CAN vs. RS-485: Why CAN Is on the Move

By Robert Gee, Executive Business Manager, Core Products Group, Maxim Integrated

Both RS-485 and controller area network (CAN) interface protocols have been around since the mid-1980s, when they were introduced as communication standards. RS-485 was an evolutionary step from previous transceiver (physical layer) standards like RS-423, RS-422, and RS-232. RS-485 enabled systems to have multiple master nodes in a single system. Around the same time as these commonly used interfaces were being used in applications such as computer keyboards and mice, printers, and industrial automation equipment, the CANbus interface was being developed as an automotive communication platform (by Robert Bosch GmbH) to reduce cost in automobile manufacturing. It was considered an alternative to the conventional, large multi-wire looms needed in automobiles, simplifying cabling and taking advantage of multi-node buses. In fact, when it was first introduced on a BMW 850 in 1986, the automotive CAN interface saved over 2km of cabling! On top of that, the number of connectors was significantly reduced and the estimated weight savings in cable and connector alone was 50kg[1]. RS-485 was defined for the industrial market, while CAN was primarily developed for the automotive/vehicle/transportation segment. Since its release, the CANbus interface has slowly been adopted to applications outside of the automotive and aerospace industries.

Due to its robustness in harsh electrical environments, fault protection capabilities, and unique message handling, CANbus is being adopted into many applications where it has never been used. Current market trends show an ever-increasing adoption of CANbus, even replacing the RS-485 in traditional industrial applications. According to available market reports, CANbus usage is growing at a high single-digit rate, which is exceptional for the interface market. Although reports do not split out the industrial and automotive markets, many agree that industrial markets make up around 20-30% of the total market units. Growth within the automotive industry can be attributed to the increase of electronics used in vehicles today. Modern automobiles have complex microprocessing systems used for functions like back-up cameras, self-parking, infotainment, blind spot awareness, and more. The emergence of these automotive sub-systems stems from the increasing number of in-vehicle sensors and microcontrollers needed to handle all of the complex systems within an automobile. Back in the '90s, many auto manufacturers started transitioning from mechanically controlled automatic transmission shift points to electronically controlled shift points based off of data collected by speed, throttle position, and barometric sensors and fed to the microcontroller. Today, there are more than 100 sensors and microcontrollers per vehicle, with many of them talking CAN. Even the full electric vehicle Tesla S has 65 microcontrollers[2].

In the industrial market, CAN adoption is also increasing. Industrial CAN applications have a wide reach and are used anywhere from commercial drones to elevator lift controls to commercial-grade lawn mowers. IC suppliers are recognizing this and developing products to address the ever-increasing need for CAN outside of the automotive market. Other contributors to the rising use of CAN in the industrial space could be the theory that many automotive engineers have moved over to the industrial segment, bringing with them the expertise of CAN and its unique benefits. There's certainly credence to this theory, especially given that the job markets in Detroit and other automotive mainstays were weak due to US recessions during the early 1990s and early 2000s. Another reason for CAN adoption in the industrial market is due to its inherent fault tolerance and the way it handles frame messages on a multi-node bus.

To explain the advantages of CAN over RS-485, it is best to go over the similarities and differences between the two standards, the ISO-11898-2 and TIA/EIA-485, respectively. Both standards define the electrical components of the transceivers and are represented in the diagram (Figure 1) below for the transmit side.

Both protocols feature differential outputs. The RS-485 output is a classical differential signal where one signal is the inverted, or mirror, version of the other. Output A is the non-inverting line and output B is the inverting line. The differential range from +1.5V to +5V is a '1' or mark and -1.5V to -5V is a '0' or space. The area between -1.5V and +1.5V is undefined. It's good to note that when RS-485 is not driven, it is in a high impedance state. For CAN, the output differential is slightly different where the two outputs, CANH and CANL data lines, are a reflection

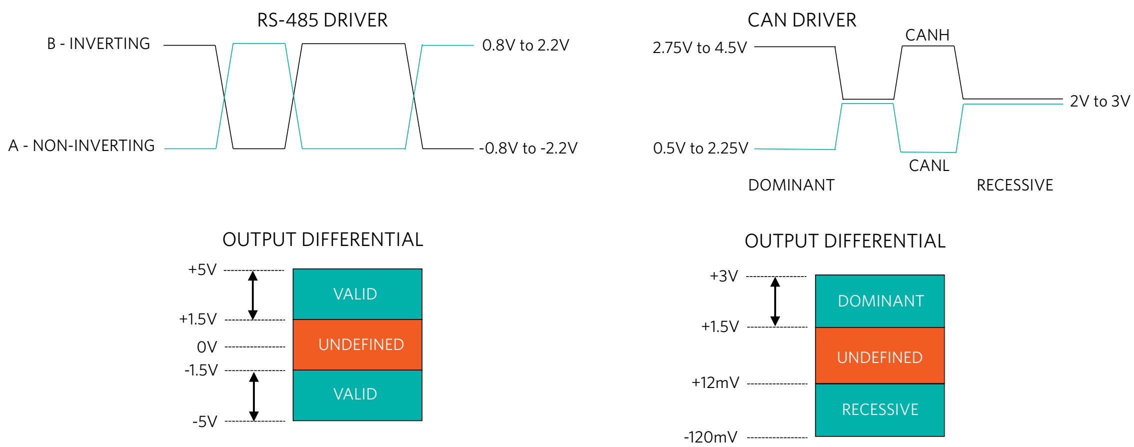

Figure 1. Comparison of Output Differentials for RS-485 and CAN Drivers

of each other as depicted and represent opposite logic. In the dominant state (a zero bit, used to determine message priority), CANH-CANL are defined to be logic '0' when the voltage across them is between +1.5V and +3V. In the recessive state (a 1-bit and the state of the idle bus), the driver is defined to be logic '1' when differential voltage is between -120mV and +12mV, or when it is near zero. For the receiver side, the RS-485 standard defines the input differential to be in between ±200mV to +5V. For CAN, the input differential signal is between +900mV and +3V, while the recessive mode is in between -120mV and +500mV. When the bus is idle or when it's not loaded, the transceiver is in a recessive state where CANH and CANL must be between 2V and 3V. Both RS-485 and CAN have room for margin in applications where the signal can be attenuated by the quality (shielded or unshielded) or length of the cables, which may affect the capacitance of the overall system. See Figure 2 for a comparison of receiver input differentials for RS-485 and CAN receivers.

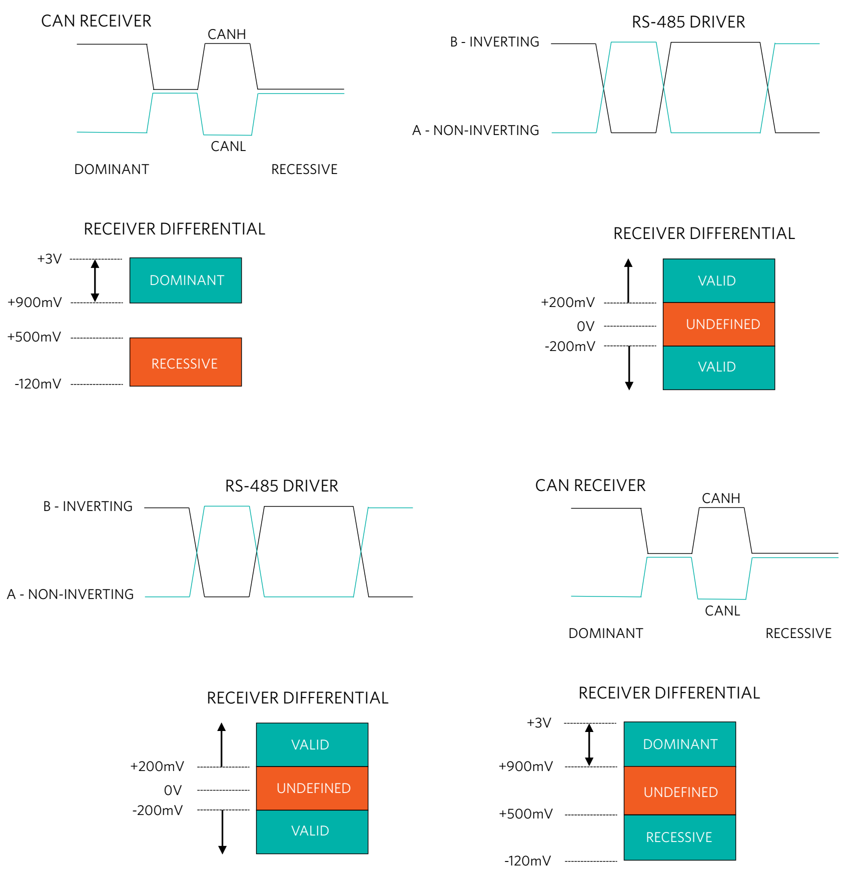

Figure 2. Comparison of Receiver Input Differentials for RS-485 and CAN Receivers

Additionally, both standards have termination resistors of the same 120-Ω value at the ends of the network, to match the characteristic impedance of the transmission line and avoid reflection. Other specifications, such as data rate and the number of nodes, are helpful references as opposed to strict parameters. Plenty of RS-485 and CAN transceivers exceed the standard in terms of bandwidth and the allowable number of nodes, in order to meet the demands of the market. RS-485, such as the MAX22500E from Maxim, has reached speeds of 100 Mbps. Even though the new CAN-FD standard, ISO 11898-2:2016, defines certain timing characteristics at 2Mbps and 5Mbps, the standard does not cap the data rate at 5Mbps. CAN transceivers will exceed the standard in the same way as RS-485 transceivers. The common-mode range (CMR) is -7V to +12V for RS-485 and -2V to +7V for CAN. Many applications need a wider CMR performance from both of these interface types. This is due to the fact that they are mainly used for multiple node buses that may use differently sourced power transformers, or because the cabling is in close proximity to equipment with large enough fields that can affect the grounding between systems. Given the many different harsh industrial applications, high CMR is often needed beyond the standard levels of just -7V to +12V. To address this problem, there are new RS-485 and CAN transceivers that have a wide common-mode range of ±25V. The diagram below shows a fluctuating common mode range of a RS-485 transceiver. Despite the common-mode voltage signal going up and down, as long as the common-mode voltage (V_{CM}) is within the proper range, the differential bus signal is not affected and the receiver is able to accurately receive the signal without degradation. The diagram in Figure 3 shows a varying common-mode range within the range for RS-485.

Another feature common in both CAN and RS-485 transceivers is fault protection. Fault-protected devices have an internal overvoltage circuit on the driver output and receiver inputs to protect the devices from accidental shorts between a local power supply and the data lines of the transceivers. Maxim ICs provide industry- leading fault protection levels of ±80V with even some extra margin before breakdown of the protection, and this level of protection is present whether the transceiver is powered or un-powered.

One of the major reasons for industrial applications to design in CAN versus RS-485 transceivers is how messages are handled on the bus. In a RS-485 system with many nodes communicating to the microprocessor, there may be instances where there are several messages sent out from multiple nodes onto a bus simultaneously that may result in a collision of messages, otherwise known as contention. When this happens, the bus state could possibly be invalid or indeterminate, causing data errors. Furthermore, contention could damage or degrade the signal performance when multiple RS-485 transceivers on the bus are in one state and one single transceiver is in the opposite state. In such a condition, the lone RS-485 would cause significant current draw that would likely cause thermal shutdown of the IC or permanent damage to the system This is where CANbus has a big advantage over the RS-485 protocol. With CANbus, there is a way to resolve multiple messages on the line by way of ranking each message. Prior to bringing the system up, different faults are assigned different priorities by the system engineer. Earlier, it was mentioned that CAN had a dominant and recessive state. During contention, the message with the most consecutive dominant state 'wins' and will continue to transmit, while other nodes with lower priority will see the dominant bit and stop transmission. This method is called arbitration, where the messages are prioritized and received in an order of status. A node that loses arbitration will resend its message. This continues for all nodes until there is one node left transmitting. Here, in Figure 4, is a closer look at the format of the CAN message data frame; the diagram and table below show where arbitration happens.

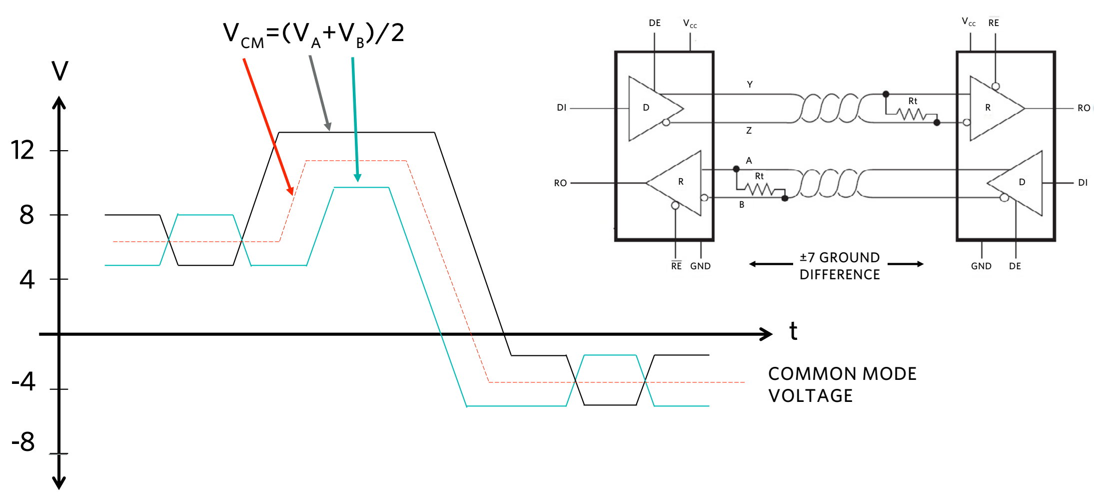

Figure 3. Common-mode Range (CMR) of an RS-485 Transceiver

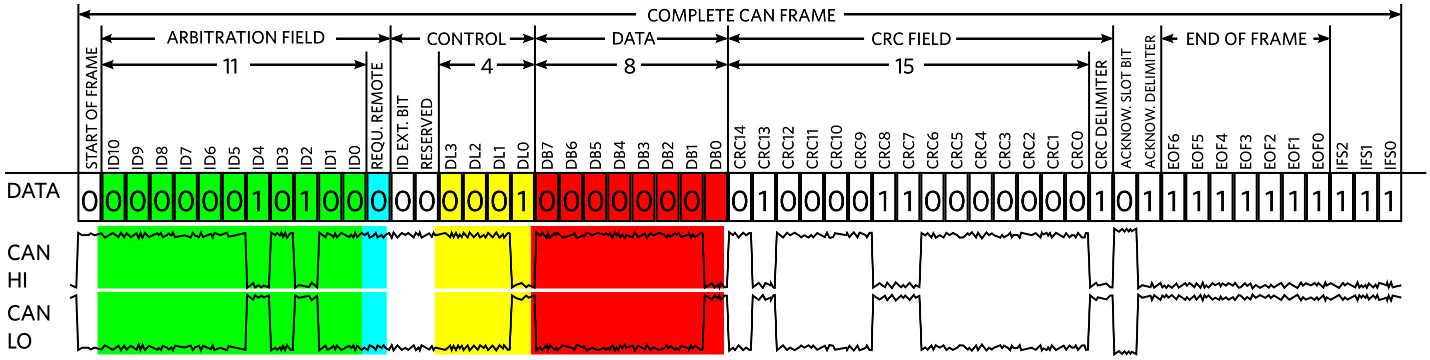

Figure 4. CAN Message Data-Frame Format

Field Name	Bit Length	Description
SOF	1	Start of frame
Identifier (green)	11/29; 12/32	Represents the message priority (11 or 29 bits for standard CAN and extended CAN; 12 or 32 bits for CAN-FD)
RTR (blue)	1	Remote transmission request
IDE	1	Identifier extension bit
r0	1	Reserved bit for future protocol expansion
DLC (yellow)	4/8/9	Code for number of data bytes (4-bit for standard CAN; 8 or 9 bits for CAN-FD)
Data Field (red)	0–64 (0–8 bytes); 0–512 (0–64 bytes)	Data to be transmitted (0–8 bytes for standard CAN; 0–64 bytes for CAN-FD)
CRC	15	Cyclic redundancy check
CRC Delimiter	1	Assigned recessive (1)
ACK slot	1	Dominant bit if error-free message; recessive to discard errant message
ACK Delimiter	1	Acknowledgement delimiter
EOF	7	End of frame

Table 1. CAN Message Data-Frame Format

Arbitration is resolved during transmission of the identifier field, an example of which is shown in Table 2. Even with the new CAN-FD standard, the arbitration phase is limited to 1 Mbps, depending on network topology. But the data-field phase is only limited by the transceiver characteristics, which means it can go much faster.

		Identifier Bits (Arbitration Files)										
	Start Bit	10	9	8	7	6	5	4	3	2	1	0
Node 1	0	0	0	0	0	0	0	0	0	1	1	1
Node 3	0	0	0	0	0	0	0	0	1	Stop Transmitting		

Table 2. Node 3 Loses Arbitration to Node 1 at Bit 3

In addition to arbitration, the data link layer (layer 2 of the OSI model) also contributes to the robustness of the overall CAN system. In this layer, the frame message is repeatedly checked for accuracy and errors. If a message is received with errors, an error frame is sent out. The error frame consists of two different fields: the error flag and the error delimiter. From a message-level perspective, the cyclic redundancy check (CRC) safeguards the information in the frame by adding redundant check bits at the end of transmission, which are then checked on the receiving side. If they do not match, then a CRC error has occurred. The other message check is the frame check, which verifies the structure by checking the bit fields against the fixed format and frame size of SOF, EOF, ACK, and CRC delimiter bits. From a bit-level perspective, there are three checks for errors: acknowledgement, bit monitoring, and bit stuffing. Acknowledgement errors are detected when the transmitter does not read a dominant ACK bit (0). This indicates a transmission error detected by the recipients, which means either the ACK was corrupted or there were no receivers. Bit monitoring checks the bus level for each node for sent and received bits. Bit stuffing is a method that "stuffs" or inserts an extra opposite bit when five of the same bits occur in succession. The opposite bit helps to differentiate error frames and EOF bits. On the receiving side, the extra bit is removed. If the sixth bit is the same as the previous five, then an error is detected by all CAN nodes and error frames are sent out. The original message will need to be retransmitted and pass through arbitration if there is contention on the line.

With CAN features such as arbitration, error-message checking, improved bandwidth, and a larger data field, it is easy to understand the appeal of CANbus in the industrial market. CAN is suitable for applications that require robust communications and reliability in harsh environments. CAN systems are able to prioritize the importance of frame messages and treat critical ones appropriately. Many different systems can be exposed to either electrically noisy sources or a local service personnel that may accidentally short to local supply rails. Maxim CAN transceivers are known for their robust serial interface, with class-leading ESD performance and high level of fault protection.

Notes:

1. http://canbuskits.com/what.php

2. https://teslatap.com/undocumented/model-s-processors-count/

Maxim Integrated
160 Rio Robles
San Jose, CA 95134 USA
408-601-1000

www.maximintegrated.com/CAN