NEXILICA

# CANnect

## CAN to CAN interface adapter

## Revision B

## 1. Features

- **Two independent CAN / CAN-FD interfaces, up to 5 Mbps**
- **DBC-driven signal mapping and ID translation**
- **Real-time bridge engine, low-millisecond latency**
- **Up to 64 user-configurable forwarding rules**
- **Runtime configuration over CAN, atomic two-bank persistence**
- **In-app firmware update over CAN**
- **5 V auxiliary analog input**
- **9 - 28 V wide operating supply voltage**
- **Compact size: 35 mm x 12 mm**
- **AEC-Q100 / Q1 components**

## 2. Applications

- Automotive prototyping and testing
- Legacy system integration
- CAN network simulation and replay
- Sensor data conditioning
- Protocol translation between ECUs
- Bus health monitoring and watchdog supervision

## 3. Description

The CANnect board is a flexible, real-time CAN-to-CAN bridge for automotive, industrial and embedded applications. It connects two independent FDCAN buses and forwards messages between them with optional ID translation, signal-level transformations, baud-rate conversion and DBC-driven mapping.

Built around a 32-bit Arm Cortex-M4F MCU with two independent FDCAN cores, CANnect handles classic CAN and CAN-FD frames at up to 5 Mbps. Bridging rules and signal descriptors are pushed at runtime over CAN and committed atomically to two-bank EEPROM, with no service interruption.

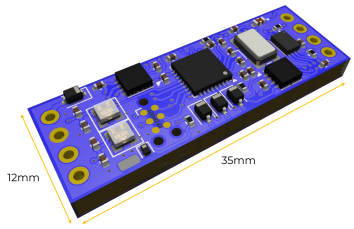


**Figure 1:** CANnect

# 4. Technical Characteristics

## Absolute maximum ratings

Stresses beyond those listed in Table 1 may cause permanent damage to the device. These are stress ratings only; functional operation at or beyond the values listed under *Electrical characteristics* is not implied.

| PARAMETER | MIN | MAX | UNIT |
|---|---|---|---|
| Supply voltage | 7 | 33 | V |
| Storage temperature | -40 | 125 | °C |

**Table 1:** Absolute maximum ratings

## Electrical characteristics

| PARAMETER | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|
| Supply voltage | 9 | 12 | 28 | V |
| Supply current (typical) | 14 | 21 | 32 | mA |
| Operating temperature | -20 | — | 105 | °C |

**Table 2:** Electrical and environmental characteristics

| CAN BAUD RATE | MAX BUS LENGTH | NOTE |
|---|---|---|
| 1 Mbps | 40 m | ISO 11898-2 |
| 500 kbps | 100 m | ISO 11898-2 |
| 250 kbps | 250 m | ISO 11898-2 |
| 125 kbps | 500 m | ISO 11898-2 |

**Table 3:** CAN bus length vs. baud rate (Classic CAN, twisted pair, terminated 120 Ω)

## 5. Board pin-out

The table below shows the corresponding pin-out based on the cable colour. VCC and GND are pass-through between the two pad strips, so the device can be installed in-line on the harness power line.

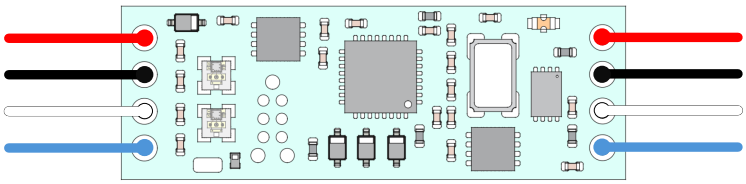

**Figure 2:** Pin-out and wire colour assignment

| PIN | SIGNAL | WIRE | RANGE | DESCRIPTION |
|---|---|---|---|---|
| — | VCC | Red | 9 – 28 V | Board supply |
| — | GND | Black | 0 V (ref.) | Board return |
| — | CAN-H | White | 0 – 3.3 V | CAN bus high (ISO 11898-2) |
| — | CAN-L | Blue | 0 – 3.3 V | CAN bus low (ISO 11898-2) |

**Table 4:** Cable colour to signal mapping

A separate analog pad on the board exposes a 0 - 5 V general-purpose input to the MCU's 12-bit ADC, scaled by an internal 1:2 divider with RC anti-aliasing and ESD protection. The input voltage range can be extended on request by changing the divider ratio and the resistor values of the front-end network.

## 6. System architecture

CANnect is organised around a single 32-bit Arm Cortex-M4F MCU that mediates all data paths. Two FDCAN cores drive independent CAN front-ends; an SPI EEPROM holds the persistent configuration; two addressable RGB LEDs report operational status. The power chain produces a 5 V rail for the transceivers and a 3.3 V rail for the MCU and digital logic from a wide 9 - 28 V input, with reverse-polarity protection and a transient clamp on the input.

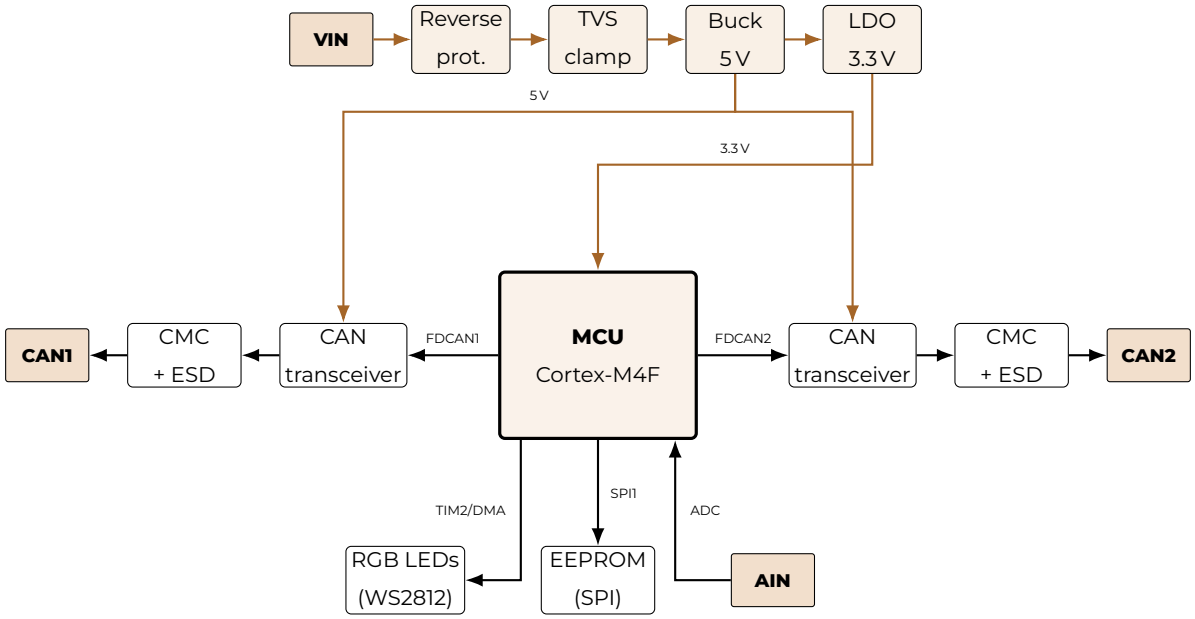


**Figure 3:** System block diagram

The auxiliary 0 - 5 V analog input passes through a small front-end before reaching the MCU's 12-bit ADC. A 5 V standoff TVS clamps over-voltage at the pad; a 10 kΩ series resistor and a 10 kΩ resistor to ground form a 1:2 voltage divider that brings the 0 - 5 V range into the 0 - 2.5 V window expected by the ADC; a 100 nF capacitor closes the RC anti-aliasing low-pass with a $-3$ dB corner around 3 kHz.

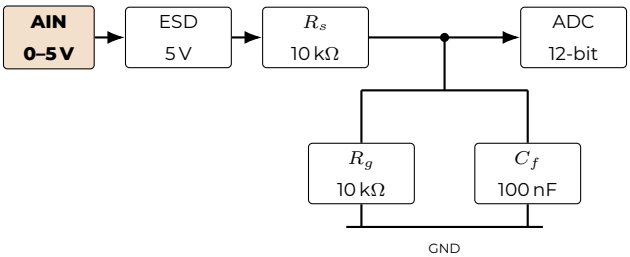


**Figure 4:** Analog input front-end

# 7. Hardware

## 7.1 Power supply

The input stage accepts a wide 9 - 28 V while VCC and GND are passed through between the two pad strips, so the device inserts in-line on the harness power line and forwards full input current to the downstream segment minus its own consumption.

## 7.2 MCU subsystem

The 32-bit Arm Cortex-M4F core has 512 KB of flash and 112 KB of SRAM. The MCU integrates two independent FDCAN cores, an SPI controller dedicated to the EEPROM, a 12-bit ADC for the analog input, and a hardware timer with DMA used to drive the LED chain without CPU bit-banging. An external 8 MHz crystal feeds the PLL and the FDCAN kernel clock.

## 7.3 CAN front-end

Each CAN channel uses an automotive-grade FDCAN transceiver paired with a common-mode choke and bus-side ESD protection. Both transceivers are permanently in normal mode (no standby), VIO matched to the MCU 3.3 V logic and VCC supplied from the 5 V rail.

The front-end supports Classic CAN at 125 kbps, 250 kbps, 500 kbps and 1 Mbps, and CAN-FD up to 5 Mbps in the data phase. The two channels are fully independent and can run different baud rates simultaneously.

The board does not populate termination resistors; the integrator must provide 120 Ω termination at both ends of each bus segment. The two CAN sides share GND and the input rail, so CANnect is not galvanically isolated and cannot be used to bridge two electrically isolated power domains.

## 7.4 EEPROM

A 16 KB automotive-grade SPI EEPROM stores the persistent configuration (rule table, signal descriptors, operation parameters and optional calibration data). The device supports more than 4 million erase/write cycles per cell and provides 200 years of typical data retention at 25 °C.

The EEPROM is accessed via SPI1 with DMA, addressed as a single chip on the bus. A two-bank layout with header-driven active-bank pointer supports atomic commit and rollback to the previous known-good configuration on commit failure or power loss.

## 7.5 Indicators

Two addressable RGB LEDs emit through cuts in the PCB and are daisy-chained on a single MCU pin driven by hardware PWM and DMA. Each LED can render arbitrary 24-bit colours and the firmware uses them as channel-specific status indicators.

| COLOUR | MEANING |
|---|---|
| Both white | Boot in progress |
| Both green steady | Boot OK, idle |
| Green flicker | TX activity (per channel) |
| Blue flicker | RX activity (per channel) |
| Yellow flicker | TX + RX activity |
| Red | Bus error, recoverable |
| Both red, blinking | Faulted, one or more rules disabled |
| Both red, steady | Fatal, safe state |
| Magenta pulse | Configuration commit in progress |
| Cyan pulse | Firmware update in progress |

**Table 5:** Status LED colour code

## 7.6 Analog input

A general-purpose 0 - 5 V analog input is exposed on a dedicated pad and routed to the MCU's 12-bit ADC through a 1:2 resistive divider with RC anti-aliasing and 5 V standoff ESD protection (see Figure 4 for the front-end schematic). The input impedance to ground is approximately 10 kΩ and the analog bandwidth is about 3 kHz.

The input voltage range can be extended on request by changing the divider ratio and the resistor values of the front-end network, to accommodate signals beyond 5 V or to trade off input impedance against bandwidth.

The input is intended for auxiliary sensor publishing on CAN (temperature probes, voltage dividers, current shunts) or for selecting operational profiles by external voltage.

# 8. Firmware

## 8.1 Boot and initialization

On reset, the firmware completes the following sequence within a target budget of 100 ms before the first CAN frame can be transmitted:

- Startup code, HAL init, NVIC and SysTick configuration
- Clock: external 8 MHz crystal → PLL → SYSCLK and FDCAN kernel clock
- Peripheral init in idle mode (FDCAN cores not started)
- BSP init: LED driver, SPI EEPROM presence and CRC check
- Configuration load from EEPROM (active bank), with automatic rollback to the alternate bank on CRC failure
- Bridge engine initialization, FDCAN filter installation

If both EEPROM banks are corrupt, the device boots silent (empty rule set) and waits for a configuration push. On unrecoverable failure the LEDs go solid red, the FDCAN cores are stopped and the watchdog resets the device.

## 8.2 FDCAN service

The FDCAN service offers a uniform API across the two channels and across classic CAN and CAN-FD frames. Each channel has independent nominal and data-phase bit-timing, configured at boot from the persisted configuration and reconfigurable at runtime.

Filters are installed per RX rule using standard or extended IDs, with exact-match acceptance. The RX path is interrupt-driven and posts to a per-channel queue with capture timestamp; the TX path uses the FDCAN TX FIFO with auto-retransmission enabled. Bus state is monitored continuously: TEC/REC counters, error-passive and bus-off flags are exposed to the diagnostic layer, and bus-off recovery is handled automatically by the service.

## 8.3 Bridge engine

The bridge engine evaluates up to 64 rules on a periodic tick. Each rule maps an RX descriptor (channel, ID, ID type, optional DLC filter) to a TX descriptor (channel, ID, ID type, frame format) and carries a list of operations from a built-in catalogue.

Trigger modes include *ONRX* (fires on a new matching RX), *CYCLIC* (fires on its own period), *ONCHANGE* and *ONTHRESHOLD* (fire on decoded-signal events) and *ASYNC* (rate-unbound ONRX). Triggers can be combined.

Rules are walked in priority order computed at commit time, so jitter on fast cyclic rules is not affected by the number of slow rules. Steady-state forwarding latency is below 2 ms on classic CAN, dominated by the bridge tick and transceiver propagation.

## 8.4 Signal pipeline

A pre-compiled binary representation of DBC signal descriptors lets the firmware decode and encode CAN frames at signal level without parsing `.dbc` files on the device. Each descriptor specifies frame ID, start bit, bit length, byte order, value type, scale, offset and saturation limits.

The operation catalogue covers ID remap and pass-through (Bridge family); DBC decode/encode, scale/offset and unit conversion (Transform); formula, filter, decimate and aggregate (Compute); threshold and timeout (Supervise); replay, pattern and responder (Simulate); and ADC read (Sense). All operations run within the per-tick time budget; rules exceeding the budget are deferred to the next tick and reported in diagnostics.

## 8.5 Configuration and storage

The runtime configuration consists of a signal descriptor table, a rule table, an operation parameter blob and the active filter set. These are held in RAM for hot-path access and persisted in EEPROM through a two-bank scheme: a small header at the top of the device addresses the currently active bank; each bank stores a manifest with payload size, CRC-32 and monotonic commit sequence number, followed by the configuration payload.

The configuration protocol is multi-frame and runs over the same CAN buses used for the data path. A host PC tool parses the user's DBC files and produces the compiled descriptor blob, which is then pushed to the device through a dedicated set of configuration IDs (filtered out of the bridge engine and never forwarded). Commit is atomic: the inactive bank is fully written and verified, then the header is flipped; an interrupted commit always leaves the previous configuration valid.

## 8.6 Diagnostics and recovery

Errors are classified in four levels: *transient* (counted, no functional impact), *recoverable* (service-level recovery + telemetry, data path continues), *faulted* (offending rule disabled, rest of system unaffected) and *fatal* (safe state and watchdog reset).

When enabled, the device emits diagnostic frames over a configurable CAN ID range: a heartbeat with uptime and active commit sequence number, periodic bus-health frames with per-channel error counters, on-event reports and a one-shot boot report describing the previous reset cause.

An independent watchdog (IWDG) supervises the firmware and resets the device if any subsystem stops servicing its liveness check.

## 8.7 Field firmware update

Firmware updates are performed over CAN through the same configuration protocol: the new image is staged, CRC-verified, then committed by a small bootloader that falls back to the previous image if the staged image fails verification.

## 9. Use cases

The bridge engine and signal pipeline support six application families that compose freely in a single rule set.

### 9.1 Bridge

Pure forwarding between the two buses, with optional ID translation and per-channel baud-rate conversion.

### 9.2 Transform

Signal-level modification of frame payloads: scale and offset, unit conversion (°C ↔ °F, kPa ↔ bar, RPM ↔ Hz), endianness conversion and signal remap between IDs.

### 9.3 Compute

Synthesis of new information using the on-board FPU: virtual signals from arithmetic on multiple sources, low-pass / moving-average / median filtering, decimation, aggregation of slow signals into a single frame and the inverse de-aggregation.

### 9.4 Supervise

Conditional forwarding and bus monitoring: threshold-triggered emission, watchdog on expected messages with timeout alarms, periodic bus-health reports.

### 9.5 Simulate

Generation of CAN traffic with little or no external input: trace replay, ECU emulation (OBD-II responder, UDS/KWP2000 server, J1939 node stand-in), pattern generation (counters, ramps, sweeps), fault injection.

### 9.6 Sense

Bringing the auxiliary 0 - 5 V analog input onto the network: publishing of NTC, voltage-divider or current-shunt readings as CAN signals, or operational profile selection by an external voltage.

## 10. Mechanical dimensions

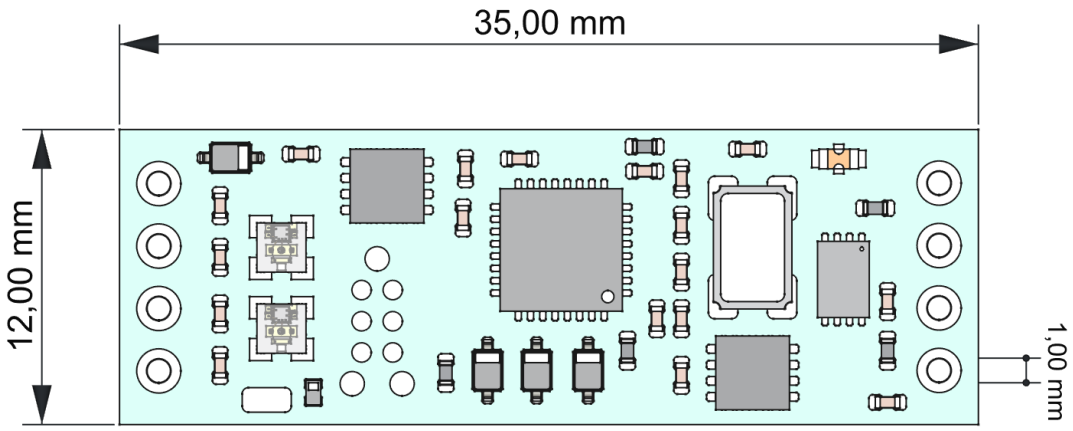


**Figure 5:** CANnect mechanical drawing