



MIPI CSI-2 Camera Bring-up

AN-003 · E1M and E1M-X SoMs

Document Number: AN-003 **Revision:** 0.3 **Date:** July 2026 **Status:** Preliminary

alplab.ai

© 2026 Alp Lab AB. All rights reserved.

1 Scope

This application note covers the carrier-board side of bringing up a MIPI CSI-2 camera: configuring the on-module per-camera LDO, routing the CSI lanes, and building the SDK's `<alp/camera.h>` capture path.

Warning: Live capture is unproven on Alp Lab hardware. No camera sensor has been wired on an Alp Lab bench to date, and the reference example is build-verified only (`build_only` on every platform in its `testcase.yaml`; its README carries an `[UNTESTED]`-on-hardware marker). The hardware guidance below — LDO sizing, lane routing, pin assignment — is datasheet-backed and safe to design against. The software walkthrough is a starting point for your own bring-up, not a reproduced result: budget bench time for first light.

Audience	Firmware engineers integrating a camera sensor into vision workloads.
Prerequisites	QS- guide completed, RPi-compatible MIPI camera module (e.g. RPi V2 or HQ camera).
Outcome	<code>ai-camera-viewer</code> builds and flashes against the <code><alp/camera.h></code> surface. First light on your sensor is your bring-up step.
Time	20 minutes to a flashed image; sensor bring-up is open-ended.
Source	<code>examples/camera-vision/ai-camera-viewer/</code> in <code>alp-sdk</code> .

Table 1 Scope summary

2 Hardware Setup

The table below is **silicon and module hardware**, not a software support matrix. See Section 3 for what the SDK actually drives today.

SoM family	CSI interfaces + LDO
E1M-AEN	1 × 2-lane CSI (CSI0). Per-camera LDO: +V_CAM0 (Z34) + CAM_VFB0 (Z33) divider; a second LDO pair +V_CAM1 (AB34) / CAM_VFB1 (AB33) is available as a general-purpose carrier rail.
E1M-X V2N / V2N-M1	2 × 4-lane CSI (CSI0, CSI1). Per-camera LDOs: +V_CAM0...+V_CAM3 with matching CAM_VFBx feedback divider.

Table 2 MIPI CSI-2 camera interfaces per family

Both SoCs carry an on-die ISP block: the Alif Ensemble ISP-Pico (VeriSilicon ISP Nano “Pico”) on E4 / E6 / E8 — but not on E3 / E5 / E7 — and an Arm Mali™-C55 on the Renesas RZ/V2N. Neither is reachable from the SDK today; Section 3 has the detail.

Place the feedback resistor pair (R1, R2) **close to the SoM CAM_VFBx pin**; long feedback traces invite oscillation. The output voltage is set by:

$$V_{\text{CAM}} = 0.6 \text{ V} \times \left(1 + \frac{R_1}{R_2} \right)$$

3 ISP Status

No ISP offload is available through the Alp SDK today, on any SoM family. The ISP hardware exists; the software path to it does not. Do not choose an E6 over an E5, or an E1M-X over an E1M-AEN, expecting the SDK to hand you a working ISP.

What that means concretely:

- On a default build, `alp_camera_configure_isp()` returns `ALP_ERR_NOSUPPORT`. The portable Zephyr video backend that serves the camera class has no in-line ISP surface.
- The Alif ISP-Pico backend is registered for **E8 only** — not E4 or E6 — and cannot currently be enabled: it is opt-in and default-off, it requires the ISP devicetree node (which ships disabled), and it links a vendor HAL wrapper whose shipped version is API-incompatible with the driver. Widening it to E4 / E6 needs explicit backend registrations plus board validation.

- Its `configure_isp` implementation latches the requested configuration into backend state and returns `ALP_OK` without programming the ISP. The matching parameter upload is pending the Alif ISP-Pico HAL pack.
- The Alif vendor-extension surface (3A windows, per-channel gain tables, lens-shading LUTs) returns `ALP_ERR_NOSUPPORT` from every entry point, for the same reason.

Warning: On V2N silicon the SDK compiles a V2N-specific camera backend by default whose `configure_isp` also latches-and-returns-`ALP_OK` with no register writes. An `ALP_OK` from `alp_camera_configure_isp()` on V2N therefore does **not** mean the ISP was programmed. Do not gate image quality on that return code.

What is available instead: the sensor pipeline — `open` / `start` / `stop` / `capture` / `release` / `close` — routes through Zephyr’s portable video API and is real code, not a stub. Plan on your sensor delivering the pixel format you need directly, and on doing any 3A or tuning work in your application until the HAL pack lands.

4 Software Walkthrough

```
west build -b <BOARD> examples/camera-vision/ai-camera-viewer
west flash
```

The example opens an OV5640 stream at 240×240 RGB565 via `alp_camera_open()`, runs a person-detect model over the frames, and composes the preview plus bounding boxes with LVGL. Two things to know before you build it:

- It ships **without** a model file, to avoid the binary-blob licence question. Compile your own `.tflite` with Arm’s Vela compiler and replace the placeholder buffer in `src/inference_loop.c`.
- Its reference board wiring drives an ST7789 240×320 TFT over SPI — not a DSI panel. See **AN-004** for the display side.

If `alp_camera_open()` fails the example does not abort: it logs `camera open failed`; running with synthetic frame and renders a placeholder preview so the UI layout can be checked against a design mock-up. A running demo is therefore **not** on its own evidence that your sensor is streaming — confirm the log line is absent.

5 Troubleshooting

- **LDO voltage off:** divider current too low; reduce $R1 + R2$ until divider current = $100 \times$ worst-case feedback-pin current (see **DS-AEN-001** §7.8 formula).
- **No CSI link:** differential lanes swapped (P/N reversed) or trace impedance not 100Ω differential.
- **Lane skew errors:** lane-to-lane length mismatch $>$ TBD ps.

6 References

- **Example source:** `examples/camera-vision/ai-camera-viewer/` in `alp-sdk`; API in `include/alp/camera.h`. Silicon staging checks: `examples/aen/aen-camera-regcheck/`, `examples/aen/aen-isp-regcheck/` — the ISP check validates the devicetree **bind** only, with the ISP driver deliberately not enabled, and reports the HAL-wrapper blocker at runtime.
- **Hardware Design Guides:** **HG-AEN-001** §5.5, **HG-V2N-001** §6.4 (MIPI CSI-2 routing rules).
- **Datasheet CSI-2 pin tables:** §2.3 in each datasheet.

7 Revision History

Revision	Changes	Date
0.1	Initial draft.	May 2026
0.2	Updated SDK references to current layout: example path <code>examples/camera-vision/ai-camera-viewer/</code> , added <code>include/alp/camera.h</code> and the <code>aen-camera-regcheck/aen-isp-regcheck</code> reg-checks; corrected the AEN on-die ISP to the Alif Ensemble ISP-Pico (<code>vsi, isp-pico</code>); aligned the stream config with the example (240×240 RGB565).	June 2026
0.3	Corrected the ISP story against <code>alp-sdk</code> source: the ISP-Pico backend registers for E8 only (not an E4/E6/E8 support tier), its <code>Kconfig</code> defaults to <code>n</code> , and the devicetree node is disabled — so no ISP offload is reachable today and the vendor-ext camera surface returns <code>ALP_ERR_NOSUPPORT</code> . Separated the genuine E4/E6/E8 ISP hardware metadata from the support story. Noted that live capture is unproven (no sensor wired on the bench) and that on V2N <code>configure_isp</code> returns <code>ALP_OK</code> without programming the ISP. Removed a fabricated expected-output log. Fixed the Alp SDK repository links (<code>alpDevs</code> → <code>alplabai</code>).	July 2026

Table 3 Revision History