

HoloQ Player - User Manual

HoloQ Technologies

2026

Contents

HoloQ Player — User Manual	1
1. What is HoloQ Player?	1
2. How to Use	1
3. Interface Guide	2
4. Setting Up Your Content (USB Drive or TF Card)	5
5. A Simple Example Playlist	7
6. Simple Configuration Changes	7
Where to Learn More	8

HoloQ Player — User Manual

A simple guide to setting up and using your HoloQ Player.

1. What is HoloQ Player?

HoloQ Player is a small, dedicated device that automatically plays your videos and images, in whatever sequence you set up, every time it's powered on. No remote control, no menus to navigate — just plug it in and it starts playing your content. Most people set their playlist up to loop continuously, but that's a choice you make when building the playlist — not something HoloQ Player forces on you.

If you'd like it to respond to a button, a sensor, or a computer — for example, switching to a different video when a visitor presses a button — HoloQ Player can do that too, by connecting a simple wire to a controller device. More on that in Section 2.

2. How to Use

2.1 Basic use (no controller needed)

This is all you need for a simple, automatic display:

1. Prepare your content on a USB drive (see Section 4).
2. Plug the USB drive into the board.
3. Connect a screen via the HDMI port.
4. Connect power.

That's it — HoloQ Player boots up and starts playing your playlist automatically, following whatever sequence you set up in `playlist.h1q` (Section 5), for as long as it's powered on.

2.2 Connecting a controller (PC, Arduino, Raspberry Pi, or any

serial device)

If you want your HoloQ Player to respond to something in the outside world — a button press, a sensor, a show-control system, or a computer program — you can connect **any device capable of sending simple serial data** to the board's **UART connector** (see Section 3). This includes:

- An **Arduino** (or similar microcontroller)
- A **Raspberry Pi**
- A **PC** (using a USB-to-serial adapter)
- Really, **any device** with a serial port

Once connected, that device can send simple, single-byte signals to tell HoloQ Player things like “jump to this scene now,” “play this clip in reverse,” or “pause.” HoloQ Player also sends signals back — for example, telling the controller when a video starts or ends — so the two devices can stay in sync.

By default, HoloQ Player responds to controller commands right away — no setup or handshake required. If you're building a permanent installation and want extra protection against a controller's own power-up noise being mistaken for a command, HoloQ Player also supports an optional one-time “introduction” step your controller can be programmed to perform first; it's off by default and most setups don't need it. If you do want it, or if you're programming the controller side yourself, full details are in the **HoloQ Player Serial Protocol Guide**.

This manual keeps things at a “how do I connect it” level. For the full, precise technical details of every signal HoloQ Player sends and listens for — the exact values, timing, and behavior — please see the separate **HoloQ Player Serial Protocol Guide**. That document is written specifically for whoever is programming the controller device (the Arduino/PC/Raspberry Pi side), and covers everything needed to build that side of the connection.

3. Interface Guide

Here's what everything on the board does.

3.1 Front of the board

Label	What it's for
Ethernet	Wired network connection (if your setup needs one)
USB 2.0	Two USB ports — for a keyboard/mouse during setup, or other USB accessories
HDMI out	Connect your screen/display here
Power (Type-C, 5V)	Connect the power supply here to turn the board on
Analog audio out	Standard headphone-style audio jack, for speakers/audio equipment that doesn't use HDMI audio
UART	The connector used to attach a controller device (PC/Arduino/Raspberry Pi) — see Section 2.2
Reset	Physical reset button

The green add-on board on the right (labeled **HoloQ-1**) is what provides the UART connector — this is the one you'll use to connect a controller device. Its four pins are labeled right on the board: **VCC 3V3**, **GND**, **TX**, and **RX**.

Here's the same board without labels, for a cleaner reference photo:

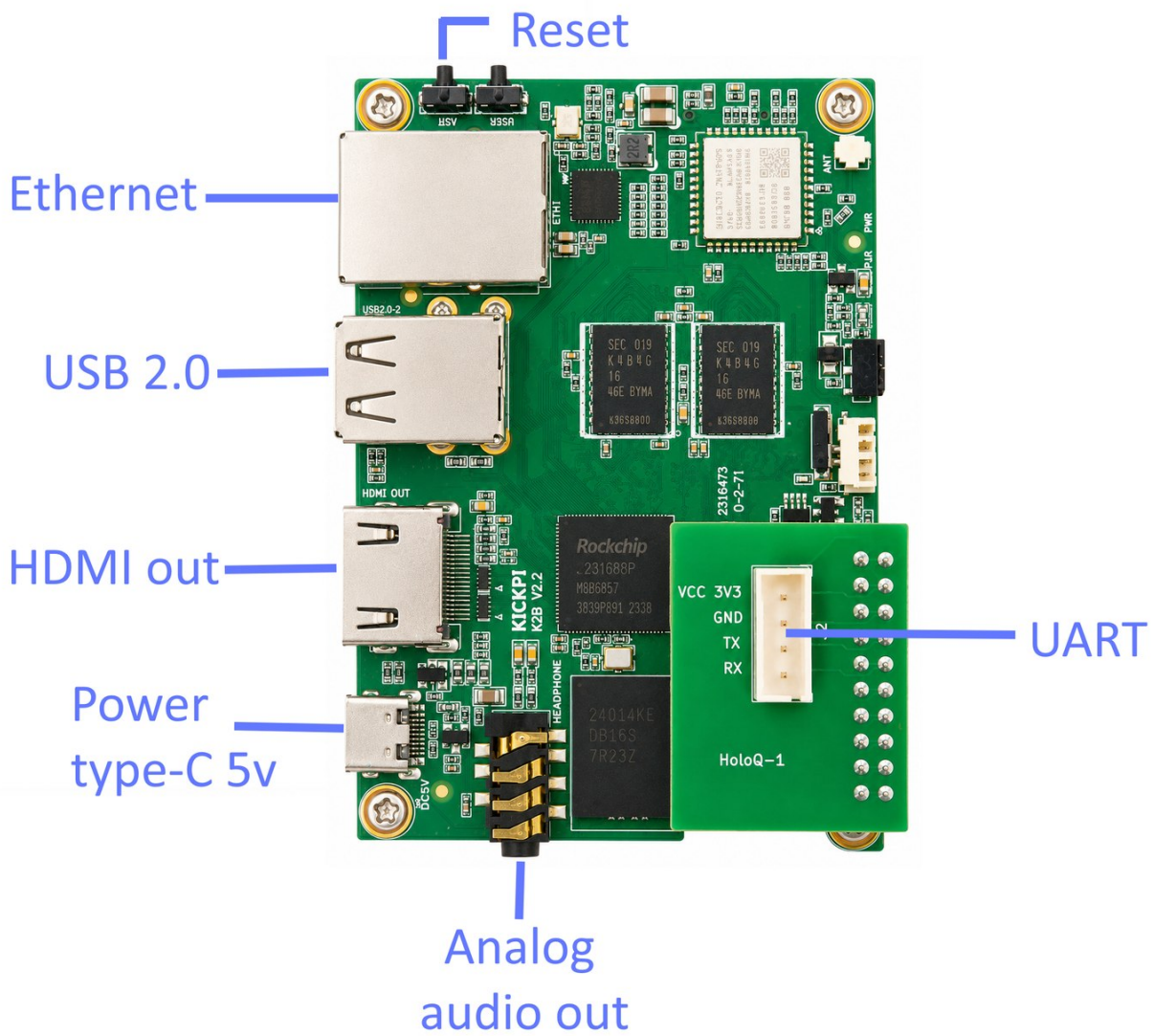


Figure 1: Front view of the HoloQ Player board with labeled connectors

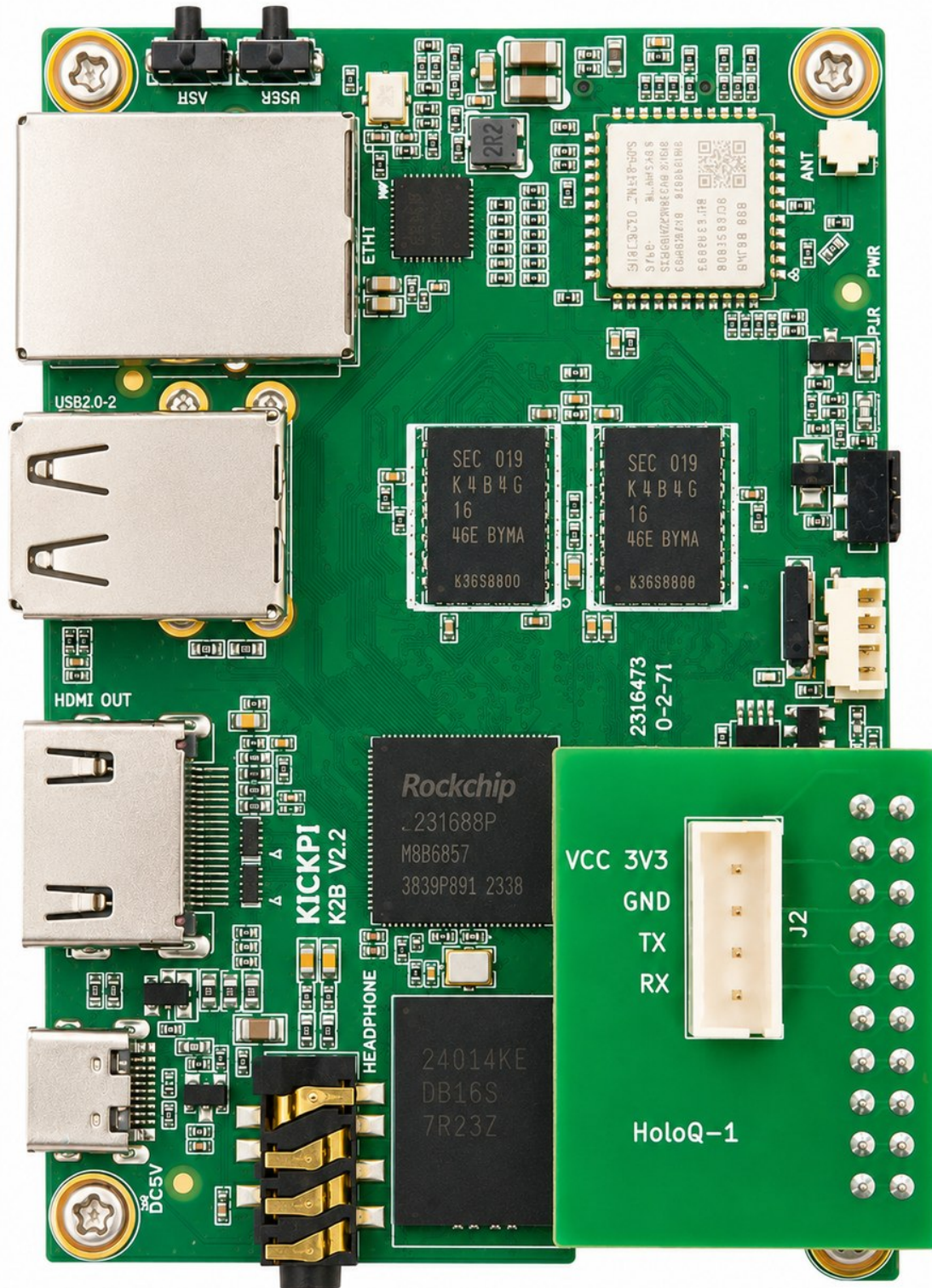


Figure 2: Front view of the HoloQ Player board, unlabeled

3.2 Bottom of the board

Label	What it's for
TF card	A microSD card slot — can be used the same way as a USB drive, to store your media files and <code>playlist.hlq</code> . Avoid removing it while the board is running.
Debug (top right, marked RX/TX/GND)	A separate, advanced debug connector used for low-level troubleshooting. This is not the same connector used for normal controller connections — for connecting a PC/Arduino/Raspberry Pi, use the UART connector on the front of the board (Section 3.1), not this one.
LED	Status indicator light

4. Setting Up Your Content (USB Drive or TF Card)

HoloQ Player reads your videos, images, and playlist from removable storage — either a USB drive plugged into the board's USB port, or a microSD (TF) card in the board's TF card slot (Section 3.2). Both work the same way; use whichever is more convenient for your setup.

4.1 Folder structure

On your USB drive, set things up like this:

```
(root of USB drive)
├── images/
│   └── (your image files go here)
├── videos/
│   └── (your video files go here)
├── playlist.hlq      <- required: tells the player what to play, and when
└── config.json      <- optional: only needed if you want to change a setting
```

- **images/** — put all your image files here (e.g. `.jpg`, `.png`)
- **videos/** — put all your video files here (e.g. `.mp4`)
- **playlist.hlq** — a text file that lists your content and the order it should play in (see Section 5 for a simple example)
- **config.json** — completely optional. If you don't include one, HoloQ Player uses sensible defaults for everything. Only add this file if you need to change a specific setting (see Section 6)

Recommended format: FAT32 gives the broadest, most reliable compatibility on either a USB drive or a TF card. Once it's set up, just plug it into the board's USB port or TF card slot.

4.2 Video walkthrough

For a full step-by-step demonstration of setting up a USB drive from scratch, watch this video:

[INSERT YOUTUBE LINK HERE]

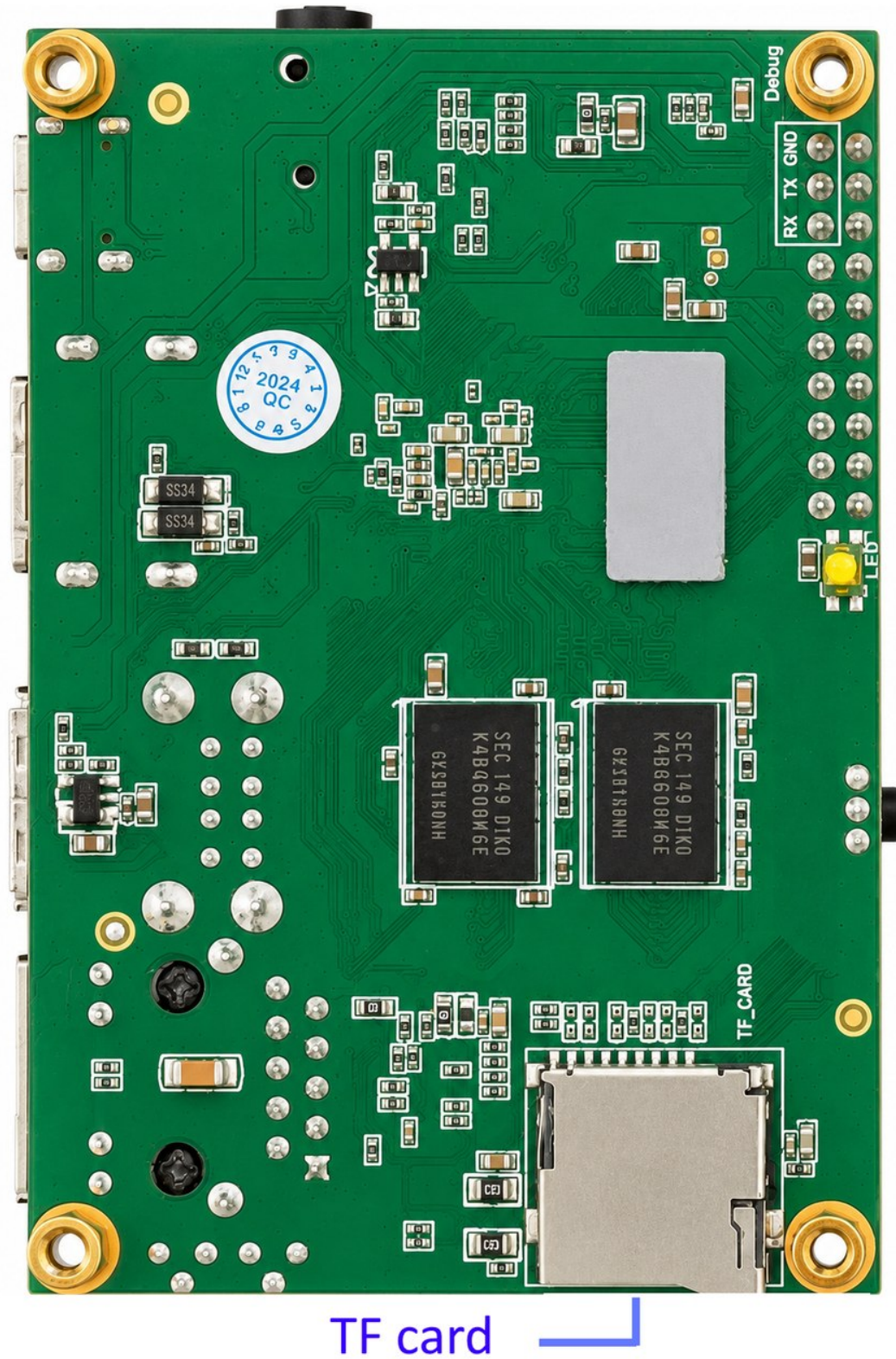


Figure 3: Bottom view of the HoloQ Player board

5. A Simple Example Playlist

Playlists don't have to loop — you can build any sequence you like, and whether it loops back to the beginning, stops on a final item, or does something else entirely is entirely up to you. Let's look at one common example: **play a video, then show an image, then play a second video, then loop back to the beginning and repeat.**

Here's a `playlist.hq` file that does exactly that:

```
{
  "play_start": "video1.mp4",
  "media_path": ".",
  "playlist": [
    {
      "type": "video",
      "file": "videos/video1.mp4",
      "next": "photo1.jpg"
    },
    {
      "type": "image",
      "file": "images/photo1.jpg",
      "timeout": 5,
      "next": "video2.mp4"
    },
    {
      "type": "video",
      "file": "videos/video2.mp4",
      "next": "video1.mp4"
    }
  ]
}
```

What this does: 1. Starts by playing `video1.mp4`. 2. When it finishes, shows `photo1.jpg` for 5 seconds. 3. Then plays `video2.mp4`. 4. When that finishes, it goes back to `video1.mp4` — because that item's "next" explicitly says so — and the whole sequence repeats.

Each item says what to play, and "next" says what should play right after it. The looping in this example isn't automatic — it happens because the last item's "next" was deliberately set back to the first item's file. If you left "next" off that last item instead, it would **not** loop back to the beginning — it would simply hold on that last item (repeating/re-displaying just that one item) forever, and the sequence would never return to `video1.mp4`. That's a perfectly valid choice too — for example, ending on a final "thank you" screen that just stays there, rather than restarting the whole show.

In short: whether (and how) your playlist loops is entirely controlled by how you set "next" on each item — HoloQ Player doesn't assume looping by default.

This is intentionally the simplest possible example. `playlist.hq` can do much more — interactive jump points triggered by a controller, timed cues, reverse-playback effects, and more. For the full guide to building richer playlists (ideally with a visual authoring tool rather than hand-writing JSON), please refer to the **HoloQ-Author User Manual**.

6. Simple Configuration Changes

Most people never need to touch `config.json` at all — HoloQ Player works great with sensible defaults out of the box. But if you do need to change something, here's the most common example: **switching where the audio comes out.**

6.1 Changing audio output (HDMI vs. Analog)

By default, HoloQ Player sends audio out through **HDMI**, along with the video. If you'd rather use the analog audio jack instead (Section 3.1) — for example, to connect separate speakers — add a `config.json` file to your USB drive with this content:

```
{  
  "audio_output": "analog"  
}
```

To send audio out through both HDMI and the analog jack at the same time:

```
{  
  "audio_output": "both"  
}
```

Or to go back to HDMI-only (the default), you can either delete `config.json` entirely, or set:

```
{  
  "audio_output": "hdmi"  
}
```

Just save whichever version you need as `config.json` at the root of your USB drive (alongside `playlist.hlg`), and restart the board for the change to take effect.

6.2 Other settings

`config.json` can control quite a bit more — things like which video output routing options are available, logging, and advanced network/serial settings. For the complete list of everything that can be configured, please refer to the **`config.json` specification document**.

Where to Learn More

This manual covers the basics of connecting and using your HoloQ Player. For anything more advanced, these companion documents go deeper:

Document	What it covers
HoloQ Player Serial Protocol Guide	Full technical detail for programming a controller device (PC/Arduino/Raspberry Pi) to talk to HoloQ Player
HoloQ-Author User Manual	How to build richer playlists — interactivity, timed cues, and more
<code>config.json</code> Specification	The complete reference for every available configuration setting