Contexte

J’utilisais depuis quelque temps un PC Windows pour diffuser les chaînes Free TV sur un écran passif via HDMI. Le système fonctionnait correctement mais présentait des inconvénients notables : consommation électrique élevée (150W), encombrement, bruit du ventilateur, et démarrage manuel nécessaire à chaque utilisation.

J’ai décidé de migrer cette installation vers un Raspberry Pi 5, avec pour objectif de créer une box TV dédiée, silencieuse, économe en énergie et contrôlable entièrement depuis un smartphone.

Architecture finale

Le système repose sur une architecture simple :

- Matériel : Raspberry Pi 5 (8 GB RAM), écran Dell P2210 en HDMI, Google Nest Mini pour l’audio
- Système : Raspberry Pi OS avec environnement graphique X11
- Backend : Serveur Node.js avec Puppeteer pilotant Chromium
- Frontend : Interface web responsive accessible depuis smartphone
- Audio : Diffusion via Google Cast (automatisée par xdotool)

Le Pi est connecté en HDMI à l’écran pour la vidéo, et l’audio est envoyé vers le Nest Mini via Cast. L’ensemble est contrôlé depuis un smartphone via une interface web hébergée par le serveur Node.js.

Fonctionnalités

Contrôle depuis smartphone

Une interface web permet de :

- Sélectionner une chaîne parmi les chaînes Free TV organisées par catégories
- Ajuster le volume
- Déclencher le Cast audio automatiquement vers le Nest Mini
- Arrêter la lecture

Automatisation complète

- Démarrage automatique du serveur au boot du Pi (service systemd)
- Réveil automatique de l'écran en cas de mise en veille
- Masquage automatique du curseur après 1 seconde d'inactivité
- Reconnexion automatique en cas de déconnexion Puppeteer

Accès distant

Configuration d'un serveur xrdp permettant l'accès graphique depuis Windows via le client RDP natif, pour les opérations de maintenance.

Défis techniques rencontrés

Navigation directe vers les streams

Contrairement à l'approche initiale sous Windows qui utilisait une page

intermédiaire, la solution Pi navigue directement vers les URLs des streams Free TV. Cela évite les problèmes de Cast API et permet d'utiliser « Cast tab audio » au lieu du « Default Media Receiver ».

Automatisation du Cast audio

L'absence de raccourci clavier direct pour le Cast dans Chromium a nécessité l'automatisation d'une séquence complète via xdotool :

1. Sortie du mode plein écran (F11)
2. Ouverture du menu Chrome (Alt+F)
3. Navigation dans les menus (19 touches Down)
4. Sélection dans les sous-menus
5. Interaction avec la popup Cast (Tab et Enter)
6. Retour en plein écran

Cette séquence, déclenchée depuis le smartphone, s'exécute en quelques secondes et établit la connexion Cast vers le Nest Mini.

Passage de Wayland à X11

Le Pi 5 utilise par défaut Wayland, incompatible avec xdotool. Le passage à X11 via raspi-config a été nécessaire pour permettre l'automatisation par simulation de frappes clavier.

Service systemd

La configuration du service de démarrage automatique a nécessité de résoudre un cycle de dépendances systemd. La solution : définir

`WantedBy=graphical.target` et `After=network-online.target` sans référence circulaire à graphical.target.

Gestion de la mise en veille

L'écran se mettait en veille après 15 minutes d'inactivité, nécessitant un clic physique pour le réactiver. Deux solutions ont été mises en place :

- Désactivation complète du DPMS et du screen blanking via xset dans autostart
- Réveil automatique de l'écran via `xset dpms force on` avant chaque changement de chaîne

Spécifications techniques

Consommation et performances

- Consommation moyenne : 5W (vs 150W pour le PC Windows)
- Coût électrique annuel : ~5€ (vs ~150€)
- Utilisation RAM : ~2 Go sur 8 Go disponibles (25%)
- Chromium : 16 processus (architecture multi-processus normale)
- Température : Stable, pas de throttling

Choix de la RAM

Le Pi 5 existe en versions 2 GB, 4 GB et 8 GB. Pour cet usage :

- 2 GB : limite, risque de swap lors des pics mémoire
- 4 GB : suffisant et confortable

- 8 GB : confort maximal, marge importante pour RDP et multitâche

Le choix du 8 GB offre une tranquillité totale et de la marge pour d'éventuelles évolutions.

Bilan

La migration vers Raspberry Pi 5 a permis de créer une box TV fonctionnelle et autonome, avec une réduction drastique de la consommation électrique et l'élimination complète du bruit. Le contrôle depuis smartphone et l'automatisation du Cast audio offrent un confort d'utilisation supérieur à la solution PC.

Le système démarre automatiquement, ne nécessite aucune intervention physique pour un usage quotidien, et reste accessible à distance pour la maintenance.

Stack technique

- OS : Raspberry Pi OS (X11)
- Backend : Node.js, Puppeteer
- Automatisation : xdotool, xset, systemd
- Accès distant : xrdp
- Outils : unclutter, cec-utils (testé), pactl

Author



Patrick

GPM Factory