

Extinction et allumage automatique des TVs (Raspberry PI 4)

Solution 1 avec CEC (c'est la meilleur solution n'est pas compatible avec toutes les télévisions si ça ne marche pas utilisez la 2ème solution) :

Le **CEC** (Consumer Electronics Control) via l'interface HDMI permet d'envoyer des commandes telles que mettre en veille, allumer et éteindre.

Le **CEC** n'est disponible et utilisable que sur des téléviseurs récents. Certaines télévisions ne peuvent pas utiliser ce protocole, c'est pourquoi une autre solution est proposée : couper le flux HDMI (le signal envoyé à la télévision ou au moniteur), ce qui permet de ne plus diffuser sur la télé et de la mettre en mode veille. Nous pouvons à tout moment la rallumer en utilisant une autre solution, qui est wlr-randr.

Installation (uniquement sur Raspberry PI avec l'OS Raspberry PI os) :

Installer cec-utils

```
sudo apt update
sudo apt install cec-utils
```

Commandes pour contrôler la TV via CEC (certaines télévisions ne marcheront pas avec ce protocole HDMI, les télévisions sont parfois trop anciennes ou n'inclut pas cette fonctionnalité) :

Allumer la télé :

Installer cec-utils

```
echo "on 0" | cec-client -s -d 1
```

Éteindre la TV (veille) :

Installer cec-utils

```
echo "standby 0" | cec-client -s -d 1
```

Scanner les appareils connectés :

Installer cec-utils

```
cec-client -l
```

ou

Installer cec-utils

```
echo "scan" | cec-client -s -d 1
```

Script Node.js avec "child_process" :

Avec "child_process", on peut utiliser "exec" pour écrire des commandes dans le terminal de la Raspberry PI (ceci est script NodeJS):

Installer cec-utils

```
const { exec } = require('child_process');

let command = '';

if (action === 'on') command = 'echo "on 0" | cec-client -s -d 1';
if (action === 'off') command = 'echo "standby 0" | cec-client -s -d 1';

if (command) {
  exec(command, (error, stdout, stderr) => {
    if (error) {
      console.error(`Erreur : ${error.message}`);
      return;
    }
    console.log(`Commande exécutée : ${stdout}`);
  });
}
```

React Electron avec "ipcMain" :

Installer cec-utils

```
const { exec } = require('child_process');
const { ipcMain } = require('electron');

// Fonction pour scanner les téléviseurs
function scanTelevisions() {
  return new Promise((resolve, reject) => {
    exec('echo "scan" | cec-client -s -d 1', (error, stdout, stderr) => {
      if (error) {
        console.error(`Error scanning TVs: ${error.message}`);
        return reject(error);
      }
      resolve(stdout);
    });
  });
}

// Exposer la commande au frontend via IPC
ipcMain.handle('scan-cec', async () => {
  try {
    const result = await scanTelevisions();
    return result;
  } catch (error) {
    return { error: error.message };
  }
});
```

Installer "cec-utils" dans le conteneur Docker (Dockerfile Player) :

Installer cec-utils

```
RUN apt-get update && apt-get install -yq cec-utils && \
  apt-get clean && \
  rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*
```

Solution 2 alternative qui marche (wlr-randr, cette solution marche peu importe le type de télé ou d'écran) :

Pré-requis :

Branchez le câble Micro HDMI à la Raspberry PI 4 au port **HDMI-1**.

Installer les dépendances nécessaires

```
sudo apt update
sudo apt install wlr-randr
```

Voir les informations sur les appareils connectés avec soit "wlr-randr" ou "xrandr" :

Pour vérifier lequel utilisé il faut savoir l'affichage de l'écran (type de session en cours dans un environnement Linux) :

```
echo $XDG_SESSION_TYPE
```

Si la réponse est "**wayland**" continuer avec "wlr-randr" :

Permet de voir les infos sur l'appareil connecté et de récupérer sur quel port (HDMI-A-1 ou HDMI-A-2) est branché l'appareil :

```
wlr-randr
```

Permet d'allumer l'appareil (il va autorisé le signal HDMI du coup un flux sera envoyé a l'écran) :

```
wlr-randr --output HDMI-A-1 --on
```

Permet d'éteindre l'appareil (il va coupé le signal HDMI du coup aucun flux ne sera envoyé a l'écran, l'écran se mettra en veille automatiquement) :

```
wlr-randr --output HDMI-A-1 --off
```

Si la réponse est "**x11**" continuer avec "xrandr" :

Permet de voir les infos sur l'appareil connecté et de récupérer sur quel port (HDMI-A-1 ou HDMI-A-2) est branché l'appareil :

```
xrandr
```

Permet d'allumer l'appareil (il va autorisé le signal HDMI du coup un flux sera envoyé a l'écran) :

```
xrandr --output HDMI-1 --auto
```

Permet d'éteindre l'appareil (il va coupé le signal HDMI du coup aucun flux ne sera envoyé a l'écran, l'écran se mettra en veille automatiquement) :

```
xrandr --output HDMI-1 --off
```

Wake on lan (Allumer ou éteindre la Raspberry depuis le serveur NodeJS, Back) :

Dans le Back (NodeJs) :

Installer wake on lan dans NodeJs :

```
npm install wake_on_lan
```

Initialiser wake on lan dans nodejs :

```
const wol = require('wake_on_lan');

// Adresse MAC de votre Raspberry Pi
const macAddress = '00:11:22:33:44:55'; // Remplacez cette adresse MAC par celle de votre Raspberry Pi

// Envoyer le paquet WoL
wol.wake(macAddress, function(error) {
  if (error) {
    console.log('Erreur lors de l\'envoi du paquet WoL:', error);
  } else {
    console.log('Paquet WoL envoyé avec succès!');
  }
});
```

Sur la Raspberry :

Récupérer l'adresse Mac :

```
ip link show
```

Vérifier si votre Raspberry prend en charge Wake-on-LAN (WOL) :

```
sudo ethtool eth0
```

Si Wake-on: d, alors Wake-on-LAN n'est pas supporté.

Si Wake-on: g, alors Wake-on-LAN est supporté.

Activer Wake-on-LAN (si désactivé) :

Si Wake-on-LAN est désactivé, vous devrez l'activer.

Solution avertissement "dpkg" (facultatif) :

Éditer le fichier status :

```
sudo nano /var/lib/dpkg/status
```

Ajouter la ligne suivante dans la section Package: acenstream-player (juste en dessous de Installed-Size) :

```
Maintainer: Acenstream
```

Réinstaller l'OS de la Raspberry pi (facultatif) :

Mettre à jour les paquets :

```
sudo apt update
```

Installer les composants nécessaires :

```
sudo apt install raspberrypi-ui-mods
sudo apt install raspberrypi-bootloader
sudo apt install raspberrypi-kernel
```

Redémarrer la Raspberry Pi :

```
sudo reboot
```

Documentation :

Outils différents sur la Raspberry : <http://papy-tux.legtux.org/doc1337/index.html>

Wlr-randr : <https://manpages.debian.org/testing/wlr-randr/wlr-randr.1.en.html>

Xrandr : <https://doc.ubuntu-fr.org/xrandr>