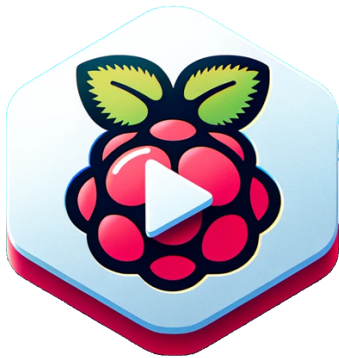


Raspberry Pi Media Player from SKAARHOJ

This page aims to assist you in getting started with SKAARHOJ's Raspberry Pi-based media player. This isn't a product for sale; It's offered **free of charge** with no guarantee of fitness for any particular purpose. Under these conditions, you're free to use it as you wish. Internally, we utilize it as a cost-effective solution for playing back video and image content during demonstrations.



Features

The media player is designed to operate on a standard off-the-shelf Raspberry Pi 4 or 5 as a binary application. It utilizes common applications like ffplay, ffmpeg, ffprobe (which are standard on the Pi), as well as VLC and mplayer in certain instances.

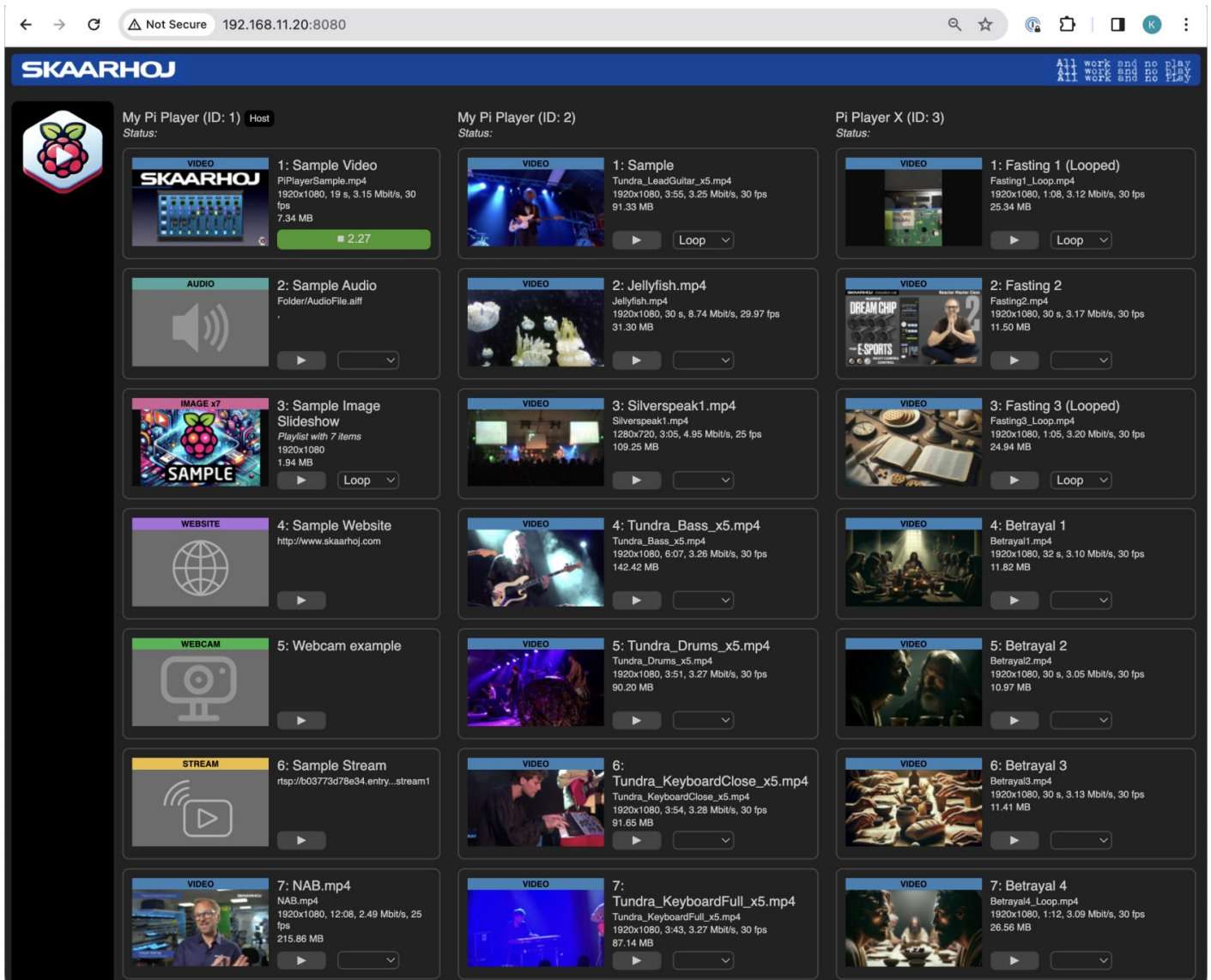
Features include:

- Full-screen playback of video, audio, and image files.
- Options for looping, exiting, or freezing after playback.
- Ability to create playlists for videos, audio, or images using VLC.
- Full-screen display of rtsp streams and webpages.
- Full-screen webcam viewing.
- Remote control through a web browser on port 8080, capable of controlling similar devices on the network.
- Compatibility with SKAARHOJ panels for control via the built-in device core interface.
- Media accessibility through a simple USB key.
- Autoplay feature upon startup.
- Configuration settings stored in a JSON file.
- Extraction of meta data and thumbnails.
- Using a locally connected Stream Deck for media selection

The media player is intended to connect to a screen or video processor system through its HDMI-1 output.

Web UI

The web UI of the player is depicted in the screenshots below. The first screenshot displays three columns, each representing a player found on the network, identified by distinct IDs and names. The first player (ID: 1) also hosts the web view, although in reality, all players display the same interface. When a new player is added to the network, the existing players will detect it and include it in this web UI.



Here's a closer look at just one of the players (the host player). The media player automatically generates thumbnails when possible, but also permits you to specify your own file for use as a thumbnail if desired. It extracts relevant metadata from the files and displays this information too. During playback, you will often see a counter indicating the current position of the player.



My Pi Player (ID: 1) Host

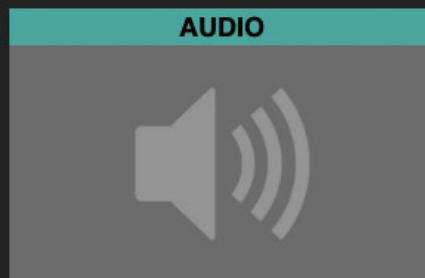
Status:



1: Sample Video

PiPlayerSample.mp4
1920x1080, 19 s, 3.15 Mbit/s, 30
fps
7.34 MB

■ 13.10



2: Sample Audio

Folder/AudioFile.aiff



3: Sample Image Slideshow

Playlist with 7 items
1920x1080
1.94 MB



Loop ▾



4: Sample Website

<http://www.skaarhoj.com>

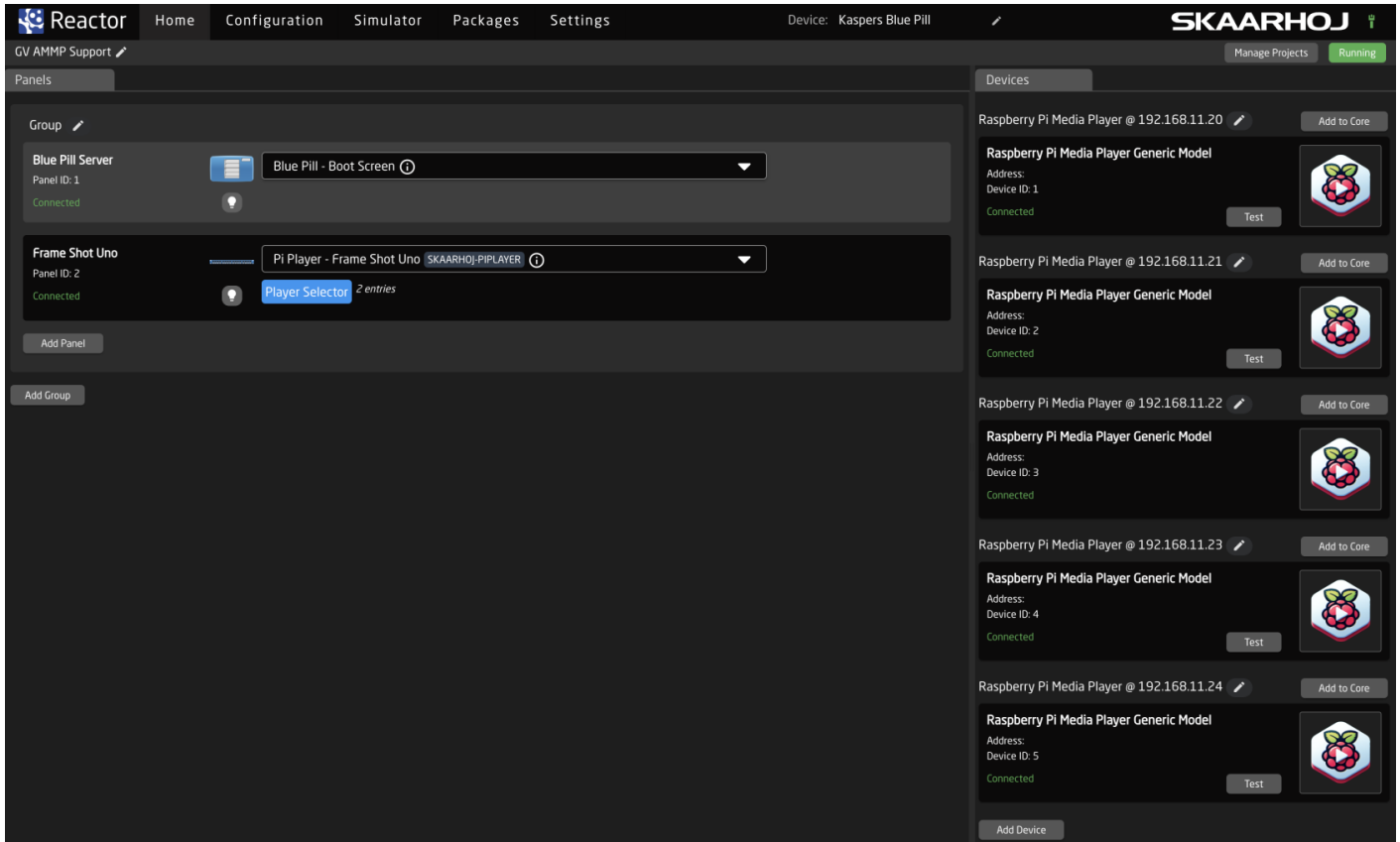


5: Webcam example



Device Core API

The player is equipped with a Device Core API, signifying that panels from SKAARHOJ, specifically those utilizing the Reactor application, will inherently support control over the media player. This integration allows users to seamlessly manage playback, and other functionalities of the media player directly from their SKAARHOJ panels. The Device Core API facilitates a direct communication channel between the media player and the panels, ensuring a smooth and responsive user experience.

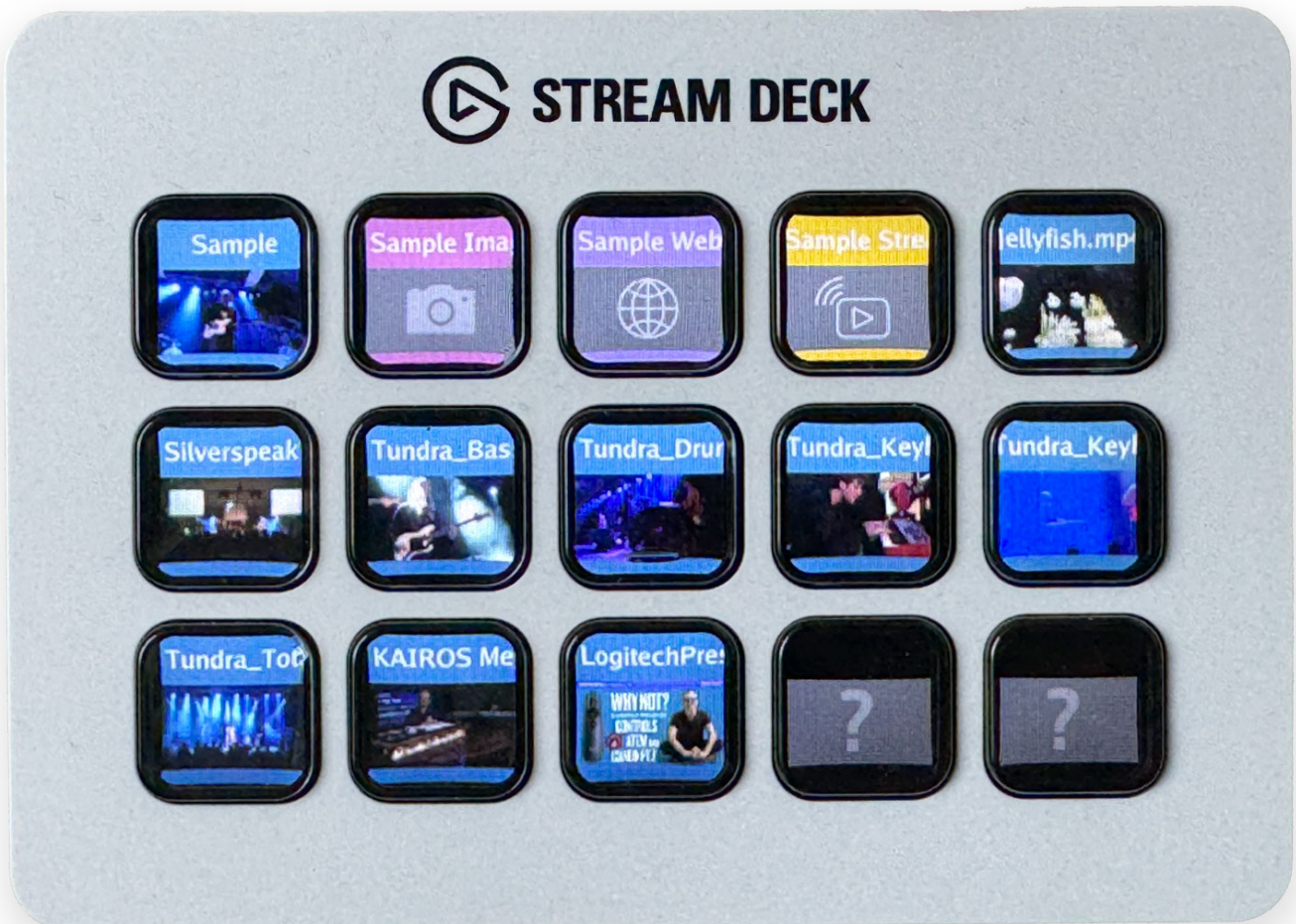


Incorporating panels like Frame Shot Pro or Frame Shot Uno extends the functionality to include displaying thumbnails of the media files available. This feature enriches the user interface by providing visual cues about the content, enhancing the ease of media selection. Such a capability is especially useful in environments where quick identification of media is crucial, allowing users to efficiently manage and control playback directly from these panels.



Stream Deck

For local adhoc playback of media, it's also possible to attach a Stream Deck panel (most models will work) directly to the Raspberry Pi.



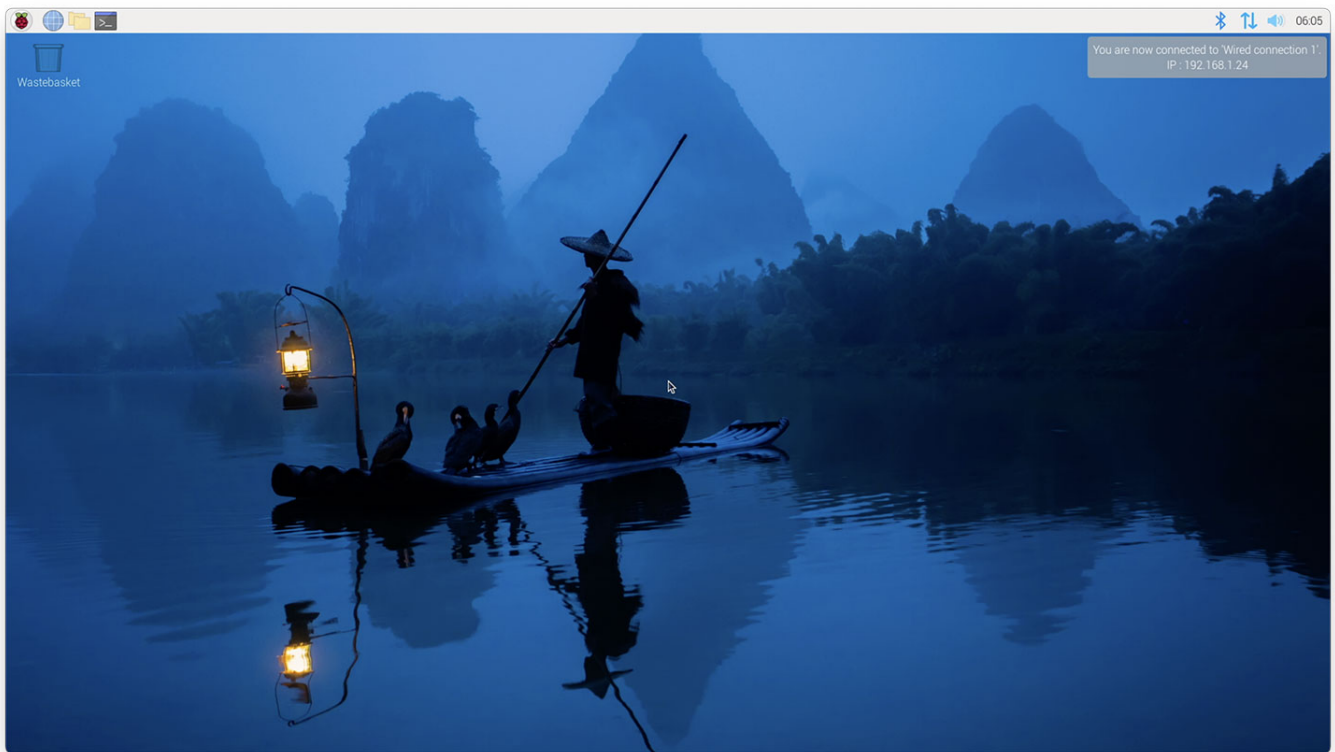
Installation

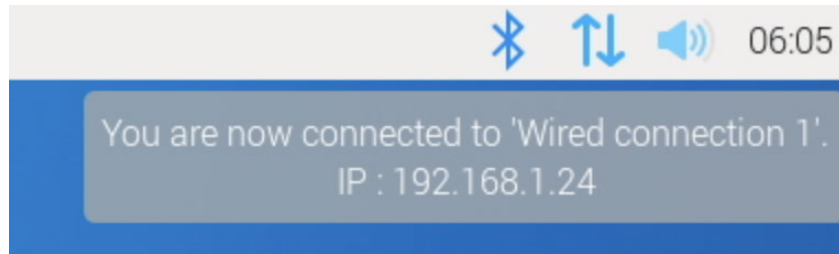
This guide is applicable to both Raspberry Pi 4 and 5 models, utilizing the Raspberry Pi OS available in either 32-bit or 64-bit versions. Detailed instructions for installation can be found in a separate article focused on Densitron screens. [Please refer to that article](#) and follow it up to the section titled 'Preparing the Operating System to run the Densitron screen over USB.' From that point onward, continue with the instructions provided below instead.

Preparing the Operating System to run the SKAARHOJ Media Player

- Insert the microSD card into the Raspberry Pi
- Connect an HDMI monitor to Micro-HDMI plug closest to the USB-C plug on the Pi (HDMI-1)
- Insert a newly formatted USB key (MS-DOS FAT32) into one of the USB slots of the Pi
- Connect the Ethernet jack to your network
- Connect your power supply to boot it up.

You should see it boot up on the HDMI monitor. If you just flashed the memory card, it will automatically reboot a few times before it finally settles with a standard desktop. It's important to stay alert because on the final desktop you will see the DHCP provided IP address shown in the upper right corner. You need that in the next step.





- On your computer, open a terminal (such as Putty on Windows) to connect with SSH to the Pi. On a Mac it would look like this:

```
kasper — admin@raspberrypi: ~ — ssh admin@192.168.1.24...
[~ > ssh admin@192.168.1.24 x 255 2m 30s ]
[admin@192.168.1.24's password: ]
Linux raspberrypi 6.1.0-rpi7-rpi-v8 #1 SMP PREEMPT Debian 1:6.1.63-1+rpt1 (2023-11-24) aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Dec 5 06:05:39 2023
admin@raspberrypi:~ $
```

- Here, you need to use the username ("admin" in this case) and password that you selected when customizing the OS earlier. If you have issues with logging in, you may need to attach a mouse and keyboard to your Pi and change password using the UI (Preferences > Raspberry Pi Configuration)

Windows Server

At the time of writing, the latest Raspberry Pi OS (codenamed "Bookworm") utilizes a display server called Wayland. While Wayland offers its own set of advantages, it is not particularly suited for this media player, primarily because managing screen resolution via the command line can be challenging. The Pi Player requires control over the screen resolution, necessitating a switch to the previous X11 window manager. Fortunately, this is an easy process:

1. Type "sudo raspi-config" into the terminal.
2. In the UI that appears, select "Advanced Options", then choose "Wayland".
3. Here, you'll have the option to select X11 instead of Wayland. Do so and confirm your choice.

4. Exit the configuration tool and reboot the Pi.

After rebooting, if you execute the command "DISPLAY=:0 xrandr", you should see information similar to the following:

```
Screen 0: minimum 320 x 200, current 1920 x 1080, maximum 7680 x 7680
HDMI-1 connected primary 1920x1080+0+0 (normal left inverted right x axis y axis) 708mm x 398mm
  1920x1080  60.00*+  50.00  50.00  59.94  30.00  25.00  24.00  29.97  23.98
  1920x1080i  60.00  50.00  59.94
  1280x720   60.00  50.00  59.94
HDMI-2 disconnected (normal left inverted right x axis y axis)
```

This indicates that HDMI-1 is connected to a monitor or other HDMI input which supports a resolution of 1920x1080. This looks promising.

Customizing the Desktop and Hiding the Menu Bar

Next, you might want to connect a keyboard and mouse to customize the desktop. By right-clicking on the desktop, you can remove or change its background image, as well as disable the display of the Trash bin and other icons. Doing so can be advantageous, especially if you prefer a clean desktop appearance when no media is being played.

Additionally, you might find it useful to set the taskbar to auto-hide, so it's not visible by default. This adjustment ensures a more immersive viewing experience by minimizing on-screen distractions. Here's how to auto-hide the taskbar:

```
sudo nano .config/wf-panel-pi.ini
```

In this file, add these two lines and save:

```
autohide=true
autohide_duration=500
```

Important Notice And Tip

It's important to ensure that a display is connected and powered on when the Raspberry Pi boots up. If not, the window system might fail to accurately detect the available screen resolutions, potentially leaving the HDMI output blank. This requirement can be quite frustrating, but it appears to be a limitation not easy to circumvent. Therefore, it's crucial to be diligent about keeping a display attached and turned on at all times to avoid these issues.

mplayer (for WebCam)

If you want to view Webcams in full screen, please install mplayer:

```
sudo apt update
```

```
sudo apt install mplayer
```

Static IP address

Another aspect that can be incredibly frustrating with a system supposedly designed to evolve, like the Raspberry Pi OS, is the increasing difficulty in setting a static IP address. If you've made it this far, you might have managed to configure IP using DHCP, but there are instances where setting a static IP address is more practical. This is especially true if you need a network configuration that works even in minimal setups, such as directly between the player and your laptop, without any additional network equipment.

After extensive searching, I discovered a command that effectively sets a static IP address:

```
sudo nmcli connection modify Wired\ connection\ 1 ipv4.method manual ipv4.addresses 192.168.11.22/23  
ipv4.gateway 192.168.10.1 ipv4.dns "8.8.8.8, 8.8.4.4"
```

In the example above the IP address is set to "192.168.11.22" and you should replace it with your own. The subnet mask is "255.255.254.0" indicated by "/23" following the IP address. In many cases, the subnet mask is "255.255.255.0"; if this applies to you, use "/24" instead. In the example provided, the gateway is "192.168.10.1"; ensure this matches your gateway.

(Source: <https://forums.raspberrypi.com/viewtopic.php?t=358069>)

You are advised to reboot the device entirely after this step.

Installing *core-skaarhoj-piplayer*

- In the terminal, run the following command to download the SKAARHOJ Media Player:

```
curl -O https://raw.githubusercontent.com/SKAARHOJ/Support/master/Files/PiPlayer/core-skaarhoj-piplayer.zip
```

The above download is generic for both 32 bit and 64 bit systems, but does not include support for Stream Deck! For Stream Deck usage you need the 64 bit version and the file is located in the same repository location, but named "core-skaarhoj-piplayer_arm64bit.zip"

- Next, we need to extract the archive we have just downloaded:

```
unzip core-skaarhoj-piplayer.zip
```

- Insert an empty USB stick and try to run it:

```
./core-skaarhoj-piplayer
```

You should see an output more or less like this:

```
INFO[0000] core-skaarhoj-piplayer started, version v1.0.0-pre1 (9e3d1a8) module=main
INFO[0000] Loading config module=main
INFO[0000] Set up discovery of other media players on the network module=main
INFO[0000] mDNS services initialized module=main
INFO[0000] Web frontend is available on http://192.168.11.23:8080 module=main
INFO[0000] Connected to XWindows Server module=main
INFO[0000] Disabling screen saver module=main
```

Setting up Auto Start of *core-skaarhoj-piplayer*:

It can be started by *systemd* creating a config file, *piplayer.service*, in */etc/systemd/system/* with the following contents:

```
sudo nano /etc/systemd/system/piplayer.service
```

```
[Unit]
Description=SKAARHOJ Pi Player

[Service]
ExecStart=/home/admin/core-skaarhoj-piplayer
Restart=always
User=admin
Group=admin
WorkingDirectory=/home/admin

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

The above config assumes you named your user "admin" and keep the binary in the home directory.

After creating this file, you must reload *systemd* and set it to auto start on every boot:

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable piplayer.service
```

Restart if needed with this:

```
sudo systemctl restart piplayer.service
```

Check status like this:

```
sudo systemctl status piplayer.service
```

Follow log messages from the service like this:

```
journalctl -u piplayer.service -f
```

Configuration

Hopefully, you've managed to set everything up as described above. All system-related settings can be tailored to your individual preferences. The goal is to use a standard, plain vanilla Raspberry Pi so you can rely on external sources for further customization.

The media player application is named `core-skaarhoj-piplayer`.

USB key with Media Files

To run `core-skaarhoj-piplayer`, a USB key is required. The application will attempt to mount `/dev/sda1` and, if not found, `/dev/sdb1`. It will keep looping until a successful mount.

Once mounted, it searches for a `playerconfig.json` file on the drive. If found, this file is used for configuration. If the "AutoUpdate" flag is set internally, the application will automatically scan the USB drive for media files in the root directory and add them to the `playerconfig.json` file.

If the USB key is empty, the Pi Player not only creates a `playerconfig.json` file with sample contents but also adds a sample video file, a thumbnail, and a small slideshow as examples. These samples help confirm that your installation is successful and demonstrate how the `playerconfig.json` file functions.

Should the USB stick contain only a single media file, such as a video, the Pi Player will use that file to create a new `playerconfig.json` file on the USB drive, setting it up for autoplay.

In summary:

- For a quick start, insert an empty USB drive. The Pi Player will write sample files to it for exploration and learning.
- To play a video in a loop on boot, simply place that file on the USB key. A configuration for looped autoplay will be created.

playerconfig.json

This file contains the player's configuration. It's advised to let the Pi Player generate this file initially to understand the example structure. Many fields are self-explanatory. Here are some tips:

- Set the `DeviceId` field to your desired device ID for device core integration. This helps organize players in the Web UI, even if you're not using it with Reactor.
- The `Name` and `Description` fields help identify your players in Web UIs.
- `AutoPlayMediaFile` specifies the media entry from the Media array to be automatically loaded and played on boot.
- Setting the `AutoUpdate` flag to true enables automatic scanning for new media files on each boot, adding unmatched files to the configuration.
- Setting the `Tag` field will group mediaplayers together in the way they discover each other in the UI. Using "*" will show all players.

The `Media` array in `playerconfig.json` is an ordered list of known media, influencing their identification in Reactor and the Web UI.

- The `Ignore` field can be set to skip certain media files.
- The `Type` field specifies the entry's format (Video, Audio, Image, Webpage, Webcam, Stream).
- `Thumbnail` specifies a dedicated thumbnail image file.
- `MediaReference` is an array of strings representing media file names. Multiple entries enable playlist functionality, while a single entry defaults to the first file unless `MediaIndex` specifies otherwise.

metadatabcache.cache

This JSON file contains metadata and thumbnail images generated by the Pi Player. Metadata and thumbnails are processed once, updating only if the media file's modification time changes.

- To regenerate all metadata, simply delete this file.

Note: Generating metadata and thumbnails for new files can delay player startup, as processing new media on a USB key may take minutes.

Troubleshooting and tips

- Use a properly cooled Pi. Prolonged video playback can heat the device to levels where it throttles and reduces clock frequency, impairing playback performance.
- If both WiFi and Ethernet are enabled simultaneously, the Web UI may display both IP addresses, resulting in duplicate columns for that player in the Web UI.
- For your information, it will utilize `/dev/video0` for the webcam functionality.
- **Debugging:** If you encounter issues with the Pi Player, you can enable more detailed logging with the command: `LOG=debug,ln ./core-skaarhoj-piplayer`. However, before running this command manually, make sure to stop the service using `systemctl` if it's set up as one.
- Note that an empty `core-skaarhoj-piplayer.toml` file could potentially cause problems. If this occurs, the file can be safely deleted to resolve the issue.

Known Issues

- Probably Obsolete: It has been frequently noted that during long times of playback, the USB key may simply be "lost" on the system so that it won't even appear as a device under `/dev/sd*`. Requires a reboot.

Revision #17

Created 30 March 2024 10:24:01 by Kasper

Updated 10 February 2026 12:18:39 by Kasper