

sunshine remote setup

This guide documents the complete, native, headless gaming and remote streaming stack on **Arch Linux** (`ul.drheck.dev`) using **Sunshine**, **Gamescope**, and **Steam**, accelerated by the dedicated **AMD Radeon AI PRO R9700 (32GB VRAM)**.

Clients connect using the open-source **Moonlight** client across any device (Desktop, Android TV, iOS, handhelds).

1. System Architecture & Network Topology

- **Host OS:** Arch Linux (Rolling, LTS Kernel `6.18.x+`)
 - **Primary Gaming GPU:** AMD Radeon AI PRO R9700 (`1002:7551`, Navi 48, DRM `/dev/dri/card1` & `/dev/dri/renderD128`)
 - **Server BMC Graphics:** ASPEED AST2600 (`card0` / `VGA-1` — ignored for gaming)
 - **Compositor:** Valve's **Gamescope** micro-compositor (DRM mode, Vulkan/RADV accelerated)
 - **Streaming Server:** **Sunshine** (Native binary from AUR, KMS capture + VA-API hardware encoding)
 - **Audio Engine:** **PipeWire** & **PipeWire-Pulse**
 - **Network Binding:**
 - **Dedicated IPv6 Address:** `[fd77:7777:5010:0:1337:f1ac:d00d:4799]`
 - **Dedicated IPv4 Address:** `10.0.0.112`
 - **Admin Web Portal:** `https://sunshine.drheck.dev` (Authentik forward-auth protected, routed via Nginx)
-

2. Required Packages

Install the native gaming stack, graphics drivers, and audio subsystems:

```
# Core tools, AMD Vulkan drivers, Gamescope, Steam, and audio
sudo pacman -S --needed gamescope steam vulkan-radeon lib32-vulkan-radeon \
```

```
pipewire pipewire-pulse wireplumber seatd socat libcap
```

```
# Sunshine binary from AUR
paru -S sunshine-bin
```

Ensure user `luke` belongs to the required hardware access groups:

```
sudo usermod -aG video,audio,input,render,seat luke
```

Enable systemd lingering so the user's services run 24/7 in the background without requiring an active shell login:

```
loginctl enable-linger luke
```

3. Headless Display via Custom Boot EDID

On a headless server with no physical monitor plugged into the AMD GPU, the `amdgpu` driver will report `disconnected` and reject mode setting. To solve this without purchasing physical HDMI/DP dummy dongles, the kernel is configured to load an **EDID firmware file at boot** to simulate a physical 1080p 144Hz monitor on `DP-1`.

Step 1: Place Verified EDID Binary

Download and save a verified 1080p DisplayPort EDID (256-byte CEA-861 compliant binary) to the firmware directory:

```
sudo mkdir -p /usr/lib/firmware/edid
curl -s "https://git.linuxtv.org/v4l-utils.git/plain/utils/edid-decode/data/aoc-c24g1-dp" |
sudo tee /usr/lib/firmware/edid/aoc_1080p.bin > /dev/null
```

Step 2: Bundle EDID into Initramfs

Because `amdgpu` loads early during boot (via the `kms` hook), the EDID binary must be packaged inside the initramfs. Edit `/etc/mkinitcpio.conf`:

```
FILES=(/usr/lib/firmware/edid/aoc_1080p.bin)
```

Regenerate the initramfs:

```
sudo mkinitcpio -P
```

Step 3: Configure Bootloader Kernel Parameters

In `/boot/loader/entries/arch.conf`, append the `drm.edid_firmware` and `video` parameters to the `options` line:

```
title Arch
linux /vmlinuz-linux-lts
initrd /amd-ucode.img
initrd /initramfs-linux-lts.img
options root="UUID=4496ac5e-4538-4ce4-8ca2-afc0b828ee80" rw drm.edid_firmware=DP-1:edid/aoc_1080p.bin video=DP-1:1920x1080@60e
```

After rebooting, the kernel recognizes `DP-1` as a permanently connected 1080p DisplayPort monitor with all standard refresh rates.

4. Dual-Stack Networking & IPv4 socat Bridge

In our dual-stack architecture on `u1`, Sunshine binds natively to its dedicated IPv6 ULA address. To allow IPv4 clients to connect without asymmetric routing issues, a dedicated `socat` `systemd` service listens on `10.0.0.112` and forwards bidirectionally to the IPv6 address.

Bridge Script: `/usr/local/bin/sunshine-ipv4-bridge.sh`

```
#!/usr/bin/env bash
IPV4="10.0.0.112"
```

```
IPV6="fd77:7777:5010:0:1337:flac:d00d:4799"
```

```
TCP_PORTS=(47984 47989 47990 48010)
```

```
UDP_PORTS=(47998 47999 48000 48002 48010)
```

```
pids=()
```

```
term_handler() {
```

```
    for pid in "${pids[@]}; do
```

```
        kill -TERM "$pid" 2>/dev/null
```

```
    done
```

```
    wait
```

```
    exit 0
```

```
}
```

```
trap term_handler SIGINT SIGTERM
```

```
for port in "${TCP_PORTS[@]}; do
```

```
    socat TCP4-LISTEN:"$port",bind="$IPV4",reuseaddr,fork TCP6:["$IPV6"]:"$port" &
```

```
    pids+=($!)
```

```
done
```

```
for port in "${UDP_PORTS[@]}; do
```

```
    socat UDP4-LISTEN:"$port",bind="$IPV4",reuseaddr,fork UDP6:["$IPV6"]:"$port" &
```

```
    pids+=($!)
```

```
done
```

```
wait
```

Make executable: `sudo chmod +x /usr/local/bin/sunshine-ipv4-bridge.sh`

Systemd Unit:

`/etc/systemd/system/sunshine-ipv4-bridge.service`

```
[Unit]
```

```
Description=Sunshine IPv4 to IPv6 socat Bridge
```

```
After=network.target
```

```
[Service]
Type=simple
ExecStart=/usr/local/bin/sunshine-ipv4-bridge.sh
Restart=always
RestartSec=3

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

Enable and start:

```
sudo systemctl daemon-reload
sudo systemctl enable --now sunshine-ipv4-bridge.service
```

Nginx Reverse Proxy Config

Nginx on `ul` handles SSL termination and Authentik ForwardAuth for the web interface in `/etc/nginx/sites-available/sunshine`:

```
location / {
    proxy_pass https://10.0.0.112:47990;
    proxy_ssl_verify off;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection $connection_upgrade_keepalive_sunshine;
    proxy_set_header Host $host;
    proxy_buffering off;
}
```

5. Permissions & Hardware Capabilities

1. Game Controller Permissions (`/dev/uinput` and `/dev/uhid`)

Sunshine emulates Xbox controllers via `uinput` and native PlayStation 5 DualSense controllers via `uhid`. Ensure `/etc/udev/rules.d/85-sunshine-uinput.rules` contains:

```
KERNEL=="uinput", SUBSYSTEM=="misc", GROUP="input", MODE="0660"
KERNEL=="uhid", SUBSYSTEM=="misc", GROUP="input", MODE="0660"
```

Ensure the `uhid` kernel module loads at boot:

```
echo "uhid" | sudo tee /etc/modules-load.d/uhid.conf
```

2. Session Management (`seatd`)

To allow Gamescope to open DRM display sessions without an interactive login seat, configure `/etc/systemd/system/seatd.service.d/group.conf`:

```
[Service]
ExecStart=
ExecStart=seatd -g video
```

Enable and start: `sudo systemctl enable --now seatd`

4. DualSense HIDAPI Permissions (`/dev/hidraw*`)

Steam accesses DualSense controllers via its raw HIDAPI driver on `/dev/hidraw*`. On a headless server without an interactive physical seat, the default `uaccess` tag fails to grant permissions to non-root users. Create `/etc/udev/rules.d/99-steam-controller-perms.rules`:

```
KERNEL=="hidraw*", KERNELS=="*054C:0CE6*", GROUP="input", MODE="0666"
KERNEL=="hidraw*", ATTRS{idVendor}=="054c", ATTRS{idProduct}=="0ce6", GROUP="input",
MODE="0666"
SUBSYSTEM=="hidraw", ENV{HID_NAME}=="*DualSense*", GROUP="input", MODE="0666"
SUBSYSTEM=="hidraw", ENV{HID_NAME}=="*Wireless Controller*", GROUP="input", MODE="0666"
SUBSYSTEM=="hidraw", ENV{HID_PHYS}=="libvirtualhid/uhid/*", GROUP="input", MODE="0666"
```

Reload udev rules:

```
sudo udevadm control --reload-rules && sudo udevadm trigger
```

3. Binary Capabilities

Allow Gamescope to set real-time thread scheduling and Sunshine to capture DRM frames without root:

```
sudo setcap 'cap_sys_nice=eip' /usr/bin/gamescope  
sudo setcap 'cap_sys_admin,cap_sys_nice=eip' /usr/bin/sunshine
```

6. Sunshine Configuration

File: `~/.config/sunshine/sunshine.conf`

```
system_tray = disabled  
min_log_level = info  
origin_pin_allowed = lan  
origin_web_ui_allowed = lan  
csrf_allowed_origins = https://sunshine.drheck.dev  
  
# Network binding  
bind_address = fd77:7777:5010:0:1337:flac:d00d:4799  
address_family = both  
external_ip = 10.0.0.112  
  
# Explicit GPU & Display pinning (prevents capturing motherboard ASPEED VGA)  
adapter_name = /dev/dri/card1  
output_name = DP-1  
gamepad = ds5
```

File: `~/.config/sunshine/apps.json`

```
{
  "env": {
    "PATH": "${PATH}:${HOME}/bin:${HOME}/.local/bin"
  },
  "apps": [
    {
      "name": "Steam Big Picture",
      "cmd": "/home/luke/bin/launch-gamescope-steam.sh",
      "image-path": "steam.png"
    }
  ]
}
```

7. Gamescope Launch Script

Create `/home/luke/bin/launch-gamescope-steam.sh`:

```
#!/usr/bin/env bash
set -e

WIDTH="${SUNSHINE_CLIENT_WIDTH:-1920}"
HEIGHT="${SUNSHINE_CLIENT_HEIGHT:-1080}"
FPS="${SUNSHINE_CLIENT_FPS:-60}"

export XDG_RUNTIME_DIR="/run/user/1000"
export LIBSEAT_BACKEND="seatd"
export RADV_PERFTEST="video_encode"
export MESA_VK_DEVICE_SELECT="1002:7551"

# Force Steam to use SDL evdev instead of broken HIDAPI on virtual DualSense
export SDL_JOYSTICK_HIDAPI_PS5=0
export SDL_JOYSTICK_HIDAPI=0

# Verified DualSense mapping for Sunshine virtual pad (including PS Guide button b10)
export SDL_GAMECONTROLLERCONFIG="050057564c050000e60c000000006800,PS5
Controller,crc:5657,platform:Linux,a:b0,b:b1,x:b3,y:b2,dpleft:h0.8,dpright:h0.2,dpup:h0.1,dpdown:h0.4,leftx:a0,lefty:a1,leftstick:b11,rightx:a3,righty:a4,rightstick:b12,leftshoulder:b4,lef
```

```
ttrigger:a2,rightshoulder:b5,righttrigger:a5,back:b8,start:b9,guide:b10,steam:2,050000004c050000e60c000011810000,PS5
```

```
Controller,crc:519b,platform:Linux,a:b0,b:b1,x:b3,y:b2,dpleft:h0.8,dpright:h0.2,dpup:h0.1,dpdown:h0.4,leftx:a0,lefty:a1,leftstick:b11,rightx:a3,righty:a4,rightstick:b12,leftshoulder:b4,lefttrigger:a2,rightshoulder:b5,righttrigger:a5,back:b8,start:b9,guide:b10,steam:2,"
```

```
exec gamescope -e --backend drm --prefer-vk-device 1002:7551 -W "$WIDTH" -H "$HEIGHT" -r "$FPS" -- steam -bigpicture
```

Make executable: `chmod +x /home/luke/bin/launch-gamescope-steam.sh`

8. Sunshine Systemd User Service

Create `~/.config/systemd/user/sunshine.service`:

```
[Unit]
Description=Sunshine Game Stream Host (Headless)
After=network.target pipewire.service
Wants=pipewire.service

[Service]
Type=simple
ExecStart=/usr/bin/sunshine /home/luke/.config/sunshine/sunshine.conf
Restart=always
RestartSec=3

[Install]
WantedBy=default.target
```

Enable and start:

```
systemctl --user daemon-reload
systemctl --user enable --now sunshine.service
```

9. Dedicated Game Storage Configuration (`/data/games/steam`)

To prevent Steam games, shader caches, and downloading staging files from filling up the root NVMe SSD (`/`), the game library is permanently linked to the **151 TB ZFS storage pool** at `/data/games/steam/`:

1. Storage Directory:

```
mkdir -p /data/games/steam/steamapps
chown -R luke:luke /data/games/steam/steamapps
```

2. Permanent Symlinks:

```
ln -sf /data/games/steam/steamapps ~/.local/share/Steam/steamapps
ln -sf /data/games/steam/steamapps ~/.steam/steam/steamapps
```

3. Steam Library Config (`~/.local/share/Steam/config/libraryfolders.vdf`):

Only library folder `"0"` (`/home/luke/.local/share/Steam`) is configured, seamlessly routing 100% of game downloads directly to `/data/games/steam/steamapps/`. Zero bytes of game files touch the root drive.

10. Troubleshooting & Gotchas

Gotcha 1: Moonlight Pairing Fails with "Client is not authorized"

- **Symptom:** You enter the 4-digit PIN in the Sunshine web UI and it reports success, but Moonlight immediately says *"Host PC returned error: client is not authorized"*.
- **Root Cause:** A known Sunshine bug (Issue #5696). If `sunshine_state.json` accumulates duplicate device entries or certificate collisions from multiple re-pair attempts, TLS verification fails.
- **Fix:**
 1. Stop Sunshine: `systemctl --user stop sunshine.service`
 2. Clear the paired devices array:

```
python3 -c '
import json

p = "/home/luke/.config/sunshine/sunshine_state.json"

with open(p) as f: d = json.load(f)

d["root"]["named_devices"] = []

with open(p, "w") as f: json.dump(d, f, indent=4)
'
```

3. Start Sunshine: `systemctl --user start sunshine.service`
4. Pair your Moonlight client with a fresh PIN.

Gotcha 2: Sunshine Startup Encoder Probe Delay ("Establishing Connection...")

- **Symptom:** When connecting right after starting/restarting Sunshine, Moonlight hangs at *"Establishing connection..."* for up to 50 seconds.
- **Explanation:** At startup, Sunshine runs sequential self-tests for all hardware codecs (H.264, HEVC, AV1 in 8-bit and 10-bit). During this probe, the streaming engine is busy and pauses client connections. Wait ~50 seconds after service start before initiating a stream.

Gotcha 3: Motherboard ASPEED Video Hijacking Stream

- **Symptom:** The stream opens but displays the Linux text console (TTY1) with kernel messages instead of the game.
- **Root Cause:** Dual-GPU servers have an onboard ASPEED AST2600 BMC chip (`card0` / `VGA-1`). Without explicit configuration, Sunshine selects `card0`.
- **Fix:** Ensure `adapter_name = /dev/dri/card1` and `output_name = DP-1 gamepad = ds5` are explicitly set in `~/.config/sunshine/sunshine.conf`.

Gotcha 4: PS5 DualSense Controller Layout & Android TV

- **Sony / Android TV Home Button:** Android TV consumes the physical PS button as an OS-level Home key. To access the Steam menu in Big Picture:
 - **Hold the** `Start / Options` **button** for 1.5 seconds, OR

- Press **Share + Options** simultaneously.
- **Touchpad as Mouse:** Android TV does not pass through gamepad touchpad data over Bluetooth; use the analog stick or Moonlight's built-in mouse mode (**Hold Start**).

Gotcha 5: Native Linux Builds vs. Proton

- For games that offer official Linux binaries (e.g. *Hollow Knight*, *VirtuaVerse*), force the native build in Steam:
 - Game page → **Properties** (Gear) → **Compatibility** → Check "*Force the use of a specific Steam Play compatibility tool*" → Select "**Steam Linux Runtime**".
 - This runs the game natively on Linux without Wine or DXVK overhead.

Gotcha 6: Linux Console Displaying on TV on Multi-GPU Servers

- **Symptom:** Moonlight displays a Linux text console instead of Steam Big Picture.
- **Root Cause:** On dual-GPU servers (ASPEED BMC `card0` + discrete gaming GPU `card1`), Linux `fbcon` binds console output across all framebuffers (`fb0` and `fb1`). If Gamescope is not actively running, `tty1` mirrors to `DP-1`.
- **Fix:** Keep `getty@tty1.service` enabled for the BMC/VGA console. Ensure Gamescope is active so it claims the display surface, or map `fbcon` exclusively to the BMC adapter (`fbcon=map:0`).

Gotcha 7: Controller Registers as Keyboard & Mouse in Games (e.g. Portal)

- **Symptom:** The mouse moves on screen and clicks, but physical controller buttons are not registering in the game or act like keyboard keys.
- **Root Cause:**
 1. For PC releases of games like *Portal* and *Half-Life 2*, Steam Input defaults to a community layout called "*Keyboard (WASD) and Mouse*".
 2. Additionally, Moonlight Android TV features an on-screen mouse emulation mode toggled by holding **Options / Start for 1.5 seconds**.
- **Fix:**
 - In Steam Big Picture: Navigate to the game page → click the **Controller Icon** → change **Current Layout** to "**Gamepad**".
 - On your controller: Ensure Moonlight is not in mouse mode by holding **Options / Start** for 1.5s to toggle back to Gamepad mode.

Gotcha 8: PS5 Controller Drops After 5 Seconds (`hid_read` failure)

- **Symptom:** The virtual DualSense appears in Steam Controller Settings upon connection, but after ~5 seconds it disconnects, reports `Controller device closed after hid_read failure` in `~/.local/share/Steam/logs/controller.txt`, and disappears from the UI.
- **Root Cause:** Steam has a built-in HIDAPI driver for Sony DualSense (`054c:0ce6`). When permissions are granted to `/dev/hidraw*`, Steam attempts to poll proprietary Sony hardware feature reports (gyro calibration, adaptive trigger registers). Sunshine's virtual `uhid` controller does not implement these low-level reports, causing `hid_read` to time out and Steam to terminate the controller connection.
- **Fix:** In `/home/luke/bin/launch-gamescope-steam.sh`, export:

```
export SDL_JOYSTICK_HIDAPI_PS5=0
export SDL_JOYSTICK_HIDAPI=0
```

This forces Steam to bypass raw HIDAPI and interact with the controller via standard Linux evdev (`SDL://`), which works reliably.

Gotcha 9: Controller Moves Mouse Cursor / "NO CONTROLLER" in Games (e.g. Hollow Knight)

- **Symptom:** Controller navigation works in Steam Big Picture, but when launching a game (e.g., *Hollow Knight*), no gamepad buttons work, the right stick or touchpad moves the desktop mouse cursor, and the in-game controller menu displays "NO CONTROLLER".
- **Root Cause:**
 1. Gamescope was run without the `-e` / `--steam` flag, preventing Gamescope from relaying Wayland window focus state back to Steam.
 2. For games flagged in Steam's database with "Native Controller Support", Steam defaults Steam Input to **Disabled** for that specific game (`Opted-in Controller Mask: 1000`). When Steam Input is disabled or window focus is ambiguous, Steam drops the controller into **Desktop Layout** (`PollState Changed from 2 to 1`), translating physical buttons into mouse clicks and keyboard keys.
- **Fix:**
 1. Ensure Gamescope is executed with `-e` (`gamescope -e --backend drm ...`) in the launch script.
 2. In Steam Big Picture, navigate to the game page (e.g. *Hollow Knight*) → click the **Controller Icon** (next to Play) → set **Steam Input** to "Enable Steam Input".

3. Under Steam **Settings** → **Controller** → **Desktop Layout** → **Edit**, set the template to "**Gamepad**". This ensures the controller continues sending standard gamepad events even if a game triggers desktop state.

Gotcha 10: Accessing Steam Menu & Quitting Games with PS5 Controller

- **Symptom:** Pressing the physical PlayStation (PS) button does not open the Steam Overlay or allow quitting the game.
- **Explanation & Navigation Shortcuts:**
 - **Client OS Interception:** On most client devices (especially Android TV, Fire TV, and Apple TV), the operating system captures the physical PS/Home button to return to the host OS home launcher rather than forwarding it to Moonlight.
 - **Moonlight Guide Emulation:** Moonlight allows sending the Steam Guide command to the host:
 - **Hold** `Start / Options` **for 1.5 - 2 seconds**, OR
 - Press `Share / Select + Options / Start` simultaneously.
 - **Steam Chord Shortcut to Force Quit:** Once Steam receives the Guide input, hold **Guide + B** for 3 seconds to force-kill the currently running game and return directly to Steam Big Picture.
 - **Moonlight Stream Disconnect / Exit:**
 - On Android TV / Shield: Double-tap or hold **Start/Options** to trigger Moonlight's on-screen menu overlay, where you can select "Quit Current Session".
 - On Desktop Moonlight: Press `Ctrl + Alt + Shift + Q` to disconnect.

Revision #1

Created 2026-09-21 02:00:10 UTC by luKe

Updated 2026-09-21 21:22:59 UTC by luKe