



CAN-BUS Network Integration

AN-005 · E1M and E1M-X SoMs

Document Number: AN-005 **Revision:** 0.3 **Date:** July 2026 **Status:** Preliminary

alplab.ai

© 2026 Alp Lab AB. All rights reserved.

1 Scope

Bring up the on-module CAN-FD transceiver, terminate the bus correctly, and exchange frames between the SoM and a peer node. The CAN transceivers are **on-module** (TI TCAN1044AVDRBRQ1) – the carrier provides only termination and the connector.

Audience	Firmware engineers building industrial / automotive applications with CAN networks.
Prerequisites	QS- guide completed, second CAN node (e.g. a USB-to-CAN adapter) connected to the EVK's CAN connector with proper termination.
Outcome	Bidirectional CAN-FD frame exchange at 1 Mbps arbitration / 5 Mbps data rate.
Time	15 minutes.
Source	examples/peripheral-io/can-loopback/ in alp-sdk .

Table 1 Scope summary

2 Hardware Setup

SoM family	CAN ports + on-module transceiver
E1M-AEN	1 × CAN-FD (CAN0). On-module TI TCAN1044AVDRBRQ1. Carrier exposes bus-level CAN0H/CAN0L.
E1M-X V2N / V2N-M1	2 × CAN-FD (CAN0, CAN1). Same TI transceiver. Carrier exposes both bus-level pairs.

Table 2 CAN support per family

Termination: 120 Ω at each end of the bus, between CANH and CANL. The EVK has a jumper-selectable 120 Ω terminator on each port. Enable it when the EVK is at the bus end; disable when in the middle of a multi-drop bus.

Warning: Do not add an external CAN PHY between the SoM and the connector. The on-module transceiver is mandatory per **E1M Spec §6.5**. Adding a second PHY breaks conformance and may damage the on-module driver.

3 Software Walkthrough

```
west build -b <BOARD> examples/peripheral-io/can-loopback
west flash
```

The example opens CAN0 (via the portable BOARD_CAN0 alias from `<alp/board.h>`) in **loopback** mode at 500 kbps classic CAN, installs a permissive RX filter with a receive callback, sends one 8-byte frame (ID 0x123), and confirms the controller routes it straight back to its own RX path. Loopback is the canonical bring-up test: it validates clocking, the TX/RX encode-decode paths, and filter dispatch without the bus being wired. Set `loopback = false` (and switch to ALP_CAN_MODE_FD with a non-zero `bitrate_data_hz` for CAN-FD) to drive the real differential pair against a peer node.

4 Expected Output

```
[can] open BOARD_CAN0 @ 500 kbps loopback
[can] add_filter -> status=0 fid=0
[can] start -> 0
[can] send id=0x123 -> 0
[can] rx id=0x123 dlc=8 data[0..3]=01 02 03 04
[can] rx_count=1
[can] done
```

5 Troubleshooting

- **No bus activity:** termination missing or wrong (must be 120 Ω at each end; not in the middle).
- **Bus error frames:** bitrate mismatch with peer, or CAN-FD vs classic-CAN mismatch.
- **TX-only, no RX:** peer node not in the same FD configuration, or BRS mismatch.
- **ESD-induced lockup:** add TVS clamp diodes between CANH/CANL and ground on the carrier connector (not on-module).

6 References

- **Example source:** `examples/peripheral-io/can-loopback/` in `alp-sdk`; portable API in `include/alp/can.h`.
- **Silicon bring-up check:** `examples/aen/aen-can-regcheck/` validates the `alp_can0` devicetree binding on E1M-AEN.
- **Hardware Design Guides:** **HG-AEN-001** §5.7, **HG-V2N-001** §6.6 (CAN-bus routing and termination).
- **Datasheet CAN pin tables:** §2.3 in each datasheet.

7 Revision History

Revision	Changes	Date
0.1	Initial draft.	May 2026
0.2	Updated example paths to the reorganised <code>examples/peripheral-io/can-loopback/</code> topic folder; added <code>examples/aen/aen-can-regcheck/</code> and <code>include/alp/can.h</code> as canonical sources; rewrote the software walkthrough and expected-output block to match the actual loopback example (500 kbps classic CAN, <code>BOARD_CAN0</code> alias, single-frame self-test).	June 2026
0.3	Corrected the Alp SDK repository links: the <code>alpDevs</code> GitHub organisation does not exist, so every link 404'd; the canonical organisation is <code>alplabai</code> .	July 2026

Table 3 Revision History