CAN-Bus Design Notes

Anthony Kramer

**The Controller Area Network Bus,** or CAN-Bus, is a data protocol developed for fast fault-tolerant transmission of data between a network of Microcontroller Units (MCUs). The CAN-Bus is used in almost all automotive systems due to it being both robust and generative.

The CAN-Bus (kind of obviously) uses a bus network topology. This is great because MCU nodes can be added or removed from the network without any changes in code. Also along with the bus topology, there is no defined master or slave nodes, so as long as the address of the MCU is known, data can be sent from and to it without having to go through a switching node. Notice that in the diagram shown below that the ends of the bus are terminated, as this will be important later.

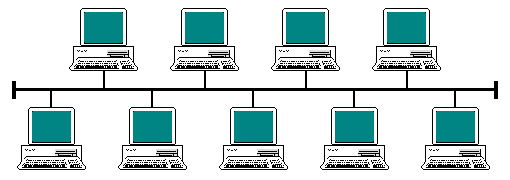


Figure 1: A diagram of the Bus Network Topology.

There are multiple specifications of CAN-Bus, but they normally boil down to two different parameters. The speed and data frame supported. CAN-Bus speed is auto negotiated between the nodes to lowest speed supported by all nodes on the bus. For example, if 9 nodes support 1Mbps and 1 node supports 112Kbps, then the entire bus runs at a max speed of 112Kbps. Data Frame is a little bit more complicated. I know of two normal types of data frames, the Standard 11-Bit Identifier data frame and the Extended 29-Bit Identifier data frame. The difference between these two data frames is that one supports more data per frame, but at a cost of nodal communication during that time the extended frame is being sent. CAN 2.0A only supports the Standard Frame. CAN 2.0B Passive can handle both the Standard and Extended data frames on the bus but ignores anything sent with an Extended frame. CAN 2.0B Active supports and read/write both Standard and Extended frames.

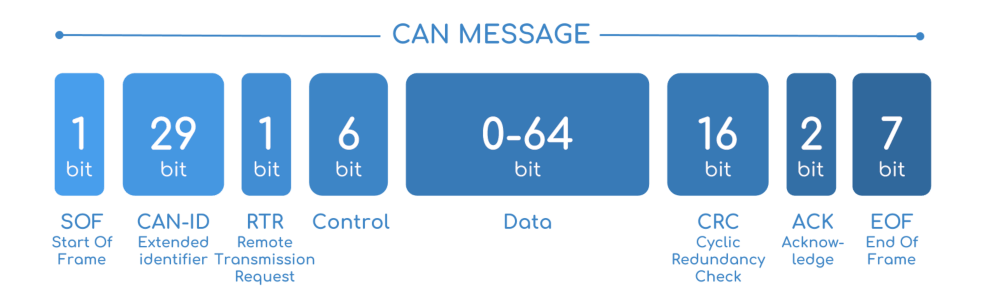


Figure 2: A simple diagram of the Extended CAN Data Frame

Electrically, the CAN-Bus is kind of pecky in terms of wiring. It is composed of two bus wires, an active-high and an active-low wire. The two wires must be in a twisted pair as they are meant to act as together as a robust signal. The wires must be terminated together at both ends of the bus with (normally) a 120Ohm resistor. The bus must also have a very low latent capacitance, as any capacitance will disrupt the signaling of the bus. The wiring is very important as it also can limit the bus speed if done improperly or if the bus is very long.

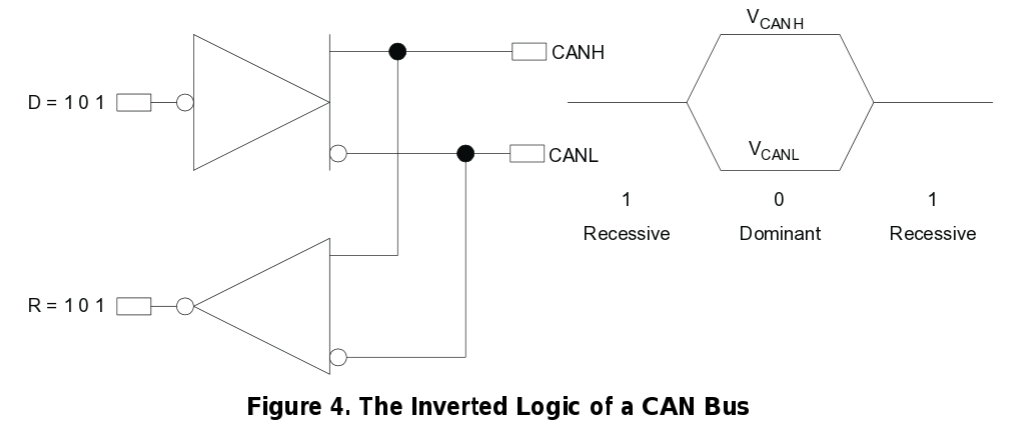


Figure 3: A simple explanation of how the High and Low bus work together for better signal to noise ratios.

In most applications, raw data is handed from the MCU to a CAN-Bus interface, with then controls a CAN-Bus transceiver. Many CAN-Bus interfaces can handle two transceivers for either a load-balancing signal approach or for normal-emergency dual operation. Because CAN-Bus is a fault tolerant network, there is built in fault detection (FD). While FD-CAN is extremely helpful because in most cases, data can be assumed to be received, it also means the bus can be overloaded quickly. If two nodes transmit data at the same time, they will automatically decide which node transmits first after a period of no transmission. However, if many devices all try to transmit information at the same time, the period of no transmission may be a relatively long time on the bus. If this occurs often, the bus will be dramatically overloaded. The cost of transmission increases almost exponentially with each node added. This causes a balancing act between complexity/economy and speed.

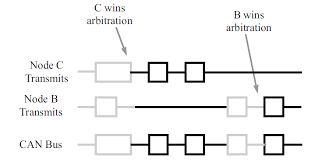


Figure 4: Two nodes “deciding” which node will transmit first.

**This is just a basic explanation of a select aspect of the CAN-Bus, I suggest you try to read the TI article linked below.**

References:

Wikipedia Article on the CAN-Bus

<https://en.wikipedia.org/wiki/CAN_bus>

CSS Electronics Article (Has good explanation of data frame)

<https://www.csselectronics.com/screen/page/simple-intro-to-can-bus/language/en>

Texas Instruments: Introduction to the Controller Area Network (Technical as Frig)

<http://www.ti.com/lit/an/sloa101b/sloa101b.pdf>