The Process of Transmitting Data Over the Controller Area Network (CAN) Bus

Zak Rowland
April 29th, 2020
Oregon Institute of Technology
Advanced Technical Writing – WRI327

The Process of Transmitting Data Over the Controller Area Network (CAN) Bus

Introduction

The Controller Area Network bus, better known as the CAN bus, is the almost universally standard serial bus protocol for vehicles that allows computers and other electronics to communicate with each other without a primary computer that processes the data first. The protocol is used in a variety of applications including normal passenger vehicles, aviation, and other industrial automation systems. In a passenger vehicle for example, the CAN bus is used to transmit data from all the various sensors to the computers on the bus that need it, such as the Engine Control Unit (ECU.) Learning the process of how data is transmitted over the bus is important to those interested in fields including automotive, aerospace, industrial automation, and more.

History

Before the CAN bus, computers in vehicles required direct wiring to the various sensors and electronics. This was not a problem at first, but as the systems became more complex, the wiring became even more complicated. In 1986, Robert Bosch developed the CAN protocol to solve this issue (CSS Electronics, 2020.) After a revision, CAN 2.0, the protocol became an international standard. CAN is still standard today and engineers continue to look for ways to improve the protocol to allow for more data at faster rates.

The Physical Connections

A CAN bus consists of just two wires, CANH (CAN high) and CANL (CAN low.) These wires are a twisted pair that use differential signaling. Put simply, differential signaling means that the receiving circuits reads data from the difference between the two signals instead of the difference between each signal and ground. The signals are the same except one is inverted, which means when CANH is driven high, CANL is driven low at the same time. When the bus is not being driven, the voltage of CANH is equal to or even less than CANL. This is known as the recessive state, which is read as a ‘1’ by devices on the bus. The dominant state, when the voltage of CANH is driven high and CANL is driven low, is read as a ‘0’ by devices on the bus.

Nodes on the Bus

All the devices attached to the bus are called also called nodes, and each node requires a CAN transceiver, a CAN controller, and a processor. The CAN transceiver reads the analog CANH and CANL voltages, then outputs the appropriate digital signal to match, as well as reading a digital signal to transmit back to the bus. The CAN controller, which is usually integrated with the processor as part of a microcontroller, reads and writes the serial data to and from the transceiver. Finally, the processor interprets the data and reacts accordingly. A visual representation of a node on a CAN bus can be seen in Figure 1.

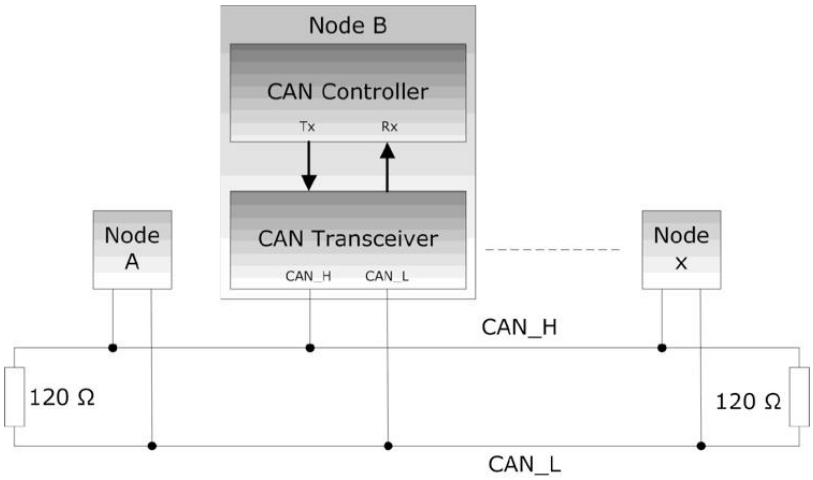

Figure 1. The components of a CAN node, from Copperhill Technologies.

Transmitting Data

The nodes on the bus transmit data in frames, and there are two different formats which are base frame and extended frame. The only difference between base and extended frame format is that extended frame format contains 29 identifier bits instead of 11. The identifier bits let other nodes on the bus know who sent the message and the priority of the message. A base format data frame begins with a start of frame bit, which is driven dominant (0) to indicate the start of a frame. The 11 identifier bits are transmitted next to identify the sender and priority. Next, the Remote Transmission Request bit is sent next. This bit is used to indicate if the node is sending data or requesting data. The next bit, called the Identifier Extension Bit (IDE,) is dominant if 11-bit identifiers are used. A reserved bit comes after the IDE bit, which means it can be either dominant or recessive. Next is 4 bits of data that contain the number of data bytes being transmitted in the CAN frame, from 0 to 8 bytes, called the Data Length Code (DLC.) These 6 bits make up the control section seen in Figure 2 below. After the control bits comes the actual data bits, which can be up to 64 bits (8 bytes) in length. An example of when the data section would likely be 0 bits is if a node is requesting data instead of sending it. 16 bits is used for error correction, which comes after the data bits, known as a Cyclic Redundancy Check (CRC.) The next two bits are known as the acknowledgement (ACK) bits. A transmitter will send a recessive (1) bit, and a receiver will assert the bit (0.) The second bit of ACK will always be recessive. Finally, 7 recessive bits marks the end of the CAN frame.

Figure 2. The bits that make up a CAN frame, from CSS Electronics.

Conclusion

CAN is a valuable protocol to have knowledge of, especially for those interested in embedded systems. The protocol is used in a wide variety of fields and is still being improved upon, which only increases its value. In the future, CAN will have the ability to transfer more data at much faster speeds. Not only will this increase efficiency, but this allows for even more nodes to be attached to the bus. It will be interesting to see what kind of vehicles make use of this bus in the future.

References

CAN bus. (2020, April 25). Retrieved April 29, 2020, from https://en.wikipedia.org/wiki/CAN_bus

CAN Bus Explained – A Simple Intro (2020). Retrieved April 29, 2020 from https://www.csselectronics.com/screen/page/simple-intro-to-can-bus/language/en

Pinkle, Carsten. "The Why and How of Differential Signaling - Technical Articles." All About Circuits, 16 Nov. 2016, www.allaboutcircuits.com/technical-articles/the-why-and-how-of-differential-signaling/

A Brief Introduction to Controller Area Network. Retrieved May 5, 2020 from https://copperhilltech.com/a-brief-introduction-to-controller-area-network/