



I²C Bus Scanning and Sensor Bring-up

AN-001 · E1M and E1M-X SoMs

Document Number: AN-001 **Revision:** 0.3 **Date:** July 2026 **Status:** Preliminary

alplab.ai

© 2026 Alp Lab AB. All rights reserved.

1 Scope

This application note shows how to walk the on-EVK I²C bus, identify which peripherals respond, and read a first byte from a known sensor. It applies to **all** E1M and E1M-X SoMs through the Alp SDK™'s cross-family `<alp/peripheral.h>` I²C API.

Audience	Firmware engineers writing their first peripheral driver against the SDK.
Prerequisites	QS-E1M-EVK-001 or QS-E1M-X-EVK-001 completed (hello-world running).
Outcome	i2c-scanner running on the EVK, printing a list of responding addresses including the on-board sensor lineup.
Time	10 minutes.
Source	<code>examples/peripheral-io/i2c-scanner/</code> and <code>docs/tutorials/02-i2c-scan.md</code> in alp-sdk .

Table 1 Scope summary

2 Hardware Setup

No external wiring is required – the EVKs already populate sensors on the primary sensor bus, exposed portably as `BOARD_I2C_SENSORS` (`E1M_I2C0` on the E1M EVK, `E1M_X_I2C0` on the E1M-X EVK).

EVK + SoM	On-board peripherals on <code>BOARD_I2C_SENSORS</code> (default-build 7-bit addresses)
E1M EVK + AEN801	On-module: OPTIGA Trust M (0x30), TMP112 (0x48), 24C128 EEPROM (0x50), RV-3028-C7 RTC (0x52). On-board (EVK-only): TCAL9538 I/O expander (0x72). Soldered IMUs (BMI323, ICM-42670) and the BMP581 barometer are also fitted; the INA236 current-monitor array sits at 0x40–0x46.
E1M-X EVK + V2N101	On-module (BRD_I2C): OPTIGA Trust M (0x30), TMP112 (0x48), RV-3028-C7 RTC (0x52), 5L35023B clock (0x68), GD32G553 supervisor bridge (0x70), ACT8760 PMIC pages (0x25/0x26), DA9292 PMIC (0x1E), TPS628640 (0x4D, optional). 24C128 EEPROM (0x50) sits on the separate <code>E1M_I2C0</code> bank. On-board (X-EVK): TCAL9538 I/O expanders (0x72), INA236 current-monitor array.

Table 2 Expected I²C bus map per EVK

The full populated-parts list per SoM lives in `metadata/e1m_modules/<SKU>.yaml` in the SDK.

3 Software Walkthrough

Step	Command
1. Build	<code>west build -b <BOARD> examples/peripheral-io/i2c-scanner</code>
2. Flash	<code>west flash</code>
3. Watch console	<code>tio -b 115200 <serial-port></code> (see QS- guide for the per-OS device name)

Table 3 Build / flash / run

Replace `<BOARD>` with the target string from your EVK Getting Started Guide (e.g. `alp_e1m_aen801_m55_hp` for AEN, `alp_e1m_v2n101_m33_sm` for V2N).

The scanner opens `BOARD_I2C_SENSORS` at 100 kHz via `alp_i2c_open()` and walks 7-bit addresses 0x08–0x77. At each address it issues a single 1-byte `alp_i2c_read()` probe – a present chip ACKs its address byte (the data is discarded), an empty address NACKs – and prints a row for every responder. A 1-byte read is used rather than a zero-length write because some controllers (e.g. the DesignWare I²C on the Alif Ensemble) put nothing on the bus for a zero-length transfer, so no device would ever ACK.

4 Expected Output

```
[i2c] open BOARD_I2C_SENSORS @ 100 kHz
[i2c] addr 0x30 acked (OPTIGA Trust M)
[i2c] addr 0x48 acked (TMP112)
[i2c] addr 0x50 acked (24C128 EEPROM)
[i2c] addr 0x52 acked (RV-3028-C7 RTC)
[i2c] addr 0x72 acked (TCAL9538 I/O expander, EVK only)
[i2c] scan complete, 5 responder(s)
[i2c] done
```

The above is the `alp_e1m_aen801_m55_hp` + E1M EVK case (the device names in parentheses are annotations – the scanner reports addresses only, not chip identities). The exact set depends on the EVK + SoM combination. Any device listed in **Table 2** that is missing from the scan is a wiring or power issue.

5 Troubleshooting

Symptom	Likely cause / fix
Empty scan (no ACKs)	Pull-ups missing. The SoM's I2C0_SCL / I2C0_SDA need 4.7 k Ω pull-ups to VIO_OUT; on the EVK these are populated, but on a custom carrier they may be missing.
Only on-module devices ACK, no carrier sensors	VIO_OUT is up but the carrier-side sensor power gate is off. On the E1M-X EVK this is controlled by the TCAL9538 I/O expander at 0x72 (output bit TBD).
Bus locks up mid-scan	Pull-down on SDA or SCL (shorted trace, dead sensor). Probe both lines with a scope; idle state should be 1V8.

Table 4 Common failures

6 References

- **Canonical tutorial:** `docs/tutorials/02-i2c-scan.md` in `alp-sdk`.
- **Example source:** `examples/peripheral-io/i2c-scanner/` in `alp-sdk`.
- **SDK API:** `<alp/peripheral.h>` (`alp_i2c_open` / `alp_i2c_read` / `alp_i2c_write` / `alp_i2c_close`).
- **Per-SoM populated parts:** `metadata/e1m_modules/<SKU>.yaml` in `alp-sdk`.
- **EVK Getting Started:** **QS-E1M-EVK-001** (E1M EVK), **QS-E1M-X-EVK-001** (E1M-X EVK).

7 Revision History

Revision	Changes	Date
0.1	Initial draft. Customer-facing reference card pointing at the SDK tutorial as canonical source.	May 2026
0.2	Re-synced to current <code>alp-sdk</code> : example path <code>examples/peripheral-io/i2c-scanner/</code> ; API header <code><alp/i2c.h></code> → <code><alp/peripheral.h></code> (<code>alp_i2c_open/read/write/close</code>); V2N board target <code>alp_e1m_v2n101_m33_sm</code> ; bus named <code>BOARD_I2C_SENSORS</code> ; corrected I ² C bus maps and expected scanner output against <code>docs/tutorials/02-i2c-scan.md</code> and <code>metadata/(TMP112 0x48 on AEN, DA9292 0x1E on V2N, TCAL9538 0x72 I/O expander)</code> ; scanner described as a 1-byte read probe.	June 2026
0.3	Corrected the E1M-X EVK + V2N101 bus map: TMP112 was listed at 0x40, which the part cannot strap to – its ADD0 strap only permits 0x48–0x4B, and 0x40–0x46 is the INA236 array's range. Corrected to 0x48, matching the SDK's own bus model. Updated the AEN board target to <code>alp_e1m_aen801_m55_hp</code> (no AEN701 board wrapper exists). Fixed the Alp SDK repository links (<code>alpDevs</code> → <code>alplabai</code>).	July 2026

Table 5 Revision History