CAN Physical Layer Validation Procedure

A Step-by-step Procedure for CAN Testing, Validation & Debugging

Introduction

The University of Washington EcoCAR team currently lacks the knowledge and training resources to physically test and validate both the team-added and team-modified CAN networks located inside the Chevrolet Blazer. Some of these CAN networks that the Low Voltage Team has wired were made in haste and tend to have longer stub lengths than recommended for a 500-kBps network. Embarking on my research, I began to gain an understanding of CAN Physical Layer validation, which details the process of probing CAN wires with oscilloscopes and interface tools to achieve desired baud rates and prevent aliasing.

After testing the methods that I researched on a test harness of my creation as well as the CAN Buses within the vehicle, I determined the best techniques and applications to doing so. This document details the proposed step-by-step procedure that describes the summation of my research as well as my in-lab testing. Specifically, this will allow members of UW EcoCAR to test, validate, and debug each of the Blazer's CAN networks as we enter the main testing phase of the competition.

Background

Before attempting this proposed procedure, it is recommended that you understand the basics of CAN.

The Controller Area Network Bus, or CAN Bus, is the internal communications network used to allow microcontrollers and other devices to send information to one another. A general CAN Bus Training document is given at the following link: https://docs.google.com/document/d/1mRqhCepzE6NtlrLOLaOdFv-mbZxQBdMv06M1rLaoSh0/edit?usp=sharing

To reiterate some important points from this training document:

- CAN uses two wires to send and receive messages: **CAN High** (CAN H) and **CAN Low** (CAN L).
- **120 Ω resistors** are needed to terminate CAN connections at either end of each network.
- The CAN Protocol sends messages **in binary**, with 0 as a dominant bit and 1 as recessive.
- CAN messages are sent sequentially, but **some messages have higher priority than others**.
- If more than one device transmits at the same time, **the highest priority message is sent first**.

Requirements

This section gives a list of required items, cautions, and warnings you should be aware of before starting the procedure.

Cautions & Warnings

It is important to understand that this procedure includes hazards such as electrostatic discharge (ESD). Static charge can build up in your body over time due to contact with clothing, furniture, and flooring. If enough charge builds up when working with Integrated Circuits, it can jump from you to the electronics, possibly causing permanent damage. Therefore, you should always wear a grounded ESD-protected wristband or use some other method of grounding yourself to prevent ESD from destroying your electronic equipment.

Required Items

The following are the items necessary to test and validate a CAN Bus.

- CAN Controller, MCU, DSP, or Analog Discovery 2 Module
    - Function Generator can weakly simulate CAN Controller
- CAN Transceiver
- CAN Logger
- Quad-channel Oscilloscope & Probes
- Digital Multimeter & Probes
- Power Supply

Procedure

This section gives step-by-step instructions on how to construct a CAN Tester and use it to test and validate CAN signals.

Unpowered System Checks

The first step in debugging a CAN network is to check all connections for the proper continuity, resistance values, and voltage values.

1. While everything is unpowered, make sure all circuit board traces or breadboard connections are properly connected by using the resistance or continuity setting on your Digital Multimeter.
    a. Ensure the controller and transceiver grounds are properly connected.
    b. Ensure the R_S pin of the transceiver is connected to ground to select High-speed CAN mode.
    c. Ensure the transceiver's CAN High and Low pins are connected to the CAN High and Low wires of the CAN bus.
2. Ensure the CAN bus is properly terminated by a 120 Ω resistor by measuring the resistance between CAN High and Low and reading a value between 45 and 65 Ω via the multimeter. Make sure that the location you are measuring is not the OBDII port as that is connected to the serial gateway module and will not function until the car is turned on.
3. Depending on the transceiver, measuring V_{CC} to GND with a multimeter should read 3.3 or 5 V.
4. Add a local bypass capacitor from the transceiver's V_{CC} to GND pins of at least 4.7 μF to eliminate ripple voltage.

Sending & Receiving CAN Signals

Once all connections are properly made and tested, the second step is to attempt to send and receive signals using the tools at hand.

1. Connect each channel of a quad-channel oscilloscope to the CAN High Bus wire, CAN Low Bus wire, TX transceiver pin, and RX transceiver pin.
2. While the car is off, turn on the CAN Controller or function generator (if you do not have access to a CAN Controller) and set it to send a single CAN Data Frame or simple square wave, respectively.
3. Observe the output on the oscilloscope. You should see that the TX and RX signals both mostly match the CAN Low Signal, while the CAN High signal should be an inversion of the CAN Low signal. Additionally, depending on the accuracy of the oscilloscope, you may be able to see a small time delay as well.

4. If your scope has a measurement function, use it to measure the difference in voltage between CAN High and CAN Low when the signal is recessive (1) and while it is dominant (0). You should read a Voltage difference of greater than 1.2 V while the signal is in a recessive state, and between -120 mV and 12 mV while the signal is in a dominant state, as defined by CAN physical layer standards.

Receiving an Acknowledge Bit

After successfully sending and receiving CAN signals, the third step is to introduce a second controller and receive an acknowledgment that the signal was received by it.

1. Plug in the Vector CAN Logger to a location on the CAN network that is not the OBDII port.
2. Plug the USB connector of the Vector CAN Logger into a PC, open CANalyzer, and set it up such that it can log all signals sent while the car is on.
3. Ensure the car is off and the logger is active. Send a single CAN frame via a CAN Controller (a function generator cannot be used) and observe the signal on the Oscilloscope.
4. Since the logger acts as a controller on the network, it will receive the signal and add an Acknowledge or ACK bit to the end of the CAN frame. By comparing the TX and RX signals, you should observe this extra ACK bit on RX but not on TX. You should also observe that CAN High and Low should also have this ACK bit. This ACK bit should be the same size as the bits before it, else this may cause a CAN error frame.

Conclusions

This simple test procedure should confirm or deny the validity of any CAN network it is tested with. In terms of which CAN Controller to use, I would recommend the Analog Discovery 2 Module as it has a CAN Controller, Oscilloscope, Function Generator, and 5 V variable power supply all in one. However, any CAN Controller will do. In terms of a CAN Transceiver, I would recommend Microchip’s MCP2551 High-Speed CAN Transceiver IC or the TI CAN EVM module if you would like a permanent PCB. Both the MCP2551 and Analog Discovery 2 were proven to work very well in my own tests.

Overall, this document should serve as a guide for EcoCAR members to perform tests on all CAN networks in the car during the final year and a half of the competition.

Works Cited

“Analog Discovery 2 QuickStart: Video 14 - The Protocol Analyzer.” Performance by Franz Kaitlyn, *YouTube*, Digilent, Inc., 3 May 2018, youtu.be/_5Iznbeb4q4.

“CAN Bus.” Wikipedia, Wikimedia Foundation, 18 Feb. 2021, en.wikipedia.org/wiki/CAN_bus.

Haase, Jörn, and Gavin C Rogers. “Testing of CAN Signals.” *Aerospace Testing International*, 2015, pp. 40–44.

Hareendran, T K. “DIY CAN Transceiver.” *ElectroSchematics.com*, AspenCore Inc., 2 July 2015, www.electroschematics.com/diy-can-transceiver/.

“How to Connect the Oscilloscope When Dual-Trace Testing.” *Interworld Electronics*, www.interworldna.com/pico/automotive/canbus_scope.php.

McBride, Cooper. “CAN Bus Overview.” UW EcoCAR, 2018.

“MCP2551 High-Speed CAN Transceiver.” Microchip Technology Inc., 2003.

Monroe, Scott. “Basics of Debugging the Controller Area Network (CAN) Physical Layer.” *Analog Applications Journal*, 2013, pp. 18–24.