CAN Tech Tip 3 – November/December 2013

Welcome to our 3rd “Tech Tips” feature (Sort of a Christmas issue). Last month we covered the communication levels of the CAN bus, where we discussed the Physical layer portion. Here we showed the appearance of a CAN frame on the physical wire.   This edition will show how the CAN frame is packaged before sending over the physical bus.

The CAN frame

Referring to Figure 1, we showed you the appearance of CAN frame on a two channel oscilloscope. Here we illustrated the sections of the CAN frame relating to this physical data stream.

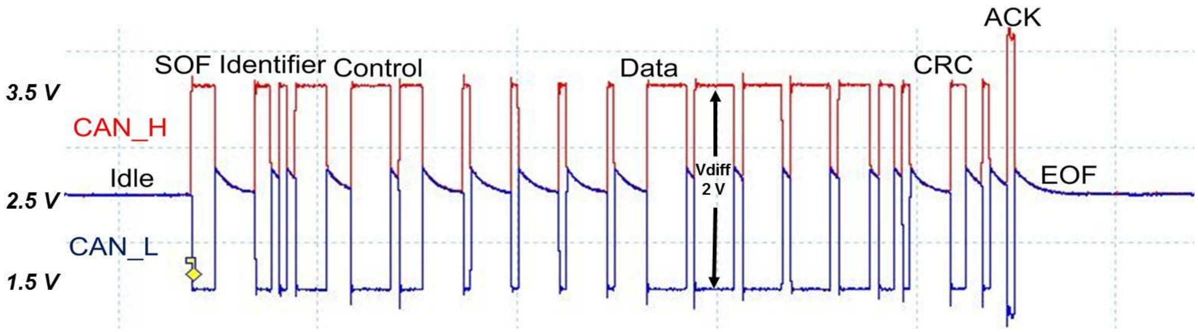


Figure 1. CAN Waveform on Oscilloscope

This newsletter will cover each section of the CAN frame, and how it is packaged for sending over the CAN bus.

The CAN Frame Message Format

When a microcontroller formats information for sharing with other ECUs, it will pass the data to the CAN chip for framing the relevant information to send onto the other ECUs. Here we will look at how the CAN interface puts the information into a frame to transmit over the CAN bus.The CAN message format simplistically appears as the block diagram in Figure 2 below:

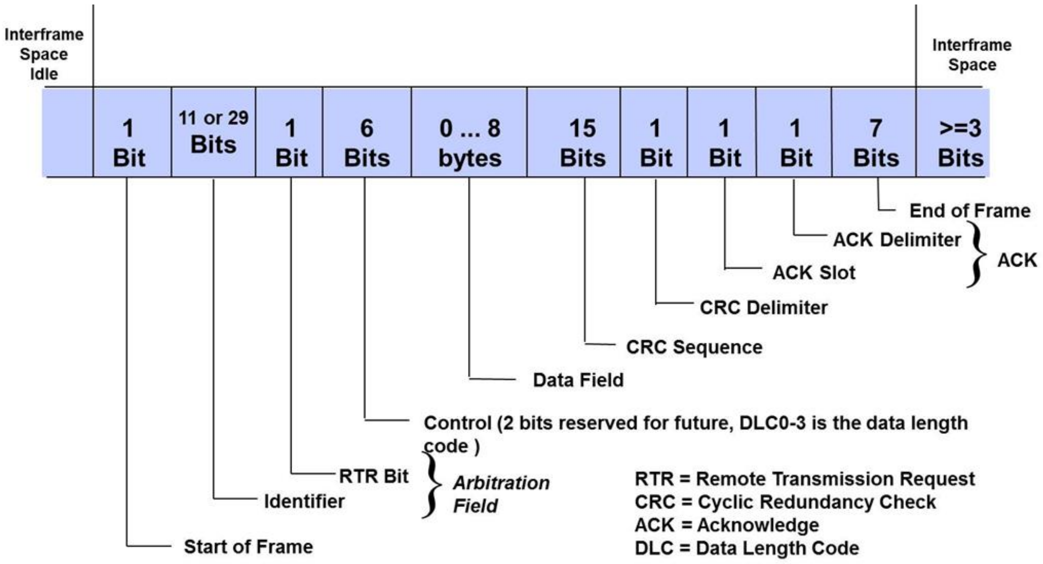


Figure 2. CAN Message frame format

The Start of Frame (SOF) is a dominant 0 to tell all the other ECUs that a message is on the way.

Following this is the **Identifier field (ID)**. This is usually a functional address (e.g. Engine parameters, Wheel speeds, etc.). In some applications, the ID can contain source and destination addresses. The value of the Identifier determines the message priority. The lower the value, the higher the priority.

The Identifier value can be 11 bits or 29 bits. This is known as Standard CAN and Extended CAN. The original CAN specification designated 11 bit identifiers. In 1991, the CAN specification was updated to include this extended CAN format. Looking at the possible number of identifiers, 2 to the power of 11 = 2048 possible IDs. 2 to the power of 29 allows for more than 500 million possible IDs. The reason for the introduction of the extended frame format was not so much to increase the number of IDs, but to adapt CAN to the requirements of the American car-manufacturers. With the 29 bit identifier a message strategy can be realised, which is similar to the J1850-protocol published by the American Society of Automotive Engineers (SAE).

Many applications implement either. The SAE J1939 standard for Trucks and Buses dictates that the extended format is to be used. This is not so much to allow for more Identifiers, but to allow for versatile types of messaging. This will be covered in a later newsletter.

The **Remote Transfer Request (RTR)** bit allows an ECU to request an ID from another ECU that has not been transferred within a specified time period. When an RTR message is sent, this bit is set to Recessive (Data 1). During normal messages, the RTR bit is always set to Dominant (Data 0).

The ID and RTR are considered to be known as the Arbitration Field, as this determines the message priority during a message arbitration. This occurs when two or more messages collide during bus access.

The **Control Field** contains the **Data Length Code (DLC)**. The DLC simply informs of the length of the data field. This can be a value between 0 and 8 Bytes. It is always in Bytes. Even if the node wants to send just one bit, a Byte must be assigned. There can be 0 Byte frames. The RTR message is a 0 Byte message. In some protocols, a 0 Byte message can be used as health/heartbeat message.

The **Data Field** contains the information to be relayed to other ECUs, e.g. the Engine Parameters message may contain info such as Engine speed, Water & Oil Temp, etc. The format of the data and how it is interpreted is covered in a later newsletter.

The **Cyclic Redundancy Check (CRC)** is a 15 bit sequence used for checking the integrity of the data. Sometimes known as Check Sum, it is a common error checking method in digital communication. The way it works is that the transmitting node will generate a 15 bit digital sequence based on the prior data (ID, Control, and Data fields). Once the frame is brought into the receiver node, that node will generate its own CRC sequence based on the received info (ID, Control, and Data fields).

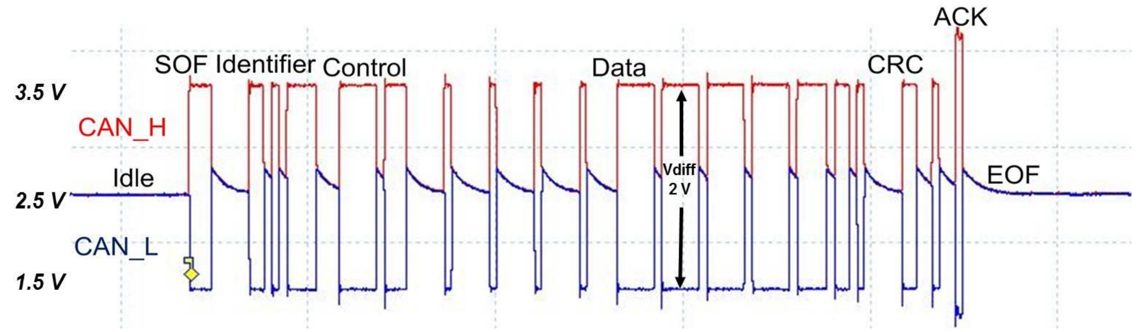
The receiver's CRC is then compared with the transmitted CRC. If there is a match, the data is deemed OK, and the Acknowledge is generated. If they do not match, there has been corruption of the data, and the receiver node will generate an Error Frame (to be covered later).

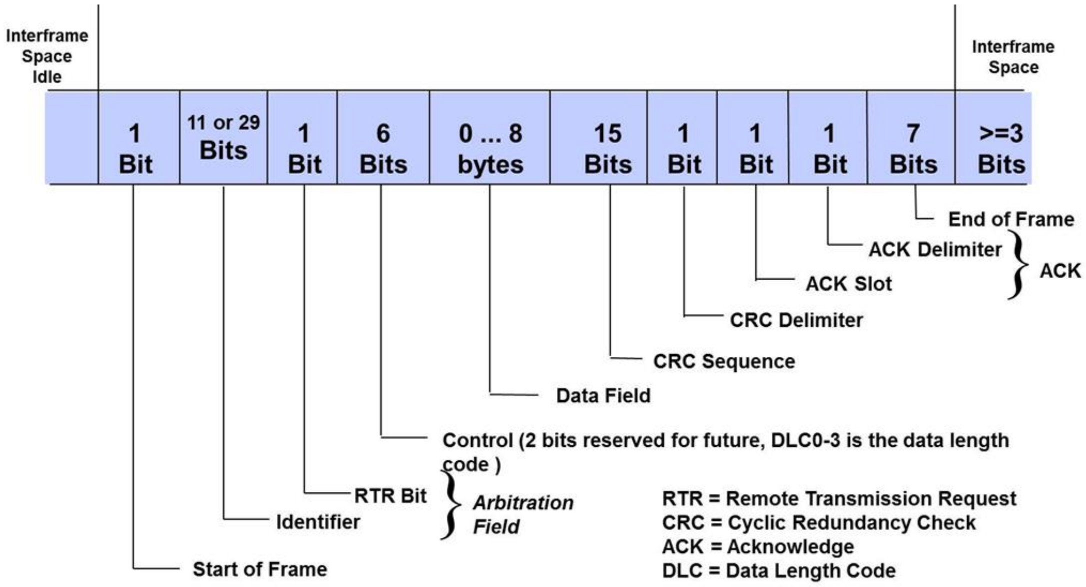
The **Acknowledge (ACK)** slot is a single bit generated by all receiving CAN node to indicate that their CRC processes are OK. The transmitter node leaves this bit recessive (Data 1), and the receiver nodes all insert a dominant bit (Data 0) in this slot. This indicates that all receivers are OK with the integrity of the CAN frame data. If there was a problem within the CRC process, a node will generate an Error Frame, negating any ACK process (covered later).

Delimiters. Note that the CRC and the ACK fields have a one bit Delimiters. These are just to allow a space for the next process.

End-of-Frame (EOF). EOF is a seven bit recessive sequence at the end of the CAN frame. This, along with the minimum of 3 bits of Interframe Space, give enough time delay to allow for the next access of CAN messages. If there are back-to-back CAN messages, you will observe a minimum of 11 bits of Recessive before the next CAN frame. These 11 bits are comprised of ACK delimiter, EOF and 3 bits Interframe space.

To compare these fields with the physical signal on the oscilloscope, here we show the two figures together.





The next CAN Tech Tip will cover the access method of the CAN bus, and how the protocol compensates for message collisions. It will be out soon after the New Year's celebrations.

Happy Holidays from the team at Warwick Control.

Common Acronyms:

ABS – Anti-skid Braking System
ARZ – Always Return to Zero
CAN – Controller Area Networks
CAN_H – CAN High
CAN_L – CAN Low
CARB – California Air Resource Board
DIS - Draft International Standard
ECU – Electronic Control Unit
Kbps – Kilobits per Second
I/O – Inputs/Outputs of microcontrollers
ISO - International Standards Organisation
Mbps – Megabits per Second
NRZ – Non Return to Zero
OBD – On Board Diagnostics
OSI - Open Systems Intercommunication
PWM – Pulse Width Modulation
RPM – Revolutions per Minute