

gstreamer.rs — Video Streaming

Receives H.264 video from the rover over UDP, decodes it, and serves it as MJPEG over HTTP so the frontend can display it in `<img>` tags.

Pipeline per camera

```
udpsrc (UDP port) → rtpjitterbuffer → rtph264depay → avdec_h264 → videoconvert → jpegenc → appsink
```

Each decoded JPEG frame is placed into a shared `FrameBuffer` (`Arc<Mutex<Option<Bytes>>>`). A separate async HTTP server (using `warp`) reads from that buffer and streams it as `multipart/x-mixed-replace` — the standard MJPEG format.

Port mapping

UDP input port	HTTP output port	Camera
4500	5000	Depth / front camera
4501	5001	Secondary camera
4502	5002	Arm camera

Feed health monitoring

A background task (`watch_feed_health`) polls each stream every 500ms. If no frame has been received within 2 seconds, the stream is considered stale. The backend emits a `camera-feed-status` Tauri event to the frontend with the payload:

```
{ "port": 5000, "stale": true }
```

The frontend listens for this event to show feed status indicators.

For development without rover hardware, run the `fake_camera_gstreamer`. For instructions see [Common Operations](#).