ices>
    <background>0</background>
    <loglevel>3</loglevel>
    <consolelog>1</consolelog>
    <pidfile>/home/hostadmin/ices2/ices_96_9.pid</pidfile>

    <stream>
        <metadata>
            <name>Diffusion Radio FM - CKOI 96.9</name>
            <genre>FM</genre>
            <artist>Radio FM - Région de Montréal</artist>
            <description>Diffusion en streaming du FM local</description>
        </metadata>
```

```
        <input>
          <module>stdinpcm</module>
          <param name="rate">48000</param>
          <param name="channels">1</param>
          <param name="metadata">1</param>
          <param
name="metadatafilename">/home/hostadmin/ices2/metadata_96_9</param>
        </input>

        <instance>
            <hostname>localhost</hostname>
            <port>8000</port>
            <password>MASQUE</password>
            <mount>/Radio_FM.ogg</mount>
            <reconnectdelay>2</reconnectdelay>
            <reconnectattempts>5</reconnectattempts>
            <maxqueuelength>80</maxqueuelength>

            <encode>
                <nominal-bitrate>192000</nominal-bitrate>
                <samplerate>48000</samplerate>
                <channels>1</channels>
            </encode>
        </instance>
    </stream>
</ices>
```

/etc/ices2_100_7_config.xml

```
<?xml version="1.0"?>
<ices>
    <background>0</background>
    <loglevel>3</loglevel>
    <consolelog>1</consolelog>
    <pidfile>/home/hostadmin/ices2/ices_100_7.pid</pidfile>

    <stream>
        <metadata>
            <name>Diffusion Radio FM - ICI Musique 100.7</name>
            <genre>FM</genre>
            <artist>Radio FM - Région de Montréal</artist>
```

```
            <description>Diffusion en streaming du FM local</description>
        </metadata>

        <input>
          <module>stdinpcm</module>
          <param name="rate">48000</param>
          <param name="channels">1</param>
          <param name="metadata">1</param>
          <param
name="metadatafilename">/home/hostadmin/ices2/metadata_100_7</param>
        </input>

        <instance>
            <hostname>localhost</hostname>
            <port>8000</port>
            <password>MASQUE</password>
            <mount>/Radio_FM.ogg</mount>
            <reconnectdelay>2</reconnectdelay>
            <reconnectattempts>5</reconnectattempts>
            <maxqueuelength>80</maxqueuelength>

            <encode>
                <nominal-bitrate>192000</nominal-bitrate>
                <samplerate>48000</samplerate>
                <channels>1</channels>
            </encode>
        </instance>
    </stream>
</ices>
```

Les quatre stations — un seul récepteur

Quatre stations écoutées, donc quatre configurations et quatre services : `rtl-fm-95-1`, `rtl-fm-96-9`, `rtl-fm-98-5`, `rtl-fm-100-7`.

Toutes poussent vers le même point de montage, `/Radio_FM.ogg`. C’est le choix structurant, et il est bon : le montage est un *poste de radio* dont on change la

station. Le serveur de musique et l’entrée d’alarme pointent une seule URL, qu’on n’a jamais à modifier.

L’unité systemd

`/etc/systemd/system/rtl-fm-98-5.service` :

```
[Unit]
Description=Radio FM 98,5 vers Icecast
Requires=icecast2.service
After=icecast2.service
Conflicts=rtl-fm-95-1.service rtl-fm-96-9.service rtl-fm-100-7.service
StartLimitIntervalSec=0

[Service]
Type=simple
User=hostadmin
Group=hostadmin
ExecStart=/bin/bash -c '/usr/bin/rtl_fm -M fm -s 200k -o 4 -A fast -r 48k -l 0 -E deemp -f 98.5M | /usr/bin/ices2 /etc/ices2_98_5_config.xml'
Restart=always
RestartSec=10

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

⚠ `ExecStart` passe par `/bin/bash -c` parce qu’il y a un tuyau. systemd exécute directement, sans shell : un `|` dans une ligne `ExecStart` nue serait passé comme argument littéral à `rtl_fm`. C’est la différence avec la source de carte son, qui n’a pas besoin de shell.

⚠ La ligne `ExecStart` est laissée entière à dessein. systemd accepte une continuation par barre oblique inverse, mais la manière dont il traite l’indentation de la ligne suivante n’a pas été vérifiée ici — et cette valeur contient une commande entre apostrophes, où une espace de trop se retrouverait dans le tuyau. La ligne est donc longue et se replie dans le PDF sans signe visible : la recopier depuis la source du document, jamais depuis le PDF.

`Requires=` et `After=icecast2.service` — les deux. Voir la page Icecast2 — Ices2 : sur un échec de première connexion, `ices2` abandonne sans réessayer, et les `reconnectattempts` du XML n'y changent rien. L'ordre de démarrage est la seule protection.

L'exclusion mutuelle

⚠ Un seul récepteur — les quatre stations s'excluent physiquement. Deux instances ne peuvent pas coexister : `rtl_fm` échouerait sur un périphérique occupé, et Icecast refuserait de toute façon une seconde source sur le même montage. Avec `Restart=always`, l'unité perdante boucherait le journal indéfiniment.

La contrainte s'écrit :

```
Conflicts=rtl-fm-95-1.service rtl-fm-96-9.service rtl-fm-100-7.service
```

Démarrer une station arrête automatiquement celle qui tourne. `sudo systemctl start rtl-fm-98-5` suffit, sans rien arrêter au préalable.

 La relation est symétrique dans systemd, et c'est un piège de lecture. Sur cette machine, une seule des quatre unités ne porte pas de ligne `Conflicts=` — ce qui a tout l'air d'un oubli. Ce n'en est pas un : déclarer le conflit dans les trois autres suffit, la réciproque est dérivée. Plutôt que de le supposer, on le demande au système :

```
systemctl show -p Conflicts -p ConflictedBy rtl-fm-95-1.service
```

La sortie montre `ConflictedBy=` peuplé des trois autres unités. La contrainte est donc complète.

La station de démarrage

⚠ Une seule unité porte la section `[Install]` à la fois, volontairement. Sans elle, `systemctl enable` refuse — ce qui empêche *structurellement* d'activer deux

stations au démarrage, qui se disputeraient le récepteur avec une issue imprévisible. La contrainte physique est ainsi inscrite dans les fichiers, pas seulement dans la mémoire de l’administrateur.

⚠ Déplacer la section `[Install]` ne suffit pas à changer de station par défaut. Ce qui décide du démarrage automatique est un lien symbolique dans `/etc/systemd/system/multi-user.target.wants/`, créé par `systemctl enable` — pas la section elle-même, et `daemon-reload` n’y touche pas. Trois gestes :

```
sudo systemctl disable rtl-fm-95-1
```

```
sudo systemctl enable rtl-fm-98-5
```

```
systemctl is-enabled rtl-fm-95-1 rtl-fm-98-5
```

Attendu : `static` puis `enabled`. Une unité `static` reste lançable à la main ; elle ne démarrera simplement pas toute seule.

La station de démarrage est celle du repli d’alarme, et le critère de choix est la qualité de réception à l’endroit où repose l’antenne — pas la préférence d’écoute.

⚠ Conséquence du montage partagé : une station laissée en marche le soir devient celle que jouera l’alarme du lendemain. Le réveil ne sera pas muet, mais pas sur la station attendue.

Une différence de paramètres à préserver

Une des quatre unités porte `-g 25` — gain manuel de 25 dB — là où les trois autres laissent le gain automatique. Ce n’est vraisemblablement pas un choix esthétique mais une compensation de réception faible : c’est justement la station dont le signal est mauvais. Ne pas uniformiser par souci de propreté, et ne pas s’étonner que la qualité y reste inférieure — le gain amplifie le bruit avec le signal, il ne crée pas de signal.

Les trois autres unités

Elles reprennent la même charpente, avec quatre écarts : la description, la fréquence de `rtl_fm`, le fichier de configuration `ices2`, et la liste `Conflicts=` qui nomme les trois autres.

⚠ Aucune des trois ne porte de section `[Install]` — c'est ce qui les rend `static` : lançables à la main, jamais au démarrage. La section vit dans la seule unité choisie comme station de démarrage, et se déplace avec elle.

`/etc/systemd/system/rtl-fm-95-1.service`

```
[Unit]
Description=Radio FM 95,1 vers Icecast
Requires=icecast2.service
After=icecast2.service
StartLimitIntervalSec=0

[Service]
Type=simple
User=hostadmin
Group=hostadmin
ExecStart=/bin/bash -c '/usr/bin/rtl_fm -M fm -s 200k -g 25 -o 4 -A fast -r 48k -
l 0 -E deemp -f 95.1M | /usr/bin/ices2 /etc/ices2_95_1_config.xml'
Restart=always
RestartSec=10
```

`/etc/systemd/system/rtl-fm-96-9.service`

```
[Unit]
Description=Radio FM 96,9 vers Icecast
Requires=icecast2.service
After=icecast2.service
Conflicts=rtl-fm-95-1.service rtl-fm-98-5.service rtl-fm-100-7.service
StartLimitIntervalSec=0

[Service]
Type=simple
User=hostadmin
Group=hostadmin
```

```
ExecStart=/bin/bash -c '/usr/bin/rtl_fm -M fm -s 200k -o 4 -A fast -r 48k -l 0 -E
deemp -f 96.9M | /usr/bin/ices2 /etc/ices2_96_9_config.xml'
Restart=always
RestartSec=10
```

`/etc/systemd/system/rtl-fm-100-7.service`

```
[Unit]
Description=Radio FM 100,7 vers Icecast
Requires=icecast2.service
After=icecast2.service
Conflicts=rtl-fm-95-1.service rtl-fm-96-9.service rtl-fm-98-5.service
StartLimitIntervalSec=0

[Service]
Type=simple
User=hostadmin
Group=hostadmin
ExecStart=/bin/bash -c '/usr/bin/rtl_fm -M fm -s 200k -o 4 -A fast -r 48k -l 0 -E
deemp -f 100.7M | /usr/bin/ices2 /etc/ices2_100_7_config.xml'
Restart=always
RestartSec=10
```

□□ La première ne porte pas de ligne `Conflicts=` non plus, et c’est correct : les trois autres la nomment, et systemd dérive la réciproque. La vérification est celle donnée plus haut, `systemctl show -p ConflictedBy`.

Mise en service

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable --now rtl-fm-98-5
```

```
systemctl status rtl-fm-98-5 --no-pager
```

Attendu dans le journal : `Tuned to …`, puis `Connected to server:`

`localhost:8000/Radio_FM.ogg`.

Ce que ce montage remplace

Les premières versions de cette installation lançaient la chaîne par des scripts en tâches cron, à heure fixe, avec un fichier de PID pour arrêter l'instance de la veille. Le passage à systemd règle d'un coup quatre défauts, et il vaut d'énumérer lesquels — ce sont ceux de tout script de service lancé par cron :

Défaut	Ce que systemd apporte
Aucun moyen d'exprimer « après Icecast »	`Requires=` et `After=`
Fichier de PID à écrire, relire, nettoyer	systemd suit ses propres processus
Relance une fois par jour	`Restart=always`, dix secondes plus tard
Sortie déversée dans un fichier qui enfle	`journalctl`, avec sa rotation

⚠ Le dernier défaut coûtait le plus cher sans être visible : les deux scripts de diffusion écrivaient dans le même fichier que les rapports de sauvegarde — jusqu'à 5,6 Mo pour une seule journée, noyant les rapports qu'on venait d'y mettre.

 Et un défaut plus insidieux encore, propre aux scripts : celui d'origine annonçait sa réussite en testant `$?` après un `sleep` plutôt qu'après les commandes qui comptaient. Il rapportait donc systématiquement un succès. Un service qui ment sur son état est pire qu'un service en panne.

Vérifier et dépanner

Le journal du service

Il n’y a pas de fichier de journal à chercher : `rtl_fm` et `ices2` tournent dans le même service, leurs deux sorties vont à `journalctl`, mêlées mais datées.

```
journalctl -u rtl-fm-98-5 -n 30 --no-pager
```

```
journalctl -u rtl-fm-98-5 -f
```

Un démarrage sain se lit dans cet ordre :

```
bash[386829]: Tuned to 98900000 Hz.
bash[386829]: Oversampling input by: 2x.
bash[386829]: Oversampling output by: 4x.
bash[386829]: Buffer size: 5.12ms
bash[386829]: Allocating 15 zero-copy buffers
bash[386829]: Sampling at 1600000 S/s.
bash[386829]: Output at 200000 Hz.
bash[386830]: INFO encode/encode_initialise Encoder initialising in VBR mode: 1
channels, 48000 Hz, nominal 192000
bash[386830]: INFO stream/ices_instance_stream Connected to server:
localhost:8000/Radio_FM.ogg
```

□□ Les deux numéros entre crochets sont les deux bouts du tuyau — ici `386829` pour `rtl_fm` et `386830` pour `ices2`. C’est ce qui permet de lire un journal où deux programmes parlent en même temps, et de voir lequel s’est tu.

Ce journal contient l’explication du décalage de fréquence, pour qui rapproche trois de ses lignes :

- `Sampling at 1600000 S/s` — le récepteur échantillonne à 1,6 MHz.
- Le quart de cette valeur fait 400 kHz.

- `Tuned to 98900000 Hz` — soit 98,5 MHz demandés plus ces 400 kHz.

C'est le décalage volontaire décrit plus haut. Le matériel est réglé 400 kHz à côté ; le logiciel ramène la station au centre. Rien à corriger.

Chacune de ces lignes peut être la dernière, et l'endroit où ça s'arrête désigne le coupable.

La dernière ligne est…	Le problème est…
aucune, ou une erreur d'ouverture USB	l'accès au périphérique — règle udev, groupe `plugdev`, ou clé absente
`Output at 200000 Hz` et rien d'`ices2`	`ices2` — fichier de configuration illisible, mal formé, ou droits refusés
`Encoder initialising`	la connexion au serveur n'aboutit pas
`Failed initial connect`	Icecast n'était pas démarré : voir la page Icecast2 — Ices2

⚠ Vérifier que les valeurs de l'encodeur sont celles attendues : `1 channels, 48000 Hz` doit correspondre au `-r 48k` de `rtl_fm` et au `<channels>1</channels>` du XML. Une discordance ne provoque aucune erreur — elle donne un son trop lent ou trop rapide, et cette ligne est le seul endroit où elle se voit.

✓ `Detached kernel driver` ne doit plus apparaître une fois la blacklist DVB en place. S'il revient, c'est que le fichier de `/etc/modprobe.d/` a disparu ou que la machine n'a pas redémarré depuis sa création.

Le détail des autres commandes de `journalctl` — sélection par date, erreurs seules, plusieurs services d'un coup — est sur la page Icecast2 — Ices2, section *Lire les journaux*.

Autres vérifications

Le flux est-il servi ? Voir la page Icecast2 — Ices2, section de vérification. En résumé :

```
curl -s --max-time 4 -o /tmp/essai.ogg http://192.168.0.11:8000/Radio_FM.ogg ; ls
```

```
-l /tmp/essai.ogg
```

Attendu : de l’ordre de 100 à 200 kio.

Le récepteur répond-il ? Arrêter le service, qui détient le périphérique, puis :

```
sudo systemctl stop rtl-fm-98-5
```

```
rtl_test -t
```

```
sudo systemctl start rtl-fm-98-5
```

`rtl_test` énumère le matériel et signale les échantillons perdus.

Une station donnée passe-t-elle mal ? Mesurer avant de régler :

```
rtl_power -f 88M:108M:100k -g 30 -i 10 -e 10 /tmp/bande-fm.csv
```

Le fichier donne la puissance reçue sur toute la bande FM. Une station faible s’y voit — et déplacer l’antenne se juge alors sur des chiffres plutôt qu’à l’oreille.

Aucun son alors que tout paraît sain ? Le diagnostic complet est sur la page Icecast2 — Ices2 — il commence par couper le problème en deux à Icecast, ce qui vaut pour les deux sources.