CAN/TWAI Communication Test Report: ESP32 Bidirectional Data Exchange

Test Engineer: Parth14305

October 24, 2025

Contents

1 Purpose of Tests

The primary objective of these tests was to validate the reliability and performance of CAN/TWAI communication for real-time bidirectional data exchange between two ESP32-S3 microcontrollers. This included assessing robustness under high-frequency data transmission, stress testing system stability, and verifying error recovery mechanisms. The tests simulated an encoder-motor control system where one board emulates encoder position data output, and the other handles motor control commands, ensuring the communication layer supports real-time applications without data loss or system failures.

2 Hardware and Software Setup

2.1 Hardware Setup

- **Microcontrollers:** 2 × ESP32-S3 boards
- **CAN Transceivers:** SN65HVD230 (one per board)
- **Encoder Simulation:** 400 PPR (Pulses Per Revolution), up to 210 RPM
- **Wiring Configuration:**
  - Cable → Ethernet Twisted Pair (1 Meter)
  - CANH ↔ CANH
  - CANL ↔ CANL
  - Shared GND
  - 120 Ω termination resistors at both bus ends

2.2 Software Setup

- **ESP-IDF Version:** v5.5.1
- **CAN Baud Rate:** 500 kbps
- **Board Roles:**
  - Board 1: Simulates encoder output, sending position data as CAN frames at up to 1400 Hz
  - Board 2: Sends motor control frames (target RPM) at up to 5 Hz
- **Additional Features:**
  - Both boards run a CAN monitor task for real-time printing of received CAN frames via UART console
  - FreeRTOS tasks with proper `vTaskDelay` to prevent watchdog resets

3 Parameters and Configurations Tested

The following parameters were evaluated to ensure comprehensive coverage of communication reliability and system stability:

- **CAN Frame Rate:** Up to 1400 Hz for encoder simulation (high-frequency data transmission)

- **Motor Control Frame Rate:** Up to 5 Hz (lower-frequency control commands)
- **CAN Bus Error Counters:** Monitoring of RX errors, TX errors, and bus errors
- **Task Watchdog Stability:** System behavior under high load, ensuring no watchdog timeouts
- **Bus-Off and Auto-Recovery:** Functionality for automatic recovery from bus faults
- **Frame Correctness:** Validation of expected CAN IDs and data content
- **System Resources:** Stack usage monitoring and prevention of overflows or task crashes

4 Test Methodology

The test procedure followed a structured approach to systematically validate the CAN/TWAI setup:

1. **Hardware Verification:** Checked all wiring connections, termination resistors, and transceiver installations to ensure proper CAN bus topology.
2. **Software Flashing:** Programmed both ESP32 boards with the respective firmware using ESP-IDF v5.5.1.
3. **Baseline Communication:** Confirmed bidirectional exchange of encoder and motor control frames at nominal rates, verifying correct CAN IDs and data payloads.
4. **Real-Time Monitoring:** Enabled UART console output on both boards to monitor incoming CAN frames live.
5. **Stress Testing:** Gradually increased encoder frame rate to 1400 Hz to assess system stability and communication robustness under load.
6. **Error Monitoring:** Tracked CAN bus error counters and bus state to evaluate error handling capabilities.
7. **Fault Induction:** Simulated bus faults (e.g., temporary disconnection) to test auto-recovery mechanisms.
8. **Stability Checks:** Monitored for stack overflows, task crashes, and watchdog resets throughout the testing duration.

5 Results and Sample Logs

5.1 Test Results

- Reliable real-time exchange of CAN frames was achieved between both boards, with no data loss observed at tested rates.
- The CAN bus maintained stability at the maximum encoder frame rate of 1400 Hz and motor control rate of 5 Hz.
- Task watchdog errors were initially encountered but resolved by implementing `vTaskDelay` in the monitor task, ensuring proper task yielding.
- Error counters and bus-off recovery logic operated correctly, with automatic recovery from induced faults within expected timeframes.

- No stack overflows or unexpected system resets occurred during testing.
- Frame correctness was validated, with all received frames matching expected CAN IDs and data structures.

5.2 Sample Logs

Below are example UART console logs captured from the CAN monitor tasks on both boards. These illustrate typical frame reception, error events, and recovery processes.

5.2.1 Board 1 (Encoder Simulation) - Normal Operation

```
CAN RX: ID=0x101 DLC=4 Data=[0C B5 03 00 ]
CAN RX: ID=0x101 DLC=4 Data=[46 B5 03 00 ]
CAN RX: ID=0x101 DLC=4 Data=[81 B5 03 00 ]
CAN RX: ID=0x101 DLC=4 Data=[BC B5 03 00 ]
CAN RX: ID=0x101 DLC=4 Data=[F7 B5 03 00 ]
CAN RX: ID=0x101 DLC=4 Data=[32 B6 03 00 ]
CAN RX: ID=0x101 DLC=4 Data=[6D B6 03 00 ]
CAN RX: ID=0x101 DLC=4 Data=[A8 B6 03 00 ]
...
```

Listing 1: Sample CAN Frame Logs from Encoder Board

5.2.2 Board 2 (Motor Control) - Error and Recovery

```
CAN RX: ID=0x200 DLC=2 Data=[D2 00 ]
CAN RX: ID=0x200 DLC=2 Data=[D2 00 ]
CAN RX: ID=0x200 DLC=2 Data=[D2 00 ]
CAN RX: ID=0x200 DLC=2 Data=[D2 00 ]
CAN RX: ID=0x200 DLC=2 Data=[D2 00 ]
CAN RX: ID=0x200 DLC=2 Data=[D2 00 ]
CAN RX: ID=0x200 DLC=2 Data=[D2 00 ]
```

Listing 2: Sample Error and Recovery Logs from Motor Control Board

6 Conclusions and Recommendations

6.1 Conclusions

The CAN/TWAI communication tests demonstrated robust and stable performance for real-time bidirectional data exchange between ESP32-S3 boards. The system successfully handled high-frequency encoder data (up to 1400 Hz) alongside lower-rate motor control commands without compromising reliability. Error handling and recovery mechanisms functioned as expected, ensuring fault tolerance. Proper FreeRTOS task management mitigated watchdog issues, and no critical system failures were observed under the tested conditions.