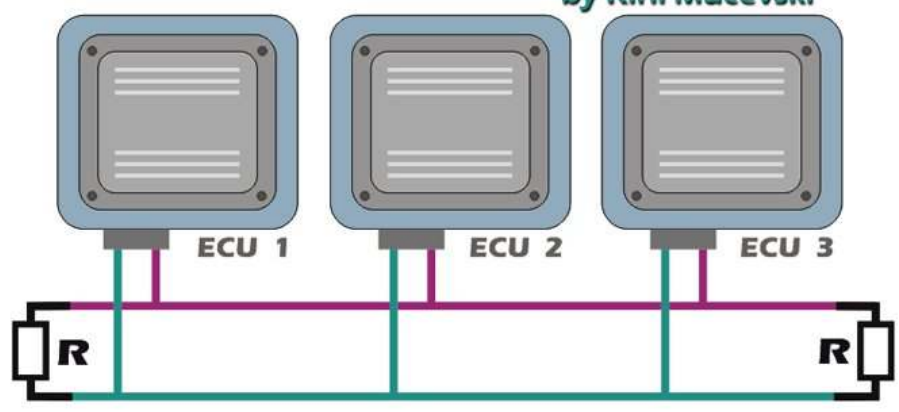


Automotive CAN Bus System

by Kiril Mucevski



Designed by Kiril Mucevski

Automotive CAN Bus System Explained

Published on December 8, 2015



Kiril Mucevski Follow
Vehicle Inspector and Conformity Analyst at Moto AMERIT

2,265

222

301

Automotive Controller Area Network System

The Controller Area Network (CAN, also known as CAN Bus) is a vehicle bus standard designed to allow electronic control units and devices to communicate with each other in applications without a host computer. As an alternative to conventional multi-wire looms, CAN Bus allows various electronic components (such as: electronic control units, microcontrollers, devices, sensors, actuators and other electronic components throughout the vehicle) to communicate on a single or dual-wire network data bus up to 1 Mb/s.

The CAN Bus is a message based protocol, designed originally for multiplex electrical wiring within motor vehicles, but also can be used in many other contexts.

Figure 1 shows a simplified schematic diagram of the CAN system and shows some of the possible units/devices that can be connected onto the CAN bus.

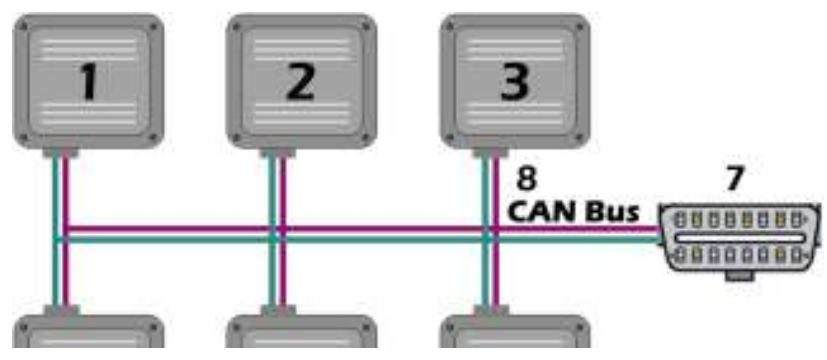




Figure 1. CAN System Schematic Diagram

1. Engine Management Electronic Control Unit
2. Transmission Electronic Control Unit
3. Anti-Lock Braking Electronic Control Unit
4. Traction Control Electronic Control Unit
5. Airbag Electronic Control Unit
6. Power Steering Electronic Control Unit
7. On-Board Diagnostic (OBD) Connector
8. Controller Area Network (CAN Bus)

CAN Bus Components and Output Signal

Typically the CAN Bus is made up two wires, CAN-H (CAN High) and CAN-L (CAN Low) which connect to all the devices in the network. The signals on the two CAN lines have the same sequence of data, but their amplitudes are opposite. So if a pulse on the CAN-H line goes from 2.5V to 3.75V then the corresponding pulse on the CAN-L line goes from 2.5V to 1.25V (opposite than CAN-H). By sending the data in equal and opposite ways like this allows for greater noise immunity and therefore less chance of the data being corrupted.

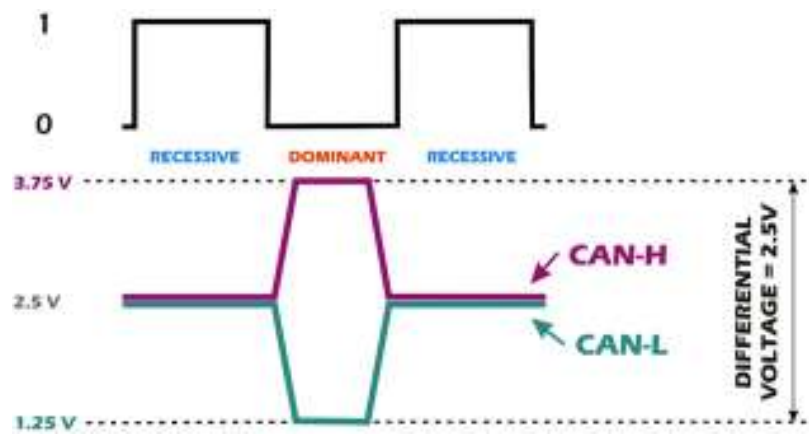


Figure 2. CAN-H and CAN-L Differential Output

Status of bit with the value 0 = 2.5V differential voltage = dominant state
 Status of bit with the value 1 = 0V differential voltage = recessive state

Figure 3 shows an example of two connected devices as well as the components that make up a CAN Bus: CAN controller, CAN transceiver, two data bus terminals (R) and two data bus lines (CAN High line and CAN Low line).





Sign in

Join now



Figure 3. CAN Bus Components

CAN Controller receives the transfer data from the microcomputer integrated in the control unit/device (also known as CAN Node). The CAN controller processes this data and relays it to the CAN transceiver. Also, the CAN controller receives data from the CAN transceiver, processes it and relays it to the microcomputer integrated in the control unit/device (CAN Node).

CAN Transceiver is a transmitter and receiver in one. It converts the data which the CAN controller supplies into electrical signals and sends this data over the data bus lines. Also, it receives data and converts this data for the CAN controller.

CAN Data Bus Terminal is a resistor (R) typically of 120 ohms. It prevents data sent from being reflected at the ends and returning as an echo.

CAN Data Transfer Process and Message Structure

The CAN data transfer process consists the following stages: supplying data, sending data, receiving data, checking data and accepting data.

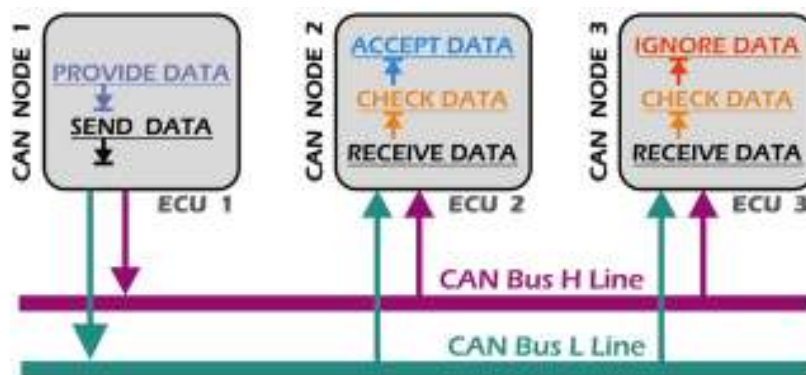


Figure 4. CAN Bus Data Transfer Process

Supplying Data: The CAN Node provides data to the CAN controller for transfer.

Sending Data: The CAN transceiver receives data from the CAN controller, converts it into electrical signals and sends them back into the network.

Receiving Data: All other CAN Nodes networked with the CAN data bus become receivers.

Checking Data: The CAN Node checks whether they require the data they have received for their functions or not.

[Sign in](#)[Join now](#)

View the full profile of Kiril Mucevski

There are two different ISO standards for CAN systems that relate to the physical layer: ISO 11898-3 low speed CAN up to 125 kb/s (distance up to 500 m) and ISO 11898-2 high speed CAN up to 1 Mb/s (distance up to 40 m).

The CAN system is further divided into two formats for the message frames 2.0A and 2.0B, the two standards differ in the size of the identifiers (ID):

Standard CAN (version 2.0A) uses 11 bit identifiers in the arbitration field.

Extended CAN (version 2.0B) supports a length of 29 bits for the identifier, made up of the 11 bit identifier (base identifier) and an 18 bit extension.

Figure 5 shows the CAN Bus data message structure. The basic structure of the message is the same for both standard and extended version.

Figure 5. CAN Bus Data Message Structure

SF = Start Field: indicates the beginning of a message with a dominant bit. So this field marks the start of the data protocol. A bit with 3.75 V (depending on the used system) is sent over the CAN-H line and a bit with 1.25 V is sent over the CAN-L line, i.e. the differential voltage is 2.5V (see figure 2).

Message Identifier: defines the level of priority of the data protocol. If, for instance, two CAN Nodes want to send their data protocol simultaneously, the CAN Node with the higher priority takes precedence. The lower the value the higher the priority of the message. As stated earlier, depending on the standard being used, the length of the frames can be in two formats: standard, which uses an 11 bit arbitration ID, and extended, which uses a 29 bit arbitration ID.

Control, also known as **Check Field:** displays the number of items of information contained in the data field. This field allows any receiver to check whether it has received all the information transferred to it.

Data Field: in this field the information is transferred to the other CAN Nodes.

CRC = Cyclic Redundancy Check, also known as **Safety Field:** Contains 15 bit cyclic redundancy check code and a recessive delimiter bit. The CRC field is used for transfer faults detection.

ACK = Acknowledge Field, also known as **Confirmation Field:** In this field the receivers signal to the transmitter that they have correctly received the data protocol. If an error is detected, the receivers notify the transmitter of this immediately. The transmitter then sends the data protocol again.

[Sign in](#)[Join now](#)

Summary: How the CAN System Works

As stated earlier, at the CAN system there is no master that controls when individual CAN Nodes have access to read and write data on the CAN Bus. When a CAN Node is ready to transmit data, it checks to see if the Bus is busy and then simply writes a CAN frame (message) onto the network. The CAN frames that are transmitted do not contain addresses of either the transmitting node or any of the intended receiving node(s). Instead, an arbitration ID that is unique throughout the network labels the frame.

All CAN Nodes on the CAN network receive the CAN frame, and depending on the arbitration ID of that transmitted frame, each CAN Node on the network decides whether to accept or ignore the received frame.

If multiple CAN Nodes try to transmit a message onto the CAN Bus at the same time, the node with the highest priority (lowest arbitration ID) automatically gets Bus access. Lower-priority CAN Nodes must wait until the Bus becomes available before trying to transmit again. In this way, you can implement CAN networks to ensure deterministic communication among CAN Nodes.

The table below shows an example of how the information of engine coolant temperature can be transferred with 1, 2 and 3 consecutive bits and respectively 2, 4 and 8 possible variations:

The higher the number of bits, the more items of information can be transferred. The number of possible items of information doubles with each additional bit.

Thank you for reading my article! If you enjoyed reading this or found useful, please let me know, provide your feedback or just click the like button.

If you would like to read similar automotive articles, repair tips or testing procedures, you are welcome to visit my [LinkedIn article page](#):

- [OBD-II Connector and Fault Codes Explained](#)
- [Inductive and Hall Effect RPM Sensors Explained](#)
- [Ignition System with Inductive Sender](#)
- [Ignition System with Hall Effect Sender](#)
- [Engine Oil Pressure Switch Explained](#)
- [Fuel Injector Operating Principles and Diagnostics](#)
- [Automotive Relays Fundamentals and Testing](#)