# Completion Report

Zak Rowland
Oregon Institute of Technology
June 10th, 2020

Author Note:

This report is written for employers who hire engineers in fields that work with the CAN bus specification. The information could be used as an introduction to these interested engineers.

## Table of Contents

# Table of Figures

## Executive Summary

The CAN bus, or Controller Area Network, is a serial bus specification invented by Bosch used in a variety of embedded system applications including automobiles, airplanes, and controlling industrial machinery. This report includes information about the process of creating a technical description and technical instructions for the CAN specification. The scope of the project was to describe the process of transmitting data over the bus, then create a set of instructions on how to decode a frame of this data. Creating an introduction to the subject for those interested and learning about it myself was the primary goal of the project. There is an overwhelming amount of information on the specification including tutorials, manuals, forum posts, videos, and more, so finding all the information needed to create the documents was not an issue. These documents are intended to give those interested in a field that utilizes CAN an introduction to the specification.

Completion Report

## Introduction

From the automotive and aerospace industries to industrial machine automation, the CAN bus standard is a widely used serial bus protocol. CAN stands for Controller Area Network, and it is a technology used to provide a communication pathway to the various devices connected in an embedded system; these connected devices will also be referred to as "nodes." A usage example of the bus can be seen in Figure 1 below.

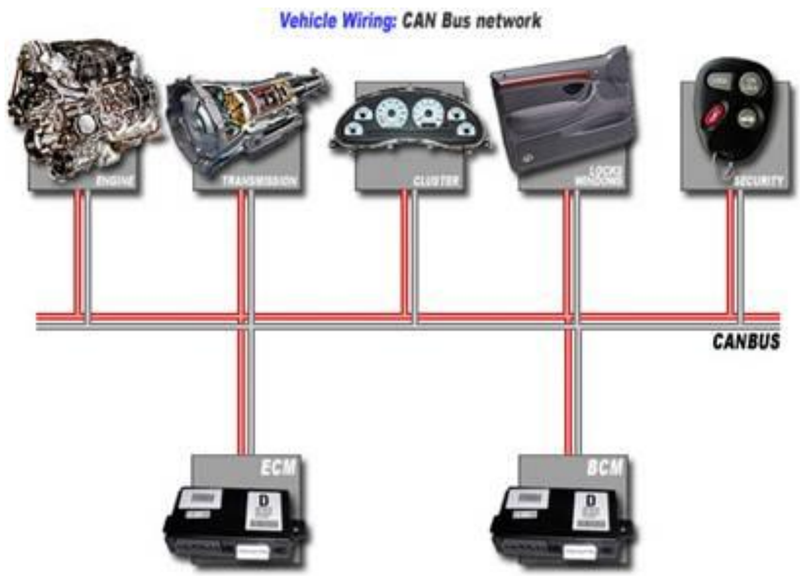



*Figure 1. A diagram of how a CAN bus is used in a vehicle*

The content of this report should serve as an introduction to the CAN bus to those interested in engineering the embedded systems that utilize the bus; see Appendix A and Appendix B for the technical description and instructions involving CAN. This report includes the projects scope and goals, research and inquiry methods, the findings from these methods, and the challenges and learning experiences that came along with the project.

## Project Scope and Goals

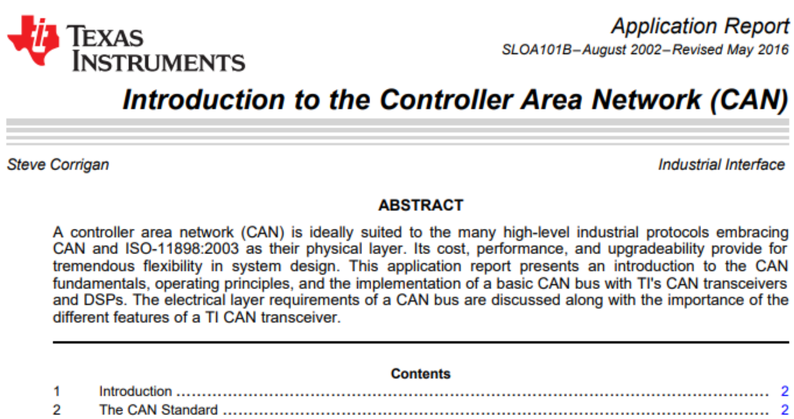
The topic of the technical description is the process of transmitting data over the CAN bus. This description briefly discusses the history of the bus standard before discussing the physical characteristics and data transmission process. The instructions continue the exploration of the bus by presenting the steps required to read and interpret a piece of data from an actual CAN bus, in this case a passenger car. Vehicles sold in the United States from the year 1996 and beyond are equipped with a port called OBD-II (On-Board Diagnostics, version 2.) This port conveniently has two pins for the CAN bus built into it, so anyone with a vehicle that fits that requirement has access to a CAN bus of their own. The goal of these documents is to introduce this commonly used technology to engineers and scientists interested in the fields that utilize it, so they have knowledge of the standard before entering the field.

## Methods of Inquiry and Research

Research naturally began with learning about the CAN bus myself, as I had no prior knowledge of it. I accomplished this by reading various informative websites, the Wikipedia entry, manuals from manufacturers, and parts of the 70-page manual from the inventors of the specification itself, Bosch. The amount of information can be overwhelming, however a background in computers, embedded systems, or electronics will help make the material much easier to understand. Once I understood what the specification was, I began learning how one might decode a message from the CAN bus. This was done by reading many tutorials on the internet and looking for other needed information in technical documents, like the Bosch manual mentioned earlier.

## Findings from Inquiry and Research

Since the CAN bus is a widespread standard, finding information about it did not prove to be difficult. Many different companies, universities, and other websites provide a plethora of information in their own style. Even Texas Instruments, widely known for their calculators, has a 15-page document on the subject, a snippet of which can be seen in Figure 2 below.



TEXAS INSTRUMENTS

*Application Report*

*SLOA101B–August 2002–Revised May 2016*

***Introduction to the Controller Area Network (CAN)***

*Steve Corrigan* *Industrial Interface*

**ABSTRACT**

A controller area network (CAN) is ideally suited to the many high-level industrial protocols embracing CAN and ISO-11898:2003 as their physical layer. Its cost, performance, and upgradeability provide for tremendous flexibility in system design. This application report presents an introduction to the CAN fundamentals, operating principles, and the implementation of a basic CAN bus with TI's CAN transceivers and DSPs. The electrical layer requirements of a CAN bus are discussed along with the importance of the different features of a TI CAN transceiver.

**Contents**

1 Introduction .......... 2
2 The CAN Standard .......... 2

*Figure 2. A snippet of a report on the CAN bus from Texas Instruments*

This wealth of information made it much easier to describe and create a set of instructions for the CAN specification. Without these resources, countless hours would be spent decoding these signals manually, trying to figure out which pieces of data mean what in the frame data.

### Project Challenges and Learning Experience

The greatest challenge of the project was simply learning all the new things needed to understand the CAN bus well enough to explain it. There are many things about the specification to learn, such as the history, applications, physical properties, functionality, etc. However, since my major is involved in this area, the documentation and material looked familiar. This made it easier to read through and understand the most important information. Another unique problem I had was the lack of lab equipment to test my instructions due to COVID-19, however I still managed with what I had. My senior project next year will involve reading data from the OBD-II port, so learning about CAN was a valuable learning experience. Overall, I learned quite a lot about the specification and where it is used, which will be helpful in future projects and possibly even my career.

### Conclusion

I would like to thank [employer] for the opportunity to research and report on this topic. Not only will prospective engineers and scientists from a variety of fields find this information useful, the research I conducted in the process will be useful to me as well. The project was a great learning experience overall and I am glad to have written on this topic. As mentioned previously, the technical description and instructions for the CAN specification are located in Appendices A and B.

# References

[Figure 1] Autoditex, (No Date). CAN BUS (CONTROLLER AREA NETWORK). Retrieved from https://autoditex.com/page/can-bus--controller-area-network-34-1.html

[Figure 2] Texas Instruments, (2016). Introduction to the Controller Area Network (CAN). Retrieved from https://www.ti.com/lit/an/sloa101b/sloa101b.pdf

Bosch, (1991). CAN Specification. Retrieved from http://esd.cs.ucr.edu/webres/can20.pdf

Appendix A

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.

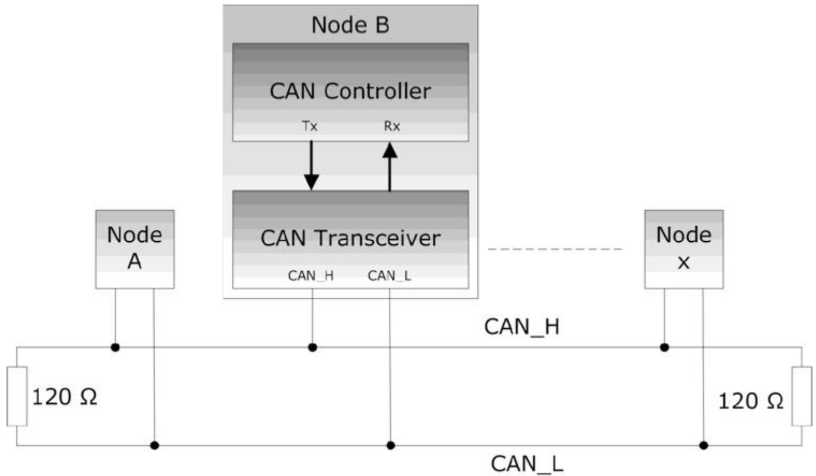



***Figure 1. The components of a CAN node, from Copperhill Technologies.***

**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.

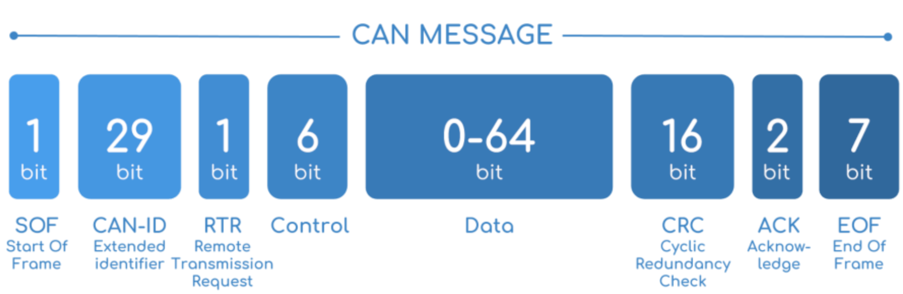



***Figure 2. The bits that make up a CAN frame, from CSS Electronics.***

**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/

# Appendix B

How to Interpret Data from the Controller Area Network (CAN) Bus

Zak Rowland
May 17th, 2020
Oregon Institute of Technology
Advanced Technical Writing – WRI327

## How to Interpret Data from the Controller Area Network (CAN) Bus

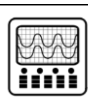
The CAN protocol is used in many industries including automotive, aerospace, industrial automation, and more. Understanding how to decode data from the bus can be valuable to anyone interested in these fields because it is so widely used. In addition, it can also be useful for mechanics and hobbyists who tinker with their vehicles at home.

*Caution*

There is always a risk of electric shock or damage to components when working with electronics. It is recommended to have previous electronics and oscilloscope experience before attempting to read data from the CAN bus.

**Useful Terms to Know**

| **Term** | **Definition** |
|---|---|
| Controller Area Network (CAN) | A vehicle bus standard that allows the various computers and devices to communicate with each other without going through a host computer. |
| Bus | An electrical connection that is shared by multiple different devices to transmit and receive data. |
| OBD-II | The On-Board Diagnostic port equipped on all vehicles of model year 1996 and beyond. |
| Oscilloscope | A tool used to display various electrical signals. |
| Trigger | The setting in the oscilloscope that marks the start of the signal capture. |
| Bit | A single piece of data used in electronics, either a ‘1’ or ‘0’. |

| **You Will Need** | |
|---|---|
|  | About 30 minutes |
|  | A vehicle with an OBD-II port (1996 and newer) |
|  | A portable oscilloscope with at least 2 channels |

1. With the vehicle turned off, locate the OBD-II port. The port is usually located somewhere below the steering wheel as seen in Figure 1.


*Figure 1. The OBD-II port location in a 2008 Nissan Altima.*

2. Determine which pins are CAN HI (CANH), CAN LO (CANL), and Ground. This may vary, however in most cases pin 6 is CANH and pin 14 is CANL. An example of the pin locations can be seen in Figure 2.

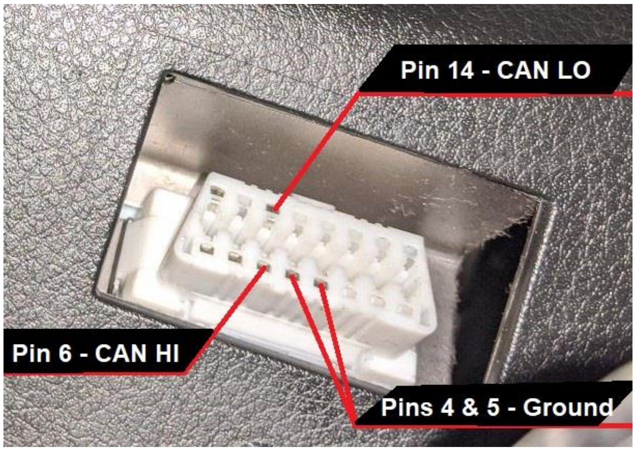



*Figure 2. The ground, CAN HI, and CAN LO pins in a 2008 Nissan Altima.*

3. Using the oscilloscope, probe the CANH pin on one channel and the CANL pin on another channel. The scope probes can be grounded at one of the ground pins or somewhere on the chassis of the vehicle.

4. Set the CANH probe to trigger on the rising edge and the CANL probe to trigger on the falling edge. Figure 3 illustrates how the pair of signals work together to form the data bus.

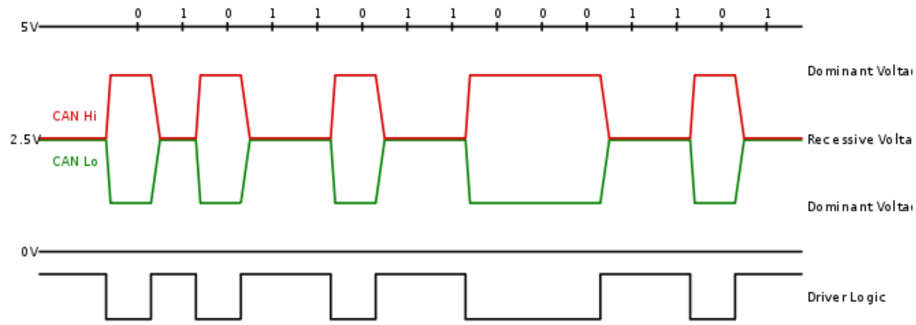



*Figure 3. An example of the CAN protocol.*

5. Start the vehicle and capture a frame of data from the bus with the oscilloscope. The format of a typical CAN frame can be seen in Figure 4.

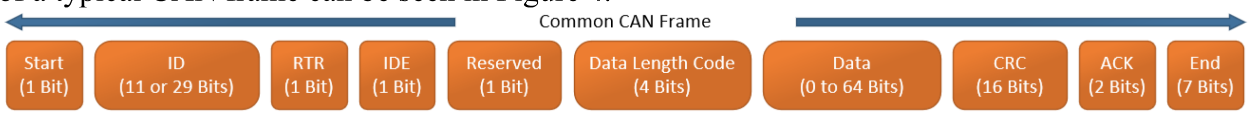



*Figure 4. The format of a typical CAN frame.*

6. Identify the start of frame bit (CANH driven high and CANL driven low,) then determine the frame ID. The frame ID indicates which device sent the message, and it will be either 11 or 29 bits. The start of a CAN frame can be seen in Figure 5.

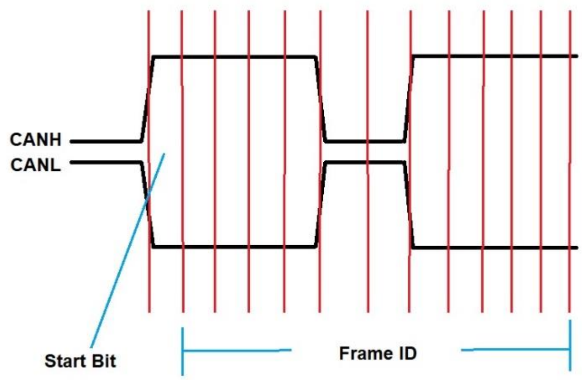



*Figure 5. The start of a CAN message and decoding the ID.*

7. Determine if the frame is sending or receiving data by locating the Remote Transmission Request (RTR) bit which comes after the frame ID. Note: The IDE and Reserved bits that come after can be ignored.

8. Determine how many bytes of data are in the message by decoding the 4 data length bits.

9. Depending on the data length bits, decode the data included in the message.

10. Locate and decode the 16 bits used for the Cyclic Redundancy Check (CRC.)

11. Identify the 2 acknowledge bits (ACK) and the 7 bits that mark the end of the frame.

**Conclusion**

Once the 7 end of frame bits are identified, the CAN message has been decoded! The data likely look like gibberish to you; however, it is important to whatever device on the bus is intended to read it.

## References

[Icon of a portable oscilloscope] (2020). Retrieved May 8, 2020, from https://images.app.goo.gl/A2QCimHjo9A5xDbb9

[CAN protocol tutorial] (2020). Retrieved May 15, 2020, from https://images.app.goo.gl/8XhMWw5FFtkmKAch6

Nuñez, Ariel. "An Introduction to the CAN Bus: How to Programmatically Control a Car" Voyage, June 4, 2016. Retrieved May 8, 2020, from https://news.voyage.auto/an-introduction-to-the-can-bus-how-to-programmatically-control-a-car-f1b18be4f377

CAN Bus Explained – A Simple Intro (2020). Retrieved May 17, 2020 from https://www.csselectronics.com/screen/page/simple-intro-to-can-bus/language/en