MXQ Pro (S905W) – From Android TV Box to Linux Node

Objectif

Recycler un boîtier TV Android basé sur un **Amlogic S905W** afin d'en faire un **nœud Linux headless**, destiné à des usages réseau tels que :

- VPN,
- accès distant,
- services légers.

Contexte

Projet personnel d'étudiant, intégré à un **homelab**, documenté dans une démarche de **portfolio technique**.
L'objectif est autant fonctionnel que pédagogique, avec un fort accent mis sur la compréhension du matériel, du processus de boot et du diagnostic système.

Présentation du matériel

Boîtier

- Modèle : MXQ Pro (Android TV Box)
- SoC : Amlogic S905W
- CPU : Quad-core ARM Cortex-A53
- RAM : ~1 Go
- Stockage interne : eMMC (environ 8 à 16 Go selon le modèle)
- Stockage externe : carte micro-SD

- Réseau : Ethernet 100 Mbps
- Usage initial : Android TV (obsolète)

Pourquoi le S905W est intéressant (et limité)

Le **S905W** est un SoC très répandu dans les TV box à bas coût.
Il n'est pas officiellement supporté par Armbian, ce qui implique une dépendance à :

- un u-boot communautaire,
- des Device Tree (`.dtb`) adaptés,
- une phase importante d'essais et d'erreurs.

Cela en fait une plateforme avec un **fort intérêt pédagogique**, malgré des limites claires en termes de support et de performances.

Sources et choix de la build Armbian

Le système d'exploitation retenu pour ce projet est **Armbian**, basé sur **Debian Bookworm**, via une build communautaire adaptée aux boîtiers TV Amlogic (S905W).

Les images officielles Armbian ne prenant pas en charge ce type de matériel, le choix s'est porté sur une **build communautaire maintenue activement**, spécifiquement conçue pour les SoC Amlogic S9xxx.

Dépôt GitHub principal

Les images utilisées proviennent du dépôt suivant :

https://github.com/ophub/amlogic-s9xxx-armbian

Ce dépôt fournit :

- des images Armbian préconfigurées pour les SoC Amlogic (S905W, S905X, S912, etc.),
- des kernels adaptés (legacy et mainline selon les versions),
- une documentation détaillant les méthodes de boot, de configuration DTB et de dépannage.

Il s'agit aujourd'hui de **la référence communautaire** pour l'exécution d'Armbian sur les TV box Amlogic.

Version utilisée dans ce projet

La build sélectionnée correspond à :

- **Armbian basé sur Debian Bookworm**
- Architecture : **arm64**
- Kernel : **vendor / legacy (5.15 ou 6.1 selon la release)**

Les releases sont accessibles directement depuis la section *Releases* du dépôt :

https://github.com/ophub/amlogic-s9xxx-armbian/releases

Le choix d'une version Bookworm a été motivé par :

- la stabilité à long terme de Debian,
- la compatibilité avec les usages serveur (VPN, SSH, services réseau),
- la cohérence avec le reste du homelab basé sur Debian / Proxmox.

Documentation associée

La documentation officielle du projet est disponible ici :

https://github.com/ophub/amlogic-s9xxx-armbian/blob/main/README.md

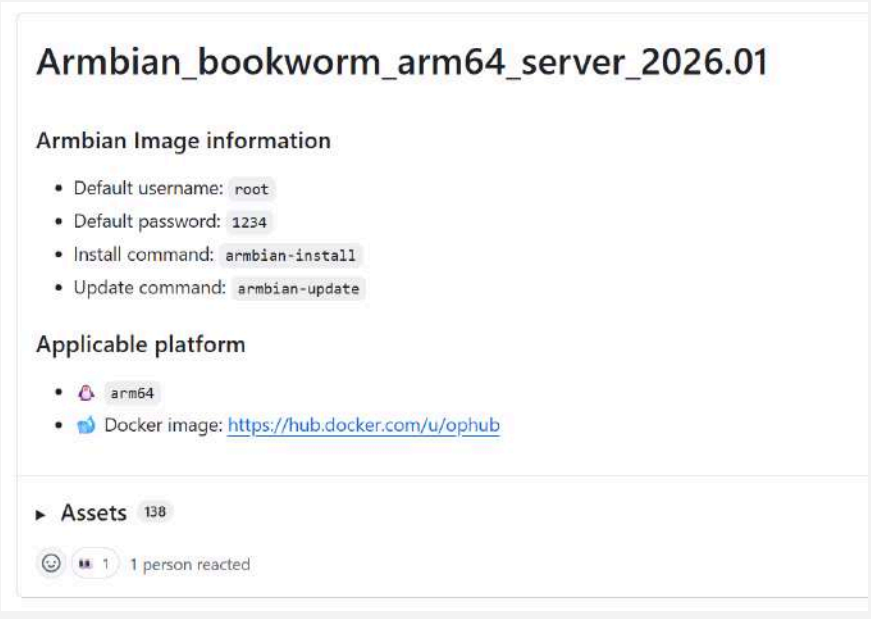
Elle couvre notamment :

- les méthodes de flash sur carte SD,
- la sélection et l'utilisation des Device Tree (`.dtb`),
- les procédures de dépannage courantes,
- les limitations connues selon les modèles de boîtiers.

Choix de l'OS et justification

OS retenu

- Armbian
- Base : Debian Bookworm
- Kernel : legacy / vendor (5.15 ou 6.1 selon l'image)



Pourquoi Debian Bookworm

- Stabilité à long terme.
- Écosystème serveur mature.
- Compatibilité native avec WireGuard et OpenVPN.
- Cohérence avec le reste du homelab (serveur Proxmox basé sur Debian).

Dans un contexte headless orienté réseau, Debian a été préféré à Ubuntu.

Device Tree (DTB) : point central du projet

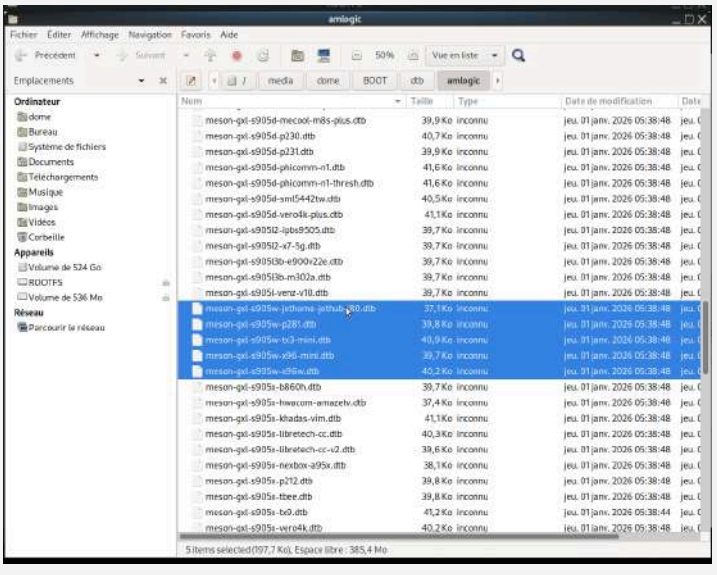
Le **Device Tree** définit la manière dont le matériel est exposé au noyau Linux.
Sur S905W, un DTB incorrect entraîne :

- absence de réseau,
- échec du boot,
- écran figé sur le logo.

DTB testés :

- `meson-gxl-s905w-p281.dtb`
- `meson-gxl-s905w-x96-mini.dtb`
- `meson-gxl-s905w-tx3-mini.dtb`

Voir photo : dossier `/dtb/amlogic` contenant plusieurs fichiers `.dtb`.

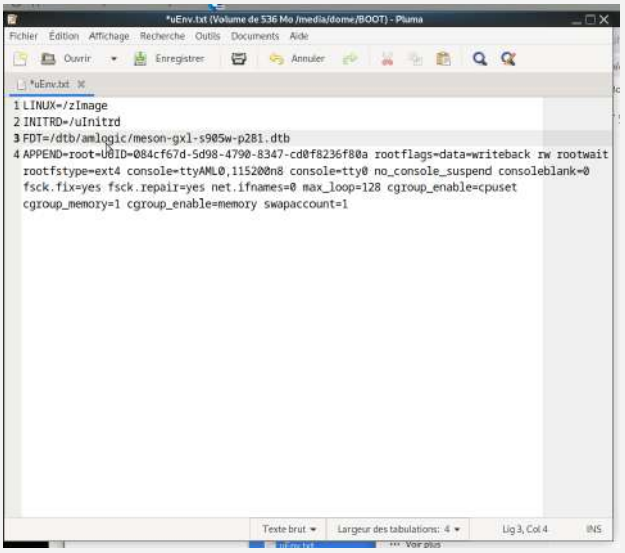


Configuration finale retenue

Dans le fichier `uEnv.txt` :

- `FDT=/dtb/amlogic/meson-gxl-s905w-p281.dtb`

Voir photo : édition de `uEnv.txt`.



Problème critique : alimentation

Un point souvent sous-estimé.

Symptômes observés

- Boot aléatoire.
- Écran bloqué sur le logo.
- Reboots en boucle.
- Services instables, notamment SSH.

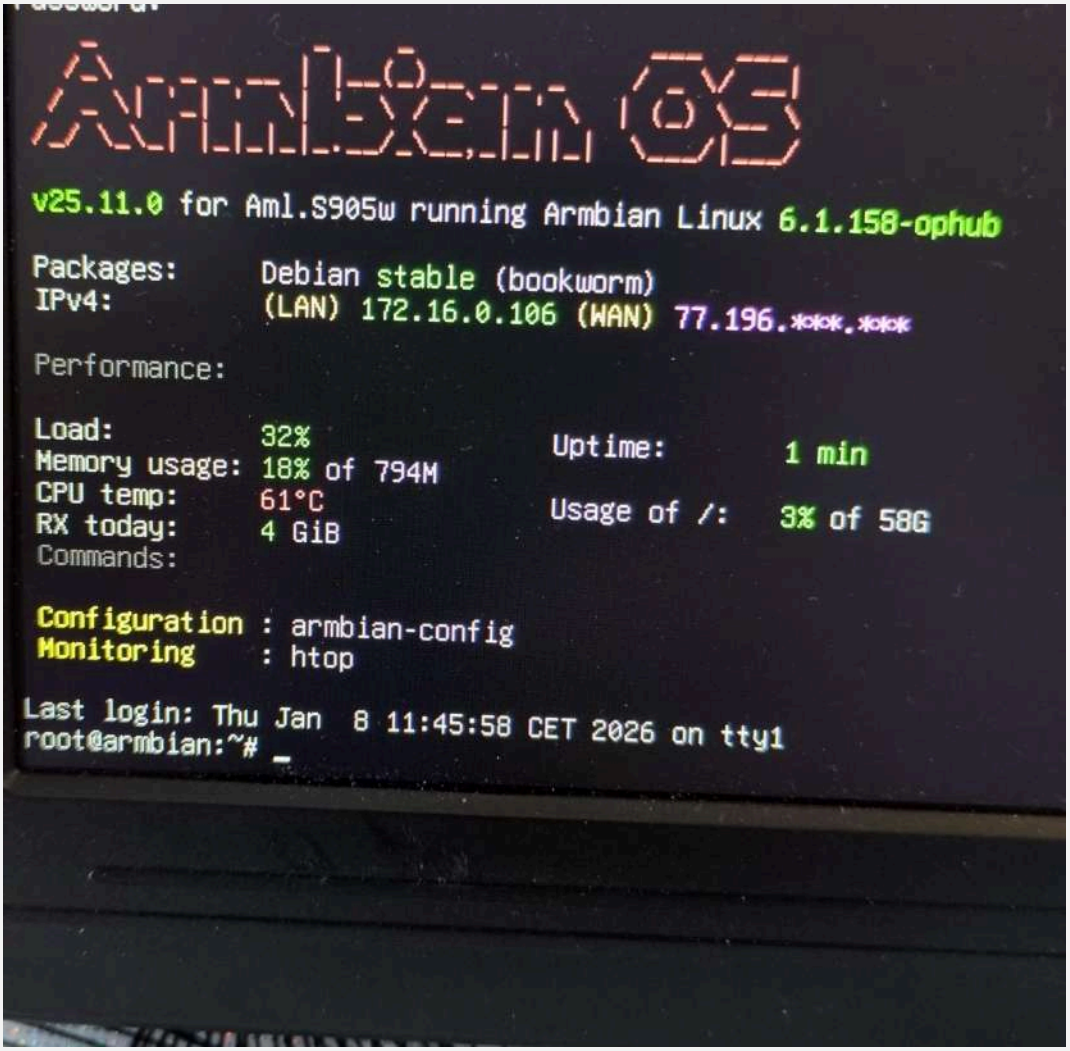
Solution

- Remplacement par une alimentation 5V 2A stable et de meilleure qualité.
- Boot immédiatement fiable après remplacement.

ancienne alimentation

nouvelle alimentation

Conclusion : le problème n’était pas logiciel.



À l’invite :

```
armbian login:
```

Identifiants par défaut :

```
login: root
```

```
password: 1234
```

Note importante: Clavier Qwerty!!

Configuration clavier (US vers FR)

Problème

- Console configurée par défaut en QWERTY.
- Inconfort important lors des phases de debug.

Solution temporaire

- `loadkeys fr`

Suffisant pour la phase d’installation.
Une configuration persistante est prévue ultérieurement dans un contexte entièrement headless.

Important : dans une démarche visant à éviter les problèmes ultérieurs, il est recommandé d’exécuter un

```
apt update && apt upgrade -y
```

afin de partir sur une base système saine et à jour.

Solution définitive et fiable pour la configuration du clavier (US vers FR)

Contexte

Sur les images Armbian destinées aux boîtiers TV basés sur des SoC Amlogic, la configuration standard du clavier via les mécanismes Debian classiques (`keyboard-configuration`, `console-setup`) n’est pas systématiquement appliquée au démarrage.

Dans ce projet, plusieurs méthodes ont été testées afin de rendre la disposition clavier AZERTY persistante :

- reconfiguration via `dpkg-reconfigure keyboard-configuration`,
- redémarrage des services `keyboard-setup`,
- modification des fichiers de configuration standards Debian.

Malgré une configuration correcte sur le papier, ces approches se sont révélées **inefficaces après redémarrage**, le clavier revenant systématiquement en disposition US.
Ce comportement est lié au processus de boot spécifique des images Armbian pour TV box, qui surcharge ou ignore certaines étapes classiques d'initialisation.

Choix de la solution

Face à l'échec des méthodes conventionnelles, il a été décidé d'opter pour une **solution plus radicale mais totalement fiable** : forcer explicitement le chargement du keymap français à chaque démarrage, indépendamment des mécanismes internes d'Armbian.

Cette approche repose sur un principe simple :

- appliquer la configuration clavier **après l'initialisation de la console**,
- garantir son exécution **à chaque boot**,
- ne dépendre d'aucun comportement implicite du système.

Étape 1 – Vérifier que le keymap FR existe

```
ls /usr/share/keymaps/i386/azerty/fr.kmap.gz
```

Si le fichier existe → OK
Sinon :

```
apt install console-data
```

Étape 2 – Créer un service systemd dédié

```
nano /etc/systemd/system/keyboard-fr.service
```

Taper exactement ceci :

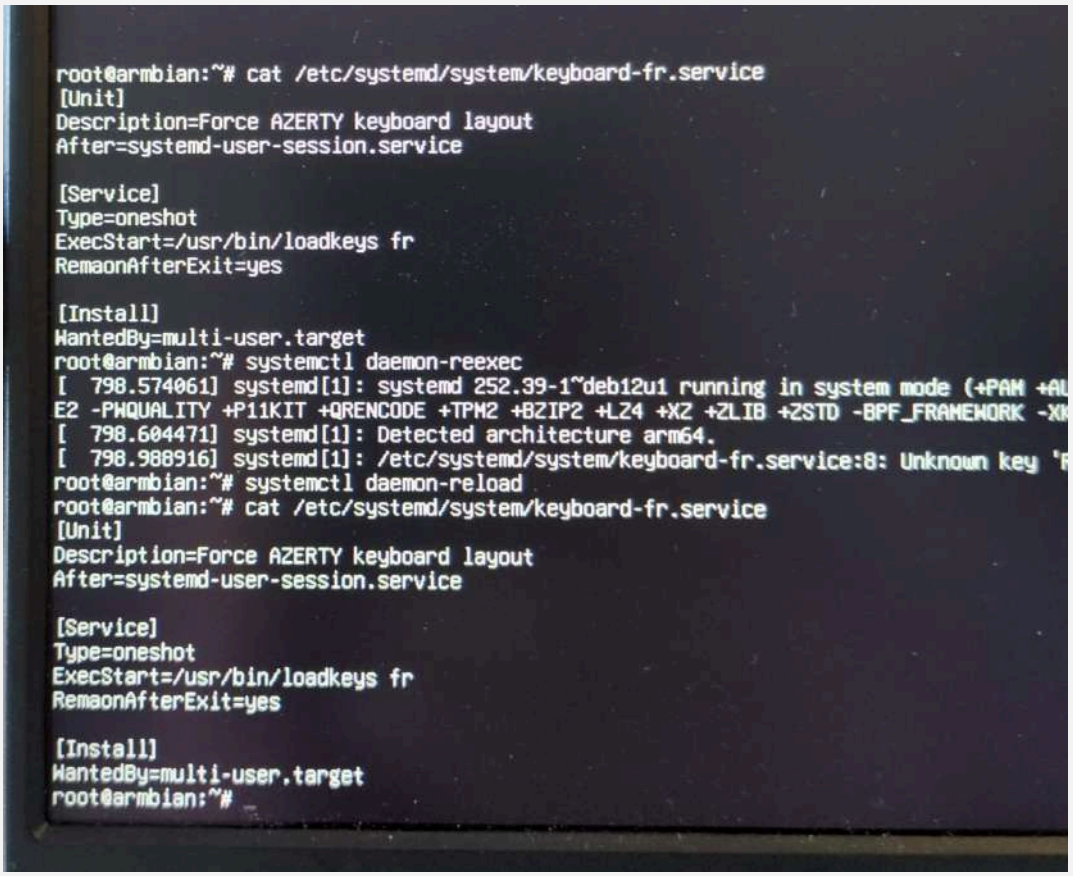
```
[Unit]
Description=Force AZERTY keyboard layout
After=systemd-user-sessions.service

[Service]
Type=oneshot
ExecStart=/usr/bin/loadkeys fr
RemainAfterExit=yes

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

Étape 3 – Activer le service

```
systemctl daemon-reexec
systemctl daemon-reload
systemctl enable keyboard-fr.service
```



Étape 4 – Reboot et validation

```
reboot
```

Après reboot, tester immédiatement :

- `a` → doit écrire `a`
- `q` → doit écrire `a`
- `;` → doit écrire `m`

Si c'est bon → **c'est définitivement réglé**.

Problèmes SSH et résolution

Symptômes

- Service `ssh.service` en échec.
- Impossible de se connecter à distance.
- `systemctl status ssh` indiquant un état FAILED.

Voir photos : erreurs SSH, codes de sortie et journaux systemd.

Diagnostic

- Fichier `sshd_config` partiellement corrompu (directive `Subsystem sftp`).
- Absence totale de clés hôte SSH.

Erreur observée :

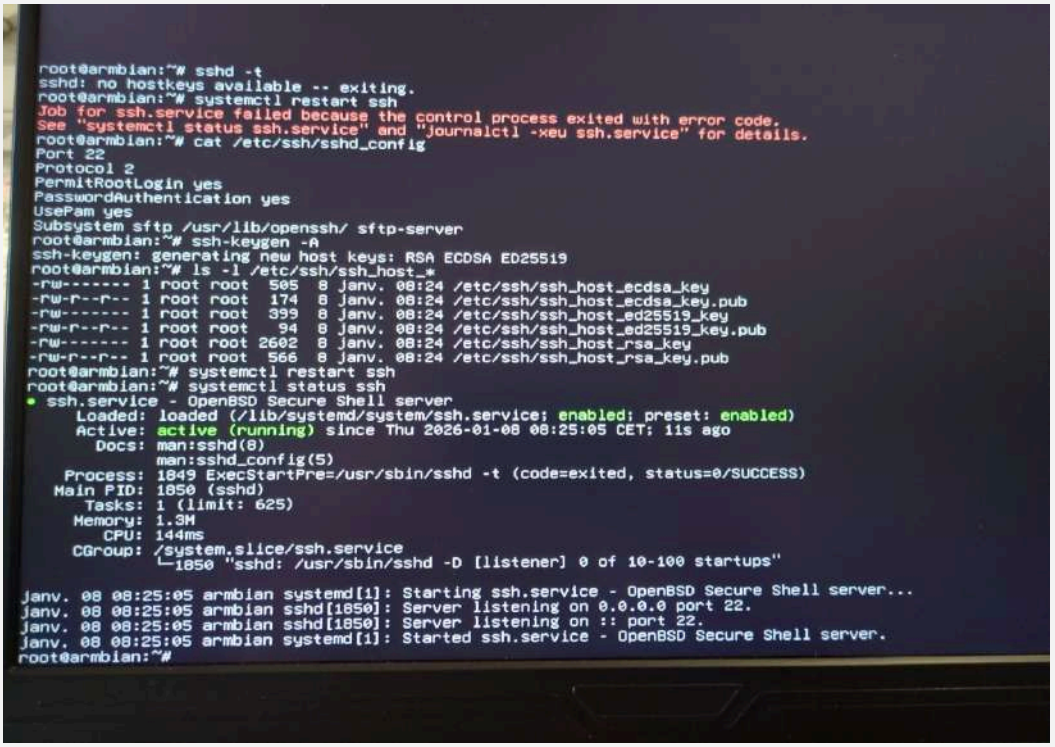
- `sshd: no hostkeys available`

Résolution

- `ssh-keygen -A`

- `systemctl restart ssh`

Voir photo : génération des clés et service SSH en état actif.



Résultat :

- Connexion SSH fonctionnelle.
- Objectif initial atteint.

Possibilités de flash sur les cartes Amlogic S905W

Sur ce type de boîtier, trois approches principales existent.

1. Boot sur carte SD (non destructif – choix initial)

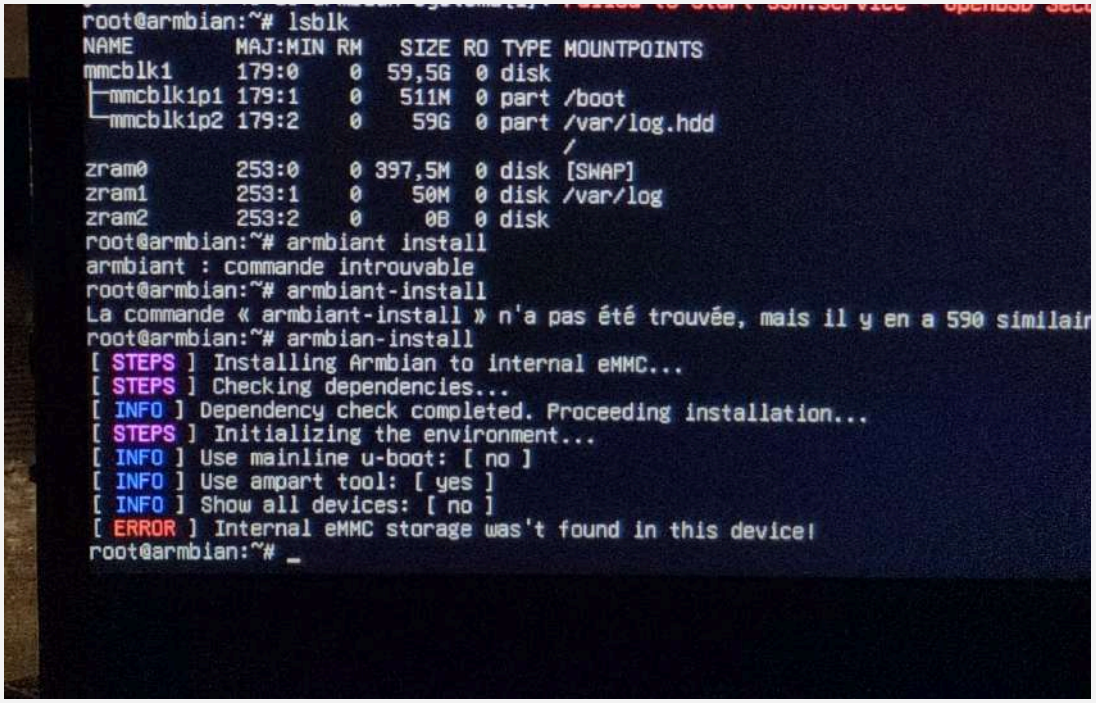
- Armbian est lancé depuis la carte micro-SD.
- L’eMMC interne reste intacte.
- Méthode la plus sûre pour tester et itérer.

---

2. Installation sur eMMC (destructif)

- Armbian remplace Android.
- Démarrage plus rapide.
- Procédure difficilement réversible sans outils Amlogic.

Voir photo : commande `armbian-install` indiquant l'absence d'eMMC détectée.



Sur certains boîtiers Amlogic S905W, l'installation d'Armbian sur le stockage interne (eMMC) est théoriquement possible via l'outil `armbian-install`.

Dans le cadre de ce projet, cette option a été étudiée mais **n'a pas pu être mise en œuvre**, pour les raisons suivantes :

- l'eMMC interne n'a pas été détectée comme périphérique cible par Armbian,
- l'outil `armbian-install` a explicitement indiqué l'absence de stockage interne exploitable,
- ce comportement est connu sur certains modèles de TV box, en fonction du matériel exact et du support du bootloader.

En conséquence, le système Armbian a été conservé en exécution **depuis la carte micro-SD**, ce qui reste une solution fonctionnelle, stable et suffisante pour les usages prévus (VPN, accès distant, services réseau légers).

Cette limitation n'est **pas liée à une instabilité du système**, mais à des contraintes matérielles et de support propres à ce modèle de boîtier.

3. Flash via USB Burning Tool (non retenu)

- Nécessite un environnement Windows.
- Dépendance à des drivers propriétaires.
- Peu cohérent avec un workflow Linux / homelab.

État final du système

- Armbian démarre de manière stable.
- Réseau Ethernet fonctionnel.
- Accès SSH opérationnel.
- Installation sur eMMC volontairement non effectuée.
- Système prêt pour un usage VPN ou gateway réseau.

Voir photo :

Prochaine étape

Un dépôt séparé sera consacré à la mise en place du VPN :

Nom prévu :

- `mxq-pro-vpn-node`

Contenu envisagé :

- WireGuard,
- accès distant sécurisé,
- intégration au homelab Proxmox,
- routage et NAT si nécessaire.

“Si vous êtes arrivé jusqu’ici, félicitations : vous avez survécu aux Device Tree, aux alimentations douteuses, aux claviers QWERTY rebelles et à un SSH plus têtu qu’un bootloader Amlogic.

Merci d’avoir pris le temps de lire ce retour d’expérience, né de nombreuses itérations, de quelques frustrations, et de beaucoup d’apprentissage pour moi.”