ALP LAB

MIPI CSI-2 Camera Bring-up

AN-003 · E1M and E1M-X SoMs

Document Number: AN-003 **Revision:** 0.7 **Date:** November 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: Bench status (Alp SDK™ v0.17.0). Live capture through `<alp/camera.h>` is bench-verified on E1M-AEN803 (Alif Ensemble E8) on the E1M EVK with RPi-style CSI-2 camera modules (OV9281 and OV5647 raw capture, IMX296 and IMX335 raw capture, and the ISP-Pico path with OV5647), and on E1M-V2M103 with an IMX296LQ (RAW10 through the Linux V4L2 backend). Other sensors and other SoMs are unverified. The `ai-camera-viewer` reference example itself is still build-verified only (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. Budget bench time for first light on your own sensor.

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	`ai-camera-viewer` builds and flashes against the `<alp/camera.h>` 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	`examples/camera-vision/ai-camera-viewer/` in alp-sdk.

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. The SDK drives the ISP-Pico on the Alif Ensemble E8 only; the Renesas ISP is not reachable from the SDK. 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:

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

3 ISP Status

ISP availability differs by SoM. 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: only the Alif Ensemble E8 (E1M-AEN801 / E1M-AEN803) has a driven ISP today.

Copyright © 2026 Alp Lab AB

Backend	Status
Zephyr video (portable, any board with a `video_*` sensor driver)	Sensor pipeline (open / start / stop / capture / release / close) is real code. `alp_camera_configure_isp()` returns `ALP_ERR_NOSUPPORT`: the portable video class has no in-line ISP surface.
Alif ISP-Pico (E8 only; not E4 or E6)	Opt-in (`CONFIG_ALP_SDK_CAMERA_ALIF_ISP`, default off; needs the ISP devicetree node). Capture through the ISP, with auto-exposure and auto-white-balance running, is bench-proven on E1M-AEN803 with an OV5647 (`examples/aen/aen-isp-capture`, `examples/aen/aen-isp-ov5647-viewfinder`). Its `alp_camera_configure_isp()` still latches the requested configuration and returns `ALP_OK`, so treat `ALP_OK` as "accepted", not as proof the ISP was reprogrammed. The Alif vendor-extension surface (3A windows, per-channel gain tables, lens-shading LUTs) returns `ALP_ERR_NOSUPPORT` from every entry point. Widening it to E4 / E6 needs explicit backend registrations plus board validation. IMX296 colour is uncalibrated (no AWB / colour-correction table).
Renesas V2N N44 ISP (zephyr)	`alp_camera_configure_isp()` and every Renesas vendor-extension ISP call return `ALP_ERR_NOSUPPORT`. The Cortex-A55 owns the CRU, CSI-2 and ISP blocks; the Cortex-M33 has no module for them.
Linux V4L2 (`yocto_drv`, any Linux target)	V4L2 plus media-controller backend. `camera_id` N maps to the devicetree alias `alp-camera<N>`; any sensor with a mainline V4L2 subdev driver works with no per-sensor code. Supports `ALP_PIXFMT_GREY8`, `RAW8` and `RAW10` only (colour formats return `ALP_ERR_NOSUPPORT`). RAW10 is bench-verified on E1M-V2M103 with an IMX296LQ; RAW8 is not.

Table 3 Camera backends and ISP status (SDK v0.17.0)

> **Warning:** `ALP_OK` from `alp_camera_configure_isp()` never means a Renesas ISP was programmed: on V2N it returns `ALP_ERR_NOSUPPORT`. Check the return code, and do not gate image quality on it on any SoM.

New: `alp_camera_get_fps()` returns the frame rate the backend actually settled on at open, in thousandths of a frame per second (29.97 fps reads as 29970). `alp_camera_config_t::fps` is only a request; read the settled value instead of assuming it.

Plan on your sensor delivering the pixel format you need directly, and on doing any 3A or tuning work in your application, except where the ISP-Pico path above applies.

4 Software Walkthrough

```
tan build --project examples/camera-vision/ai-camera-viewer
tan flash examples/camera-vision/ai-camera-viewer
```

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.

Copyright © 2026 Alp Lab AB

5 Troubleshooting

- **LDO voltage off**: divider current too low; reduce R1 + R2 until divider current = 100 × worst-case feedback-pin current (see **DS-AEN-001 §8.7** formula).
- **No CSI link**: differential lanes swapped (P/N reversed) or trace impedance not 100 Ω differential.
- **E1M-AEN hw_rev R2: camera never produces frames on the E1M EVK**: on R2 modules the P and N pads of each CSI-2 lane are swapped (see **E1M-AEN Errata E-01**). On the E1M EVK's 15-pin camera connector J3 an R2 module needs a P/N-crossing adapter: swap pins 13 ↔ 14 (D0), 10 ↔ 11 (D1) and 7 ↔ 8 (CLK) and leave grounds, power and control straight. Without it the camera answers on I^2C but no frame ever arrives, and the capture call times out. R3 modules carry the corrected pad polarity; read the module hardware revision from the identity EEPROM before applying either rule.
- **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`. Bench examples: `examples/aen/aen-camera-firstlight/` (raw first light), `examples/aen/aen-isp-capture/` and `examples/aen/aen-isp-ov5647-viewfinder/` (capture through the ISP-Pico). `examples/aen/aen-isp-regcheck/` is a test-pattern frame test of the ISP, not a sensor capture. Linux camera path: `docs/v2n-camera-csi.md` and `docs/camera-shields.md`. (The earlier `aen-camera-regcheck` example no longer exists.)
- **Hardware Design Guides**: **HG-AEN-001** §8.3, **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 `examples/camera-vision/ai-camera-viewer/`, added `include/alp/camera.h` and the `aen-camera-regcheck` / `aen-isp-regcheck` reg-checks ▸ corrected the AEN on-die ISP to the Alif Ensemble ISP-Pico (`vsi,isp-pico`) ▸ aligned the stream config with the example (240×240 RGB565).	June 2026
0.3	• Corrected the ISP story against alp-sdk source: the ISP-Pico backend registers for E8 only (not an E4/E6/E8 support tier), its Kconfig defaults to n, and the devicetree node is disabled — so no ISP offload is reachable today and the vendor-ext camera surface returns `ALP_ERR_NOSUPPORT`. • 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 `configure_isp` returns `ALP_OK` without programming the ISP. • Removed a fabricated expected-output log. • Fixed the Alp SDK repository links (`alpDevs` → `alplabai`).	July 2026
0.4	• Re-verified against Alp SDK™ v0.17.0. • Replaced "no ISP offload on any SoM" with the per-backend status: Alif ISP-Pico is bench-proven on E8 and its `configure_isp` still returns `ALP_OK`; on V2N `configure_isp` and every Renesas ISP vendor-extension call return `ALP_ERR_NOSUPPORT` (the earlier text said `ALP_OK`). • Added the Linux V4L2 backend (`yocto_drv`, `alp-camera<N>` alias, GREY8 / RAW8 / RAW10, RAW10 verified on E1M-V2M103 with an IMX296LQ) and `alp_camera_get_fps()`. • Updated the capture bench status. • References now point to `aen-camera-firstlight` and `aen-isp-capture`; `aen-camera-regcheck` was removed and `aen-isp-regcheck` is a test-pattern frame test.	October 2026

Copyright © 2026 Alp Lab AB

Revision	Changes	Date
	• Added the E1M-AEN hw_rev R2 CSI-2 P/N-swap erratum and the P/N-crossing adapter on the E1M EVK J5.	
0.5	• E1M EVK camera connector corrected for the 2641-R3 board: the 15-pin CSI-2 connector is J3 (not J5). • Lane pins are 13 / 14 (D0), 10 / 11 (D1) and 7 / 8 (CLK); the R2 P/N-crossing adapter pin pairs updated to match.	October 2026
0.6	• Release history: every entry formatted as a bullet list.	October 2026
0.7	• Product name on page 1 links to the product page on alplab.ai.	November 2026

Table 4 Revision History

Copyright © 2026 Alp Lab AB