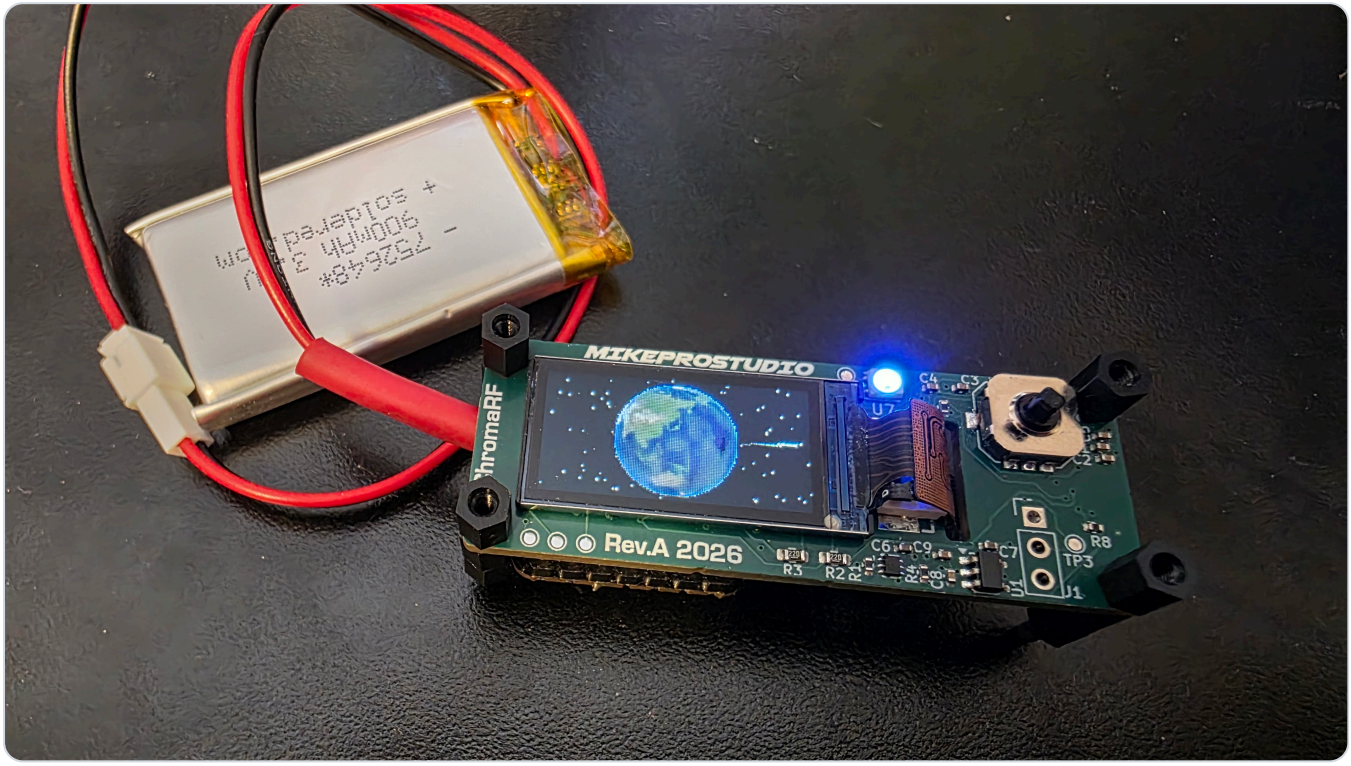
ChromaRF · Project Details

Wireless video streaming onto a 0.96" SPI display — built on a Seeed XIAO ESP32-C6

Michael Neumüller · linkedin.com/in/michael-neumueller1 · neumuellerlabs.ch · 2026

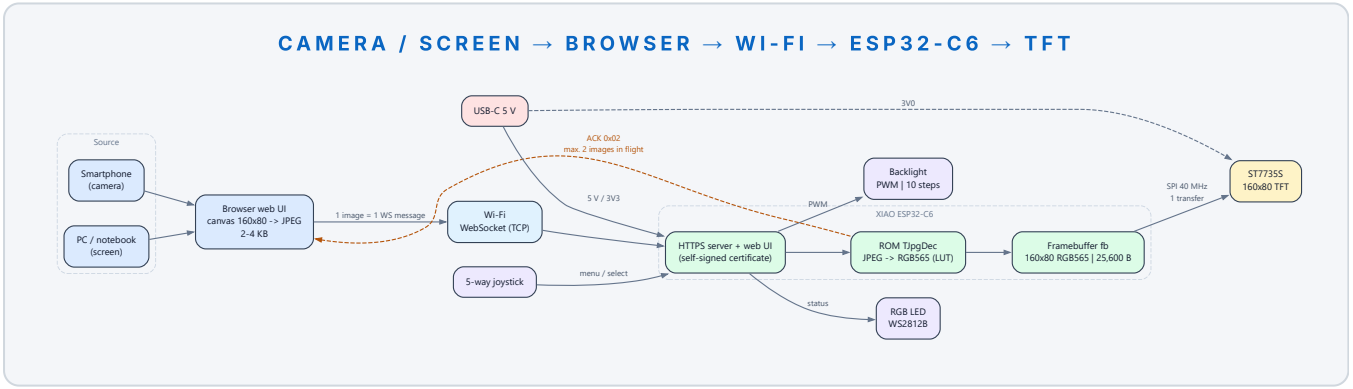


ChromaRF turns **any browser into the video source** and the device into a **self-hosted web UI**. A phone camera or a PC screen is scaled to **160 × 80**, compressed as **JPEG** and streamed over **Wi-Fi / WebSocket** to a Seeed XIAO ESP32-C6, which decodes it **directly in ROM** via TJpgDec and pushes it to a 0.96" ST7735S TFT in a **single SPI transfer**. No app and no dedicated receiver are required — local control, QR onboarding, an animated Earth screensaver and deep sleep are all on-device.

TECHNOLOGY STACK

LAYER	TECHNOLOGY
Builds	Two interchangeable builds in the **SeeedStudio XIAO** form factor: **XIAO ESP32-C6** (Wi-Fi video) and **XIAO nRF52** (true ultra-low-power operation)
MCU / SoC	Seeed XIAO **ESP32-C6** — RISC-V, single core up to 160 MHz, 4 MB flash, Wi-Fi 6 / BLE 5 / 802.15.4 · the nRF52 build runs **Zephyr RTOS**
Display	0.96" **ST7735S**, 160 × 80, SPI @ 40 MHz, RGB565 (65,536 colors)
Video path	Browser canvas → JPEG → WebSocket → **ROM TJpgDec** → RGB565 LUT → SPI
Input	5-way joystick (G-Switch GT-TCA06F-H050-L1), on-screen menu
Feedback	WS2812B status LED (GRB, RMT driver), PWM backlight via EM6M2 dual MOSFET, gamma 2.2
Connectivity	Wi-Fi station + AP fallback, mDNS alias `esp.local`, self-served web UI
Power	USB-C 5 V (VBUS), low-noise TPS7A0230 3.0 V LDO, deep sleep with LP-GPIO wake, LiPo-ready
Toolchain	Arduino / `arduino-cli` (board `XIAO_ESP32C6`, Huge APP partition), Adafruit ST7735 + GFX

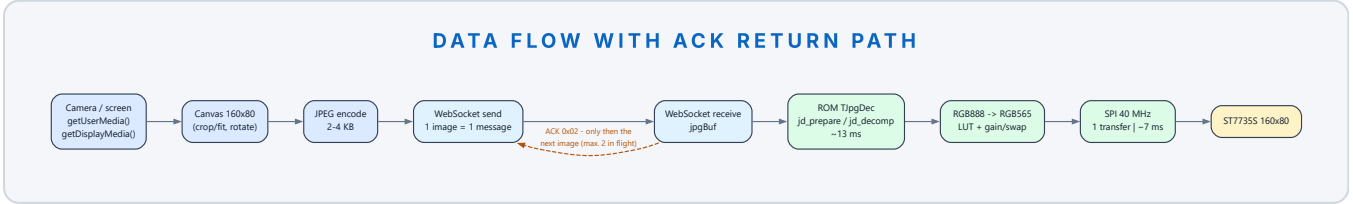
SYSTEM OVERVIEW



VIDEO PIPELINE

ChromaRF is deliberately a **single-frame stream (MJPEG-like)**, not a running video codec. Every frame is produced fresh in the browser, sent as JPEG, decoded on the ESP32 and displayed.

Camera / Screen · Canvas 160×80 · JPEG (2–4 KB) · WebSocket · ROM TJpgDec · RGB565 (LUT) · 1× SPI transfer · ST7735S



Why JPEG instead of real video

APPROACH	PROBLEM ON THE ESP32-C6
H.264 / H.265	No hardware video decoder; software H.264 in real time is impossible.
WebRTC	Browsers cannot send raw UDP; the C6 cannot terminate WebRTC / VP8 / H.264.
Raw RGB565 over TCP	25,600 B/frame > TCP window (5,760 B) → hard ceiling of ~20 fps.
JPEG (MJPEG-like)	**2–4 KB/frame**, fits one TCP window, ROM decoder available → works.

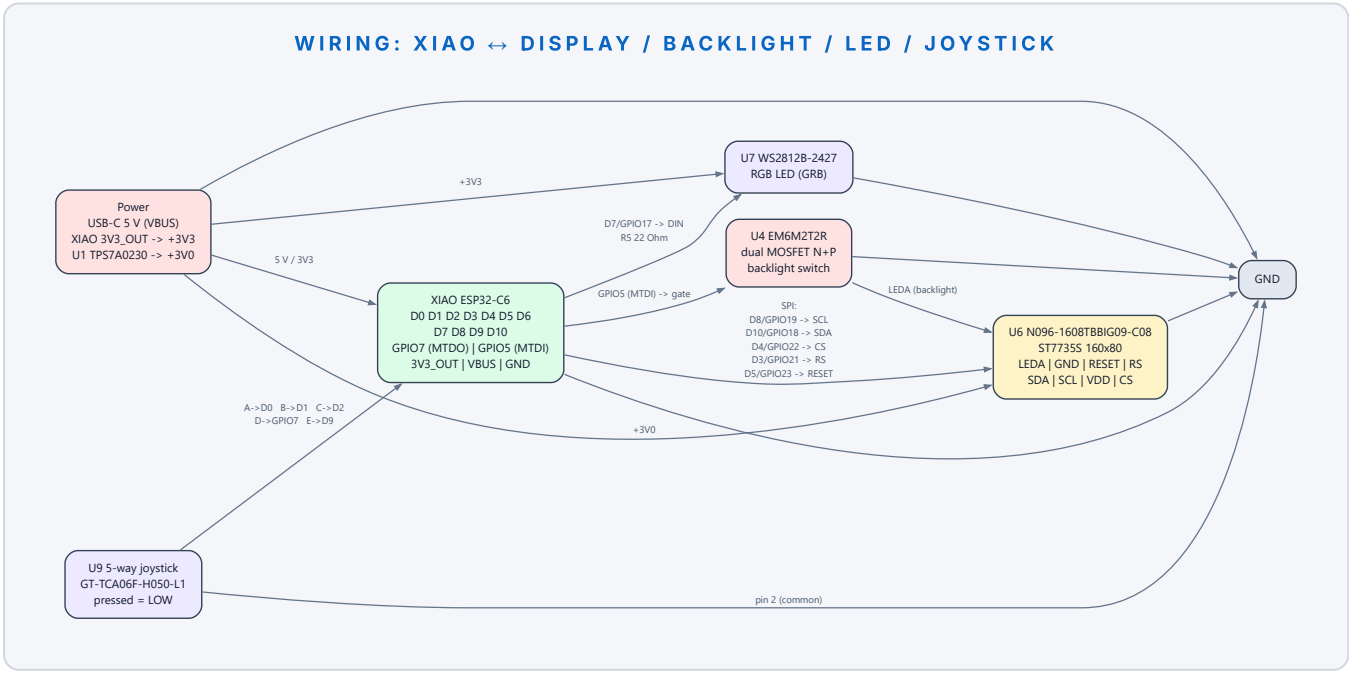
Typical latency budget (1 image in flight)

STAGE	ORDER
Camera / canvas read + JPEG encode (browser)	5–15 ms
Network (RTT)	~11 ms
Decode (`jd_decomp`)	~13 ms
SPI transfer (25,600 B)	~7 ms
Total	**roughly 35–45 ms**

Pacing uses **ACKs, not a timer** (Chrome throttles background-tab timers to ~1 Hz), with end-to-end backpressure limited to `VT_INFLIGHT_MAX = 2` images in flight.

HARDWARE & WIRING

A small carrier board ("ChromaRF XIAO") hosts the ESP32-C6 module, the TFT, the RGB LED, the joystick and the power/interface circuitry. The backlight is not driven directly from a GPIO — an **EM6M2 dual MOSFET** switches it (HIGH at the gate = backlight on).



Key components

REF	PART
U2	XIAO ESP32-C6 module socket
U6	0.96" ST7735S TFT 160 × 80
U7	WS2812B RGB status LED
U9	5-way tilt joystick
U1	TPS7A0230 3.0 V low-noise LDO
U4	EM6M2 dual MOSFET (backlight)

Deep-sleep quiescent current

PATH	ORDER
ESP32-C6 chip	~7 µA
ST7735S + panel	~10–50 µA
WS2812B "black"	~0.6–1 mA
Whole board	**~0.3–1.5 mA**

TWO BUILDS — ESP32 & NRF52 (ULTRA-LOW-POWER)

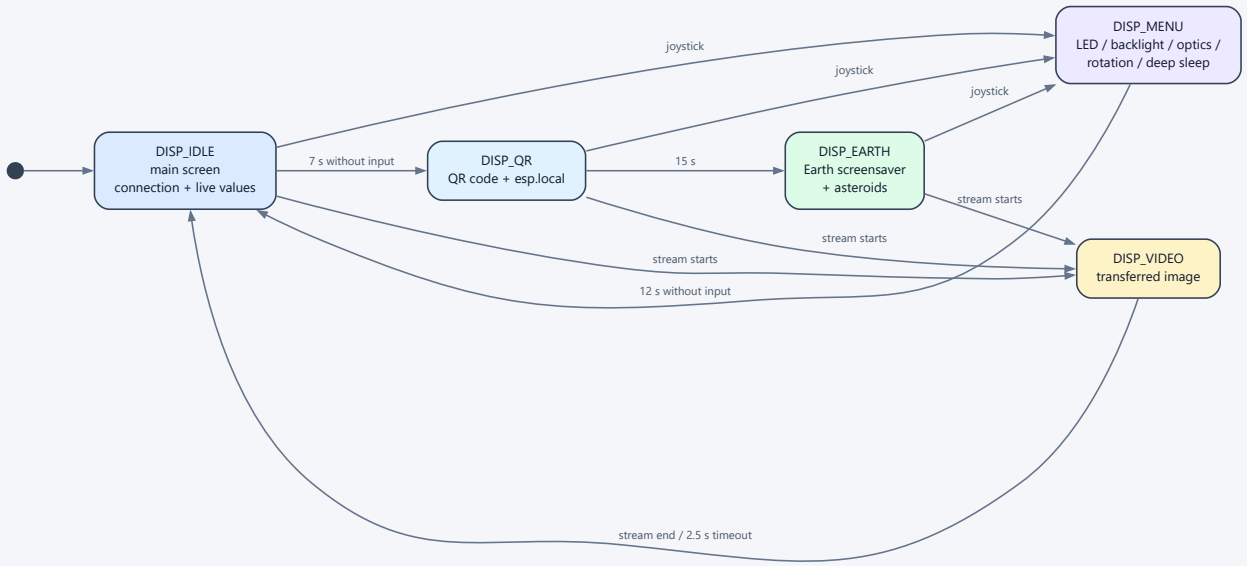
ChromaRF ships as **two hardware builds** on the same board outline. Both are compatible with the **SeeedStudio XIAO** family — same XIAO module footprint, USB-C and castellated pads — so the module can be swapped without changing the carrier board.

BUILD	PURPOSE
XIAO ESP32-C6	Wi-Fi 6 / BLE 5 build for the browser-driven JPEG video stream. Serves the whole web UI itself and decodes video in ROM.
XIAO nRF52	Ultra-low-power build (Zephyr RTOS) for battery operation. It delivers the true **ultra-low-power functionality** that the Wi-Fi path cannot reach: deep sleep, low duty cycle and long runtime.

Because both builds share the XIAO form factor, you pick the SoC that matches the use case: high-throughput streaming on the **ESP32-C6**, or energy-optimised, always-on operation on the **nRF52**.

CONTROLS & DISPLAY STATES

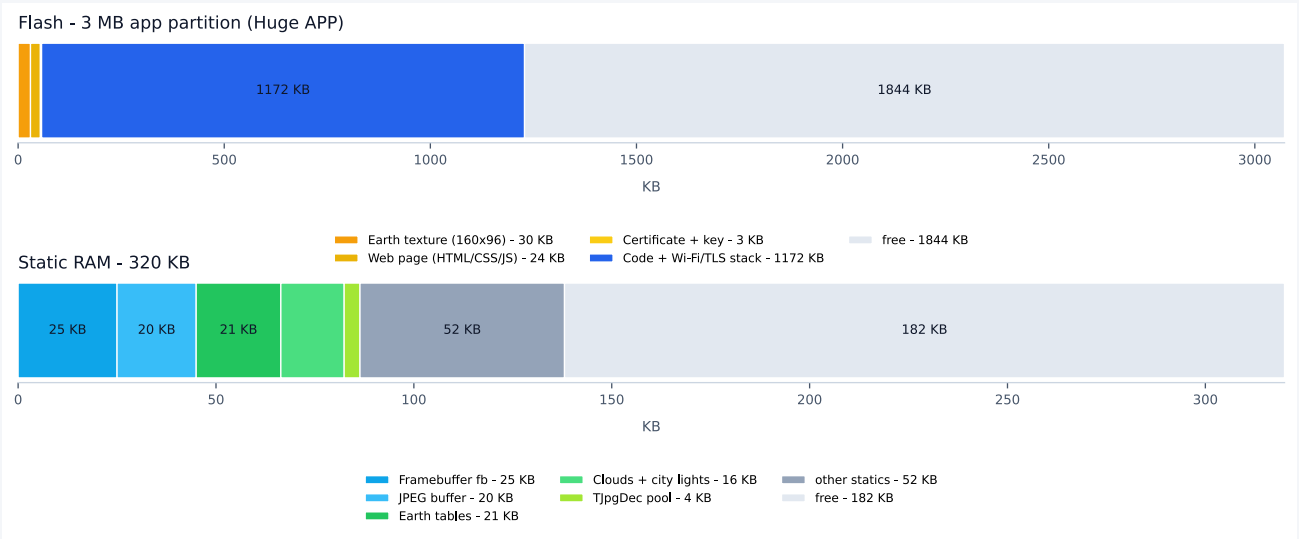
SCREEN STATES: IDLE / QR / EARTH / MENU / VIDEO WITH TIMED TRANSITIONS



- **Joystick:** 5 directions (push / left / right / up / down), debounced, with auto-repeat; also the deep-sleep wake source (LP GPIOs only).
- **Backlight:** 10-step PWM with gamma 2.2 through the EM6M2 MOSFET.
- **RGB LED:** GRB order via the core RMT driver — no extra library, no flash cost.
- **Extras:** QR code for joining, `esp.local` mDNS alias, 180° rotation, runtime color calibration, deep sleep.

RESOURCES & BUILD

FLASH AND RAM BREAKDOWN



Flash usage

ITEM	BYTES
Own app data (Earth tex, HTML, TLS)	~58,299
Sketch / program total	1,257,934
App partition (Huge APP)	3,145,728
Free for future code	**~1.80 MB**

Static RAM usage

BUFFER	BYTES
Framebuffer (160 × 80 × 2)	25,600
JPEG buffer	20,480
Earth tables	21,904
Clouds + city lights	16,384
Total static RAM	**141,376**

Single app slot → **no OTA**; updates over USB. Build with the `Huge APP (3MB No OTA/1MB SPIFFS)` partition scheme.

KEY HIGHLIGHTS

1. **No app, no receiver**
   The browser is the source; the ESP32-C6 serves the whole web UI itself.
2. **ROM JPEG decode**
   TJpgDec decodes in ROM straight to RGB565, written as one SPI transfer.
3. **Local control**
   5-way joystick, on-screen menu, dimmable backlight, WS2812B status LED.
4. **Zero-config access**
   QR onboarding, `esp.local` mDNS alias, AP fallback.
5. **Earth screensaver**
   Animated Earth with clouds, city lights and asteroid field — geometry precomputed at boot.
6. **Clean carrier board**
   TPS7A0230 3.0 V LDO, backlight MOSFET, selectable panel variants.

TECHNICAL DOCUMENTATION INDEX

The full engineering write-up is split into eight parts (available in the project folder):

DOCUMENT	CONTENT
01 — Hardware	Board, pinout, power, display, LED, joystick, schematic netlist
02 — Software architecture	Execution model, data flow, state machines, concurrency
03 — Video pipeline	Camera/screen → JPEG → WebSocket → TJpgDec → RGB565 → SPI
04 — Networking & security	Wi-Fi modes, AP fallback, TLS/HTTPS, HTTP endpoints, WebSocket protocol, mDNS, QR
05 — Controls & menus	Screens, menu, joystick, backlight PWM/gamma, RGB LED, rotation, deep sleep
06 — Earth screensaver	Earth renderer, texture, clouds/city lights, asteroids, QR phase
07 — Resources & build	Flash/RAM budget, partitioning, build commands, tools, optimizations
08 — Bring-up & troubleshooting	Bring-up, self-tests, known failure modes and root causes