

Controller Area Network (CAN) Implementation Guide

by Dr. Conal Watterson

INTRODUCTION

The controller area network (CAN) is a standard for distributed communications with built-in fault handling, specified for the physical and data link layers of the open systems interconnection (OSI) model in ISO-11898^{1,2}. CAN has been widely adopted in industrial and instrumentation applications and the automotive industry due to the inherent strengths of the communication mechanisms used by CAN.

Features of CAN include

- Allowance for multiple masters on a bus
- Inherent priority levels for messages
- Bus arbitration by message priority
- Error detection and recovery at multiple levels
- Synchronization of data timing across nodes with separate clock sources

At the physical layer, differential data transmission is supported by the CAN protocol, providing advantages such as

- Bidirectional communications across a single pair of twisted cables
- Increased immunity to noise
- Wide common-mode range allowing differences in ground potential between nodes

IMPLEMENTING A CONTROLLER AREA NETWORK

This application note considers the following aspects of how CAN is implemented in industrial applications:

- CAN implementation layers: how the CAN specification and protocols relate to hardware/software and CAN transceiver products
- CAN messages: how the message structure is fundamental to error checking/recovery and arbitration
- Arbitration: how the carrier sense multiple access method specified by CAN allows multiple driving nodes
- Error mechanisms: how the CAN specification inherently enhances communication robustness
- Physical bus: what measures ensure proper communication at the physical layer
- Isolation: signal and power isolation of CAN and integrated isolation solutions for CAN
- Stress protection: mechanisms used in CAN for protecting transceivers from electrical overstress

HOW CAN USES DIFFERENTIAL DATA TRANSMISSION

In traditional differential data transmission (for example, RS-485³), Logic 1 is transmitted as a voltage level high on one noninverting transmission line and low on the inverting line. Correspondingly, Logic 0 is transmitted as low on the noninverting line and high on the inverting line. The receiver uses the difference in voltage between the two lines to determine the Logic 1 or Logic 0 that was transmitted, as shown in Table 1.

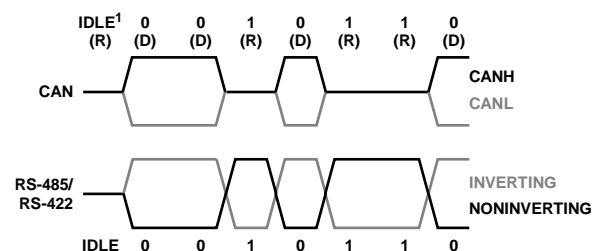
A driver on the bus can also be in a third state, with the driver outputs in a high impedance state. If all nodes are in this condition, the bus is in an idle state. In this condition, both bus lines are usually at a similar voltage with a small differential.

Signaling for CAN differs in that there are only two bus voltage states; recessive (driver outputs are high impedance) and dominant (one bus line, CANH, is high and the other, CANL, is low), with thresholds as shown in Table 1. Transmitting nodes transmit the dominant state for Logic 0 and the recessive state for Logic 1. An idle CAN bus is distinguished from recessive bit transmission simply by detection of multiple recessive bits after an end of frame or error frame.

Table 1. Comparison of CAN and RS-485 Voltage Levels

Logic	RS-485 Levels	CAN State	CAN Levels
1	$A - B \geq +200 \text{ mV}$	Recessive	$\text{CANH} - \text{CANL} \leq 0.5 \text{ V}$
0	$A - B \leq -200 \text{ mV}$	Dominant	$\text{CANH} - \text{CANL} \geq 0.9 \text{ V}$

The two states of dominant and recessive are represented by the CANH and CANL voltage levels shown in Figure 1 that compares CAN signaling to RS-485. This signaling method is fundamental both to the node arbitration and inherent prioritization of messages with lower message IDs (more initial Logic 0s as the message is serially transmitted).



NOTES

1. CAN BUS IDLE AFTER MULTIPLE RECESSIVE BITS.

Figure 1. Comparison of Differential Signaling for CAN and RS-485/RS-422

TABLE OF CONTENTS

Introduction	1	Error Frames	8
Implementing a Controller Area Network.....	1	Error Counters.....	8
How CAN Uses Differential Data Transmission.....	1	Node Error States	9
Revision History	2	Transmission Bit Verification	9
CAN Implementation Layers	3	Bit Stuffing Rules	9
Physical Layer Transceivers.....	3	CRC Check.....	9
CAN Controllers	4	Fixed Form, Bit Field Checks.....	9
DeviceNet Networks	4	Message Acknowledgment.....	9
CANopen Protocol.....	4	Physical Bus.....	10
CAN Message Frame Structure	5	CAN Physical Bus Characteristics	10
Arbitration Field	5	Termination.....	10
Data Length Code (DLC) Field	5	Isolation	11
Cyclic Redundancy Check (CRC) Field	5	Integrated Signal and Power Isolated CAN Transceivers	11
Acknowledgement (ACK) Slot	5	Stress Protection	13
End of Frame.....	5	Miswire and Short-Circuit	13
Arbitration	7	Transient Overvoltage.....	13
Message Priority	7	References.....	14
Error Mechanisms	8	Related Links.....	14

REVISION HISTORY

10/2017—Rev. 0 to Rev. A

Change to DeviceNet Networks Section	4
--	---

2/2012—Revision 0: Initial Version

CAN IMPLEMENTATION LAYERS

Communication using CAN is defined by the International Standard Organization (ISO) as ISO-11898 and can be considered in the context of the seven-layer OSI model for communications. The ISO-11898-1 standard for CAN relates to the data link layer and the effects of this on the surrounding layers. ISO-11898-2 relates to part of the data link layer and the physical layer.

Implementations of CAN depend on the following components:

- Physical layer transceiver to translate the CAN messages to/from differential signals across a physical medium such as a twisted pair cable.
- CAN controller (sometimes embedded in a microprocessor, as with the Analog Devices, Inc. Blackfin® [ADSP-BF548](#)) that implements the data link layer. These adhere to the CAN 2.0b specification⁴ to ensure communication conforms to the ISO 11898 standard.
- CAN software application, implementing the application layer protocol (translating the software application data to/from CAN messages).

The various blocks that implement a CAN application are shown in Figure 2, which shows their relationship to OSI layers and the features implemented by each block.

PHYSICAL LAYER TRANSCEIVERS

CAN transceivers provide the differential physical layer interface between the data link layer, the CAN controller (for example, embedded in some of Analog Devices Blackfin processors), and the physical wiring of the CAN bus. The Analog Devices portfolio includes transceivers with integrated *iCoupler*® digital isolation⁵ for signal isolation and *isoPower*® power isolation⁵, providing fully isolated off-the-shelf CAN PHYs.

The [ADM3051/ADM3052/ADM3053/ADM3054](#) are designed for interfacing to CAN controllers and support various CAN applications. Depending on the application, different high level protocols can be used with CAN, for example, CANopen or DeviceNet™.

- The [ADM3054](#) is a 5 kV rms signal isolated high speed CAN transceiver.
- The [ADM3053](#) is a fully isolated high speed CAN transceiver with 2.5 kV rms signal and power isolation.
- The [ADM3052](#) is a 5 kV rms signal isolated high speed CAN transceiver with an integrated bus voltage regulator to use 24 V bus power (for example, as in DeviceNet applications).
- The [ADM3051](#) is a nonisolated high speed CAN transceiver.

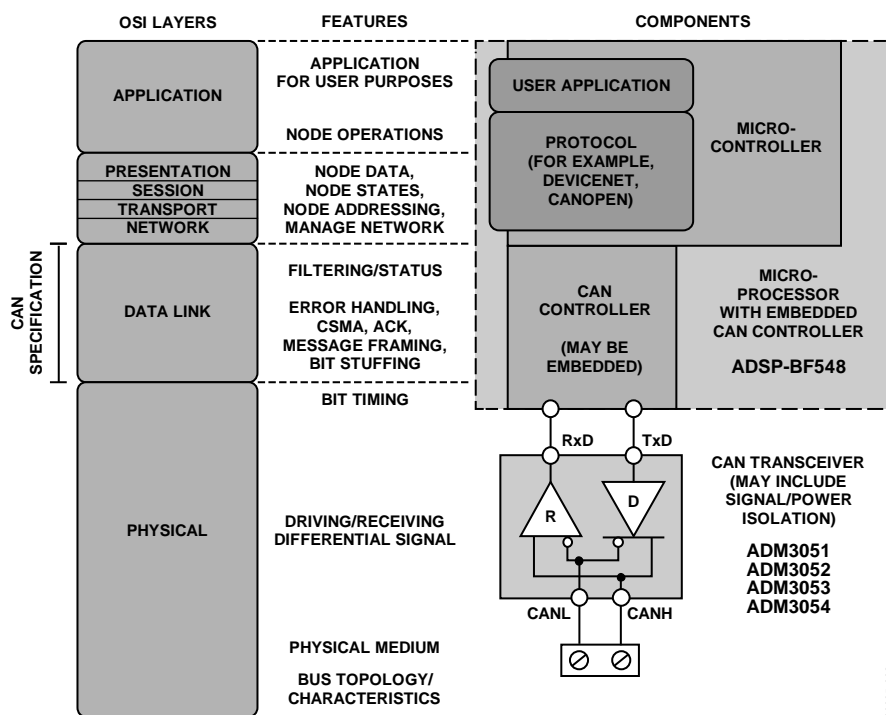


Figure 2. CAN Implementation Blocks as Related to OSI Layers and Features

CAN CONTROLLERS

The data link layer of CAN and physical bit timing is implemented by the CAN controller (sometimes embedded within a microcontroller or digital signal processor (DSP), for example, the [ADSP-BF548](#)), according to the CAN 2.0b specification and conforming to the data link layer portion of the ISO-11898 standard. The CAN controller handles message filtering, arbitration, message framing, error handling, and error detection mechanisms such as bit stuffing.

DeviceNet NETWORKS

DeviceNet⁶ is a specification managed by the Open DeviceNet Vendors Association (ODVA) for communication networks. DeviceNet specifies aspects of the physical layer, the use of CAN for physical and data links layers, and higher level communication using the common industrial protocol (CIP). Industrial and instrumentation commonly use DeviceNet for CAN applications.

DeviceNet specifies a multidrop network that supports master-slave or distributed control schemes across a linear bus topology. The network not only comprises the differential signaling bus lines, but also power and ground, so that nodes can be powered from the bus.

The physical layer specifications for DeviceNet specify features such as the use of CAN technology, protection from wiring errors and the ability to add or remove nodes from the network while it is operational. Various aspects of the physical layer are specified in detail, including the transmission media and connectors.

The [ADM3052](#) isolated CAN transceiver meets the physical layer requirements of DeviceNet, in addition to incorporating features used by DeviceNet nodes, such as signal isolation, miswire protection, and a linear regulator to power the bus side of the device from the 24 V bus power (V_+). Figure 3 provides a functional block diagram of the [ADM3052](#), as well as an application configuration.

CANopen PROTOCOL

CANopen⁷ is an application layer protocol maintained by CAN in Automation (CiA) that uses the CAN data link and physical layers and specifies standardized profiles for devices, communication, and applications. This allows interoperability across different application areas, for example, industrial automation, building control, and generic input/output (I/O). The physical layer conforms to ISO-11898, with a bus topology and data rates up to 1 Mb.

The [ADM3053](#) can be used to fully isolate communication on networks such as those using CANopen, with power for the bus side of the transceiver provided by an integrated isolated dc-to-dc converter. A functional block diagram of the [ADM3053](#) is shown in Figure 12.

Alternatively, where an isolated 5 V supply is already present in the application circuit, the [ADM3054](#) provides an integrated CAN transceiver with 5 kV rms digital isolator.

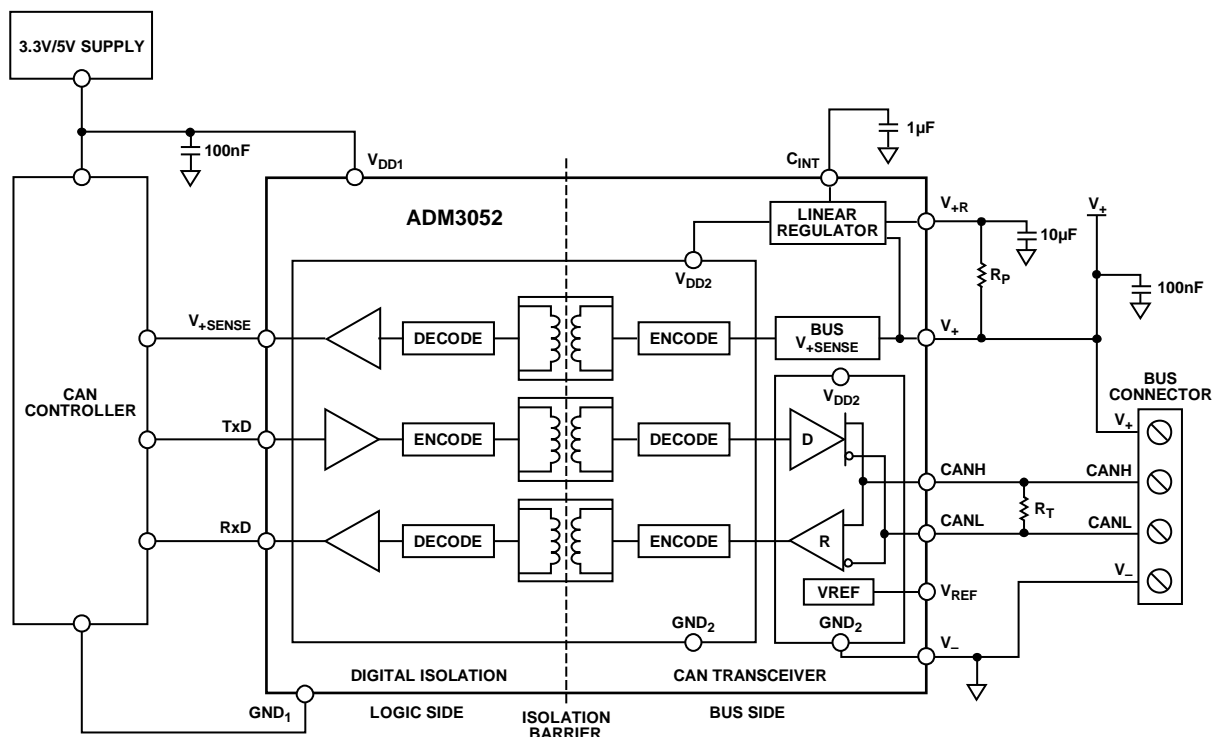


Figure 3. Application Example Using the [ADM3052](#) Isolated CAN Transceiver

10035-003

CAN MESSAGE FRAME STRUCTURE

The structure of a CAN message is fundamental to the schemes used for achieving robust communications through error detection, as well as the inherent prioritization of messages and multiple driver capability based on bit-wise arbitration. The CAN controller handles the framing of CAN messages, as specified in CAN 2.0b (for the inclusion of the extended message type detailed here).

As discussed in the Introduction section, signaling for CAN messages at the physical layer comprises leaving the bus in a high impedance recessive state for a transmission of 1 and transmitting a differential high/low dominant state for a transmission of 0. In Figure 4 and Figure 5, which show the composition of CAN message frames, white bit fields denote a recessive bus state, dark grey fields indicate a dominant state, and grey indicates bit fields that can be dominant or recessive.

ARBITRATION FIELD

As shown in Figure 4 and Figure 5, message identifiers form part of the CAN message frame. In conjunction with a number of flag bits, this section of the CAN message frame is termed the arbitration field (indicated in Figure 4 and Figure 5), with the message ID and flags denoting the message type, dictating arbitration and, as a result, message priority. There are two main attributes that can distinguish CAN message types using the flag bits:

- An IDE bit, denoting use of extended identifier
- An RTR bit, denoting a remote transmission request

Remote transmission requests are used to request that a certain message be sent by its originator (normally nodes on a bus send data autonomously). This remote transmission request is a message without any data, sent by one node, to request that another node (or nodes) transmit the message of the same ID but with data.

The SRR bit is a substitute remote request in messages with extended IDs for compatibility with nodes that only use standard CAN messages.

Some reserved bits are also present for future expansion of the CAN protocol and must be read as zero to ensure future compatibility.

DATA LENGTH CODE (DLC) FIELD

Up to eight bytes of data can be present in a CAN message frame. The length of data in the frame is encoded into the DLC field. A remote transmission request has the data length of the requested data frame. Note that only values of zero to eight are valid for the DLC field.

CYCLIC REDUNDANCY CHECK (CRC) FIELD

The CRC field is part of the error checking mechanism in the CAN protocol. The CRC field contains a sequence generated from the content of the CAN frame, which is the remainder from a polynomial division operation. The received sequence and data can be used in another polynomial division operation to check that no bits have accidentally been received in an inverted/error state.

The CRC field is delimited (bus recessive for one bit interval) from the ACK slot that follows to ensure that the CRC field is not affected by the ACK behavior.

ACKNOWLEDGEMENT (ACK) SLOT

Although the ACK slot is transmitted as Logic 1 (bus recessive) by the message originator, error-free transmission is achieved only when during this bit interval the bus is set to a dominant state by another node. Any other node that receives the previous message fields error-free will automatically attempt to transmit Logic 0 to acknowledge receipt of the message.

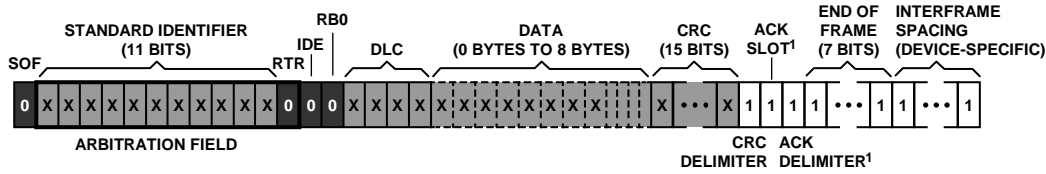
An ACK delimiter (bus recessive for one bit interval) is present to allow a space before the remainder of the message in case the node transmitting the ACK bit has slightly different timing and the dominant bit overruns into the next bit period (that is, the delimiter interval).

If the bus remains in a recessive state during the ACK slot, a node detecting this will begin transmitting an error frame in the next bit period.

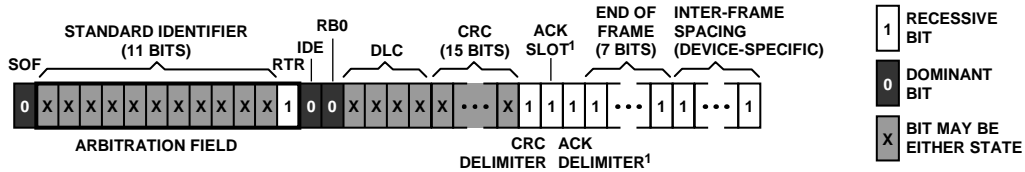
END OF FRAME

The end of frame is communicated by a 7-bit interval bus recessive state, a sequence not occurring during the rest of the message due to bit stuffing (run-length limited coding where a complementary bit is inserted after every six consecutive 1s or 0s in the message). After the end of a frame, a bus recessive interframe space is also observed (the exact length of this varies depending on the length of time specified for bus idle for a given CAN controller).

STANDARD DATA FRAME



STANDARD REMOTE FRAME



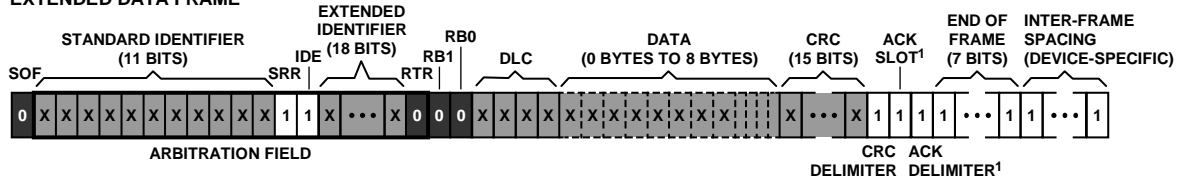
NOTES

1. ORIGINATOR OF FRAME TRANSMITS RECESSIVE (1) DURING ACK SLOT/DELIMITER. SUCCESSFUL TRANSMISSION OF MESSAGE FRAME REQUIRES AT LEAST ONE OTHER NODE TO TRANSMIT A DOMINANT (0) BIT DURING THE ACK SLOT.

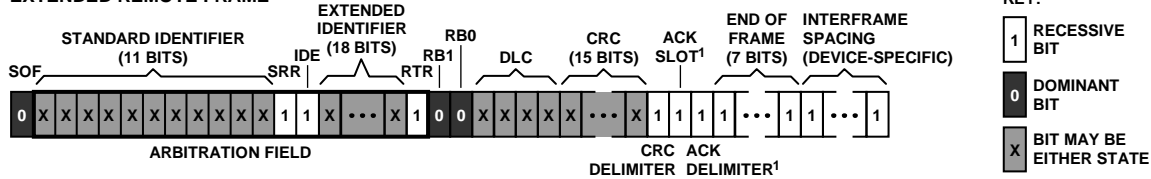
Figure 4. CAN Standard Message Frame Fields

10035-004

EXTENDED DATA FRAME



EXTENDED REMOTE FRAME



NOTES

1. ORIGINATOR OF FRAME TRANSMITS RECESSIVE (1) DURING ACK SLOT/DELIMITER. SUCCESSFUL TRANSMISSION OF MESSAGE FRAME REQUIRES AT LEAST ONE OTHER NODE TO TRANSMIT A DOMINANT (0) BIT DURING THE ACK SLOT.

Figure 5. CAN Extended Message Frame Fields

10035-005

ARBITRATION

On a single CAN bus, any node can transmit data. Nodes arbitrate for use of the bus, so that, in the event of two or more nodes attempting transmission, messages are transmitted one after another according to their priority.

Nondestructive and transparent arbitration is possible because transmission of a dominant bit overwrites the recessive bus state. The CAN controller of each node monitors the bus as it transmits and, consequently, can detect if another node wins arbitration. Nondestructive and transparent means that messages are not corrupted, and transmission of the highest priority message is uninterrupted by arbitration.

If the bus is active (a node is transmitting or has just finished transmission), no other nodes will attempt transmission. If, when the bus is idle (for at least the length of the interframe spacing), and more than one node begins transmission, arbitration occurs transparently and nondestructively. Nondestructive arbitration means that the node winning arbitration can simply continue transmission of its message without any other node having interfered with the message transmission.

Using Figure 6 as an example, both nodes have attempted transmission at the same time. Both CAN nodes monitor the bus state as they transmit, and as Node 2 attempts to transmit SID7, it can detect that another node (in this case Node 1) has written a dominant value to the bus and has won arbitration accordingly. Node 2 does not attempt retransmission until after the bus has been idle for the interframe spacing. Node 1 continues transmission but also monitors the bus to detect errors or detect if another node wins arbitration in a later arbitration field bit interval.

Note the example of bit stuffing after the initial five dominant bits transmitted by Node 1. These extra bits are important in allowing nodes to synchronize their timing and are also used for error detection.

MESSAGE PRIORITY

CAN message frames are transmitted most significant bit (MSB) first, and as the message IDs are at the beginning of a frame, they form part of the arbitration sequence. Messages with a lower ID (more initial 0s) have a higher priority. In addition, remote frames (RTR bit = 1) have a lower priority than the data frame with the same ID.

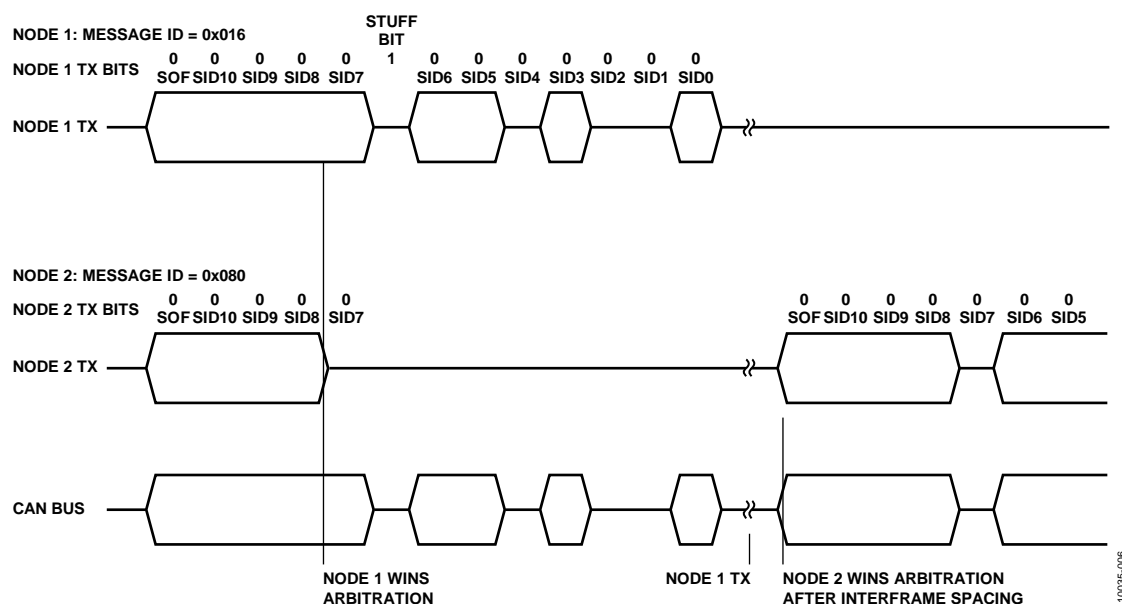


Figure 6. Example of Arbitration Between CAN Nodes

ERROR MECHANISMS

CAN incorporates various mechanisms for supporting error checking and handling. These include definitions of the following error detection schemes:

- Transmission bit verification
- Bit stuffing rules
- CRC check
- Fixed-form bit field checks
- Mandatory message ACK

These errors are handled using the following mechanisms:

- Error frames
- Error counters
- Node error states

The CAN controller detects and handles these errors and supports the error detection by framing CAN messages according to CAN 2.0b.

ERROR FRAMES

An error frame is distinguished by having six consecutive bits. This sequence is dominant or recessive depending on the state of the node transmitting the error. This sequence violates the normal transmission rules and so is detectable by other nodes.

Any node transmits error frames immediately when it detects an error. As an error frame itself highlights an error, other nodes concurrently transmit their own error frames, resulting in a superposition of multiple error frames.

The sequence of six consecutive bits is the error flag. An error frame also comprises an error delimiter to allow for the error flags from other nodes overrunning the initial six bit periods.

An example of transmission of a CAN frame with an error and a subsequent CAN active error frame is shown in Figure 7, compared to an error-free transmission.

Any node transmitting an error flag subsequently sends recessive bits until the bus is detected as being in the recessive state, after which, an additional seven recessive bits are transmitted. The node can then attempt transmission of regular CAN frames.

ERROR COUNTERS

Every CAN node must implement two error counters; a transmit error counter and a receive error counter. These are incremented on the basis of transmit or receive errors and decremented on the basis of successful transmission or receiving of messages.

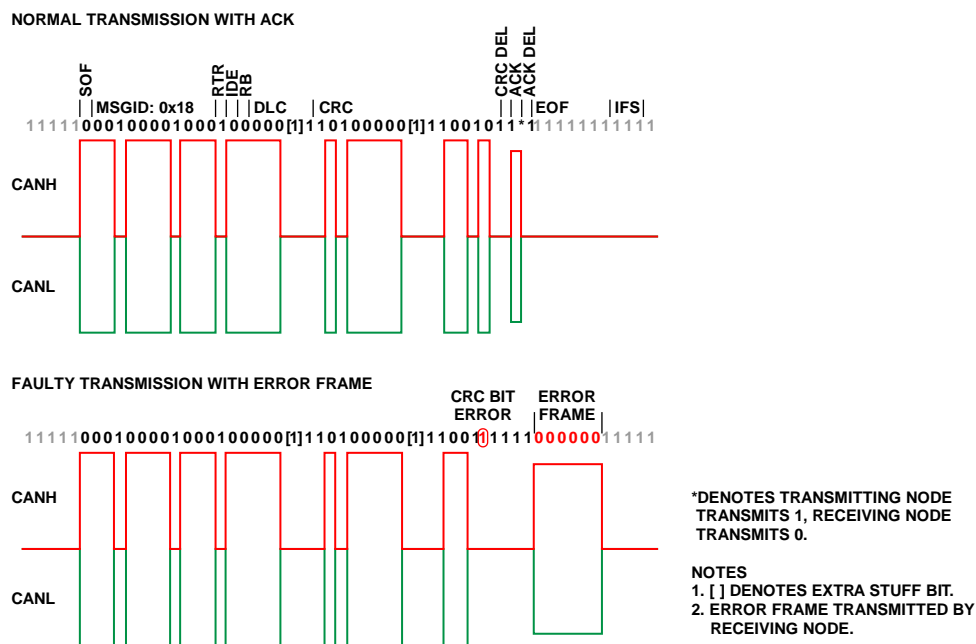


Figure 7. Normal Transmission with ACK and Faulty Transmission with CAN Error Frame

NODE ERROR STATES

Based on the error counters, a node may be in one of three states:

- Error active
- Error passive
- Bus off

In the error active state, the node is expected to be able to communicate on the bus and send active error flags when errors are detected. The transmit and receive error counters must be less than 127 for the node to remain error active.

In the error passive state, the node may communicate on the bus but only send passive error flags in the event of errors. The node enters this state when the transmit or receive error counters reach or surpass 127. The node becomes error active once these counters again decrease to 127 or below.

In the bus off state, the node is not permitted to communicate on the bus. This state occurs once the transmit error count reaches or passes 256. After 11 consecutive recessive bits have been detected 128 times, the node may have its error counters reset to 0 and can enter the error active state.

TRANSMISSION BIT VERIFICATION

A transmitting node monitors the bus as it transmits to verify that each bit appears on the bus as intended. The detection of an opposite bit level on the bus is termed a bit error, a form of transmit error. There are two exceptions where an opposite bit level may be detected without error:

- The arbitration field is transmitted as recessive; however, it is expected that the bus state will change to dominant as other nodes acknowledge the message.
- It is not a bit error if a node sending a passive error flag detects the bus in a dominant state.

BIT STUFFING RULES

The process of bit stuffing refers to the scheme whereby a transmitting node inserts a complementary bit in the transmitted bit stream after five consecutive message bits of the same value. This encoding is used for an entire CAN data or remote frame with the exception of fixed form bit fields for the CRC delimiter, ACK field, and end of frame. CAN error or overload frames are also fixed form.

Any node receiving a message that breaks the bit stuffing rules (more than five consecutive recessive or dominant bits in a sequence that should be bit stuffed) will detect this as a receive error and take action dependant on the node state (for example, transmit an active or passive error frame).

CRC CHECK

The CRC is a sequence of bits that are calculated based on the first part of the CAN frame (up to the data field). The transmitting node computes this sequence by dividing a generator polynomial into a polynomial formed from the CAN data, to provide a remainder that serves as the CRC sequence. This CRC sequence is inserted into the CAN message in the CRC field.

The receiving node divides the generator polynomial into a polynomial formed from the data and CRC sequences together. In the event of no errors, the remainder should be zero.

This procedure is termed a CRC check. If the received message is in error (that is, with inverted bits), the CRC check fails.

FIXED FORM, BIT FIELD CHECKS

Certain bit fields in a CAN message frame are of fixed form, specifically the CRC delimiter field, ACK field, and end of frame. Receiving nodes perform a form check to ensure these are correct.

MESSAGE ACKNOWLEDGMENT

Although the arbitration field is transmitted as recessive, the transmitting node checks for a dominant bit on the bus during this bit time. This dominant bit is an ACK bit sent by any node receiving the message correctly. A transmit error occurs if no other node acknowledges the message. Insertion of the ACK bit into a CAN message by a receiving node is shown in Figure 7, depicted with a lower differential such as would be observed when the receiving node is at the other end of the bus from the probe point at the transmitting node.

PHYSICAL BUS

CAN PHYSICAL BUS CHARACTERISTICS

The physical layer characteristics for CAN are specified in ISO-11898-2. This standard specifies the use of cable comprising parallel wires with an impedance of nominally $120\ \Omega$ ($95\ \Omega$ minimum and $140\ \Omega$ maximum). The use of shielded twisted pair cables is generally necessary for electro-magnetic compatibility (EMC) reasons, although ISO-11898-2 also allows for unshielded cable.

A maximum line length of 40 meters is specified for CAN at a data rate of 1 Mb. However, at lower data rates, potentially much longer lines are possible. ISO-11898-2 specifies a line topology, with individual nodes connected using short stubs. Minimize this stub length (≤ 0.3 meters at data rates of 1 Mb).

TERMINATION

In a transmission line, there are two current paths, one to carry the currents from the driver to the receiver and another to provide the return path back to the driver. CAN links are a little more complicated because of the fact that they have two signal wires that share a termination as well as a ground return path. However, the basic principles of transmission lines are the same. For reliable CAN communications, it is essential that the reflections in the transmission line be kept as small as possible. This can only be done by proper cable termination. Figure 8 demonstrates two termination schemes for CAN applications.

Reflections happen very quickly during and just after signal transitions. On a long line, the reflections are more likely to continue long enough to cause the receiver to misread logic levels. On short lines, the reflections occur much sooner and have no effect on the received logic levels.

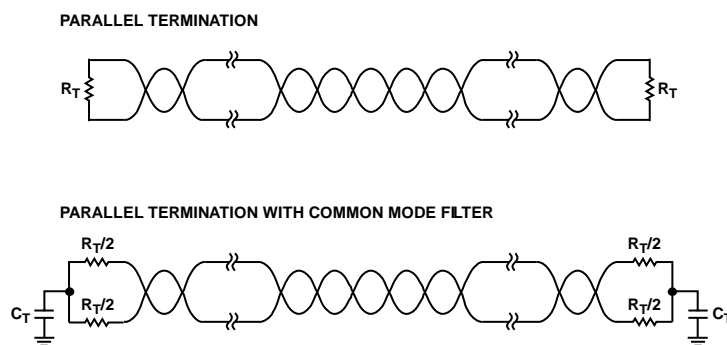
Parallel Termination

In CAN applications, both ends of the bus must be terminated because any node on the bus may transmit data. Each end of the link has a termination resistor equal to the characteristic impedance of the cable, although the recommended value for the termination resistors is nominally $120\ \Omega$ ($100\ \Omega$ minimum and $130\ \Omega$ maximum). There should be no more than two terminating resistors in the network, regardless of how many nodes are connected, because additional terminations place extra load on the drivers.

ISO-11898-2 recommends not integrating a terminating resistor into a node but rather attaching standalone termination resistors at the furthest ends of the bus. This is to avoid a loss of a termination resistor if a node containing that resistor is disconnected. The concept also applies to avoiding the connection of more than two termination resistors to the bus, or locating termination resistors at other points in the bus rather than at the two ends.

Parallel Termination with Common-Mode Filtering

To further enhance signal quality, split the terminating resistors at each end in two and place a filter capacitor, C_T , between the two resistors. This filters unwanted high frequency noise from the bus lines and reduces common-mode emissions.



NOTES

1. R_T IS EQUAL TO THE CHARACTERISTIC IMPEDANCE OF THE CABLE.

Figure 8. Termination Schemes for CAN Applications

10035-008

ISOLATION

In CAN applications, there are often long links that can cause the ground potential at different nodes on the bus to be slightly different. This causes ground currents to flow through the path of least resistance through either the common earth ground or the ground wire. If the same electrical system is used to connect the power supplies of all nodes to the same earth ground, the ground connection may have reduced noise. Note, however, that motors, switches, and other electrically noisy equipment can still induce ground noise into the system.

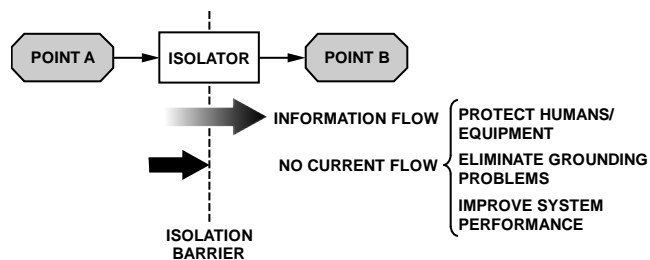


Figure 9. Galvanic Isolation Allows Information Flow While Preventing Ground Current Flow

Different power systems are required in some applications. This is likely to increase the impedance of the earth ground, and the ground currents from other sources are more likely to find their way into the ground wire of the link. Isolating the link reduces or even eliminates these problems. Galvanic isolation is the recommended solution if there is no guarantee that the potential at the earth grounds at different nodes in the system are within the common-mode range of the transceiver. Galvanic isolation allows information flow but prevents current flow (see Figure 9).

INTEGRATED SIGNAL AND POWER ISOLATED CAN TRANSCEIVERS

Analog Devices offers CAN transceivers for various applications, including

- [ADM3054](#) with integrated 5 kV rms signal isolation for systems with an existing isolated 5 V supply
- [ADM3053](#) with integrated signal and power isolation for complete isolation of CAN applications
- [ADM3052](#) with integrated signal isolation and voltage regulator for isolated applications using 24 V bus power (for example, DeviceNet)
- [ADM3051](#) for nonisolated applications or applications using traditional isolation methods

When galvanic isolation is implemented, some method is required of providing power to the bus side of the application circuit. In DeviceNet applications, 24 V is supplied to all nodes on the CAN bus, as demonstrated with the [ADM3052](#). The high level of integration that can be achieved with the [ADM3052](#) is shown in Figure 10. This part requires a minimum of external components and internally incorporates signal isolation via Analog Devices *iCoupler* technology, in addition to integrating a regulator to supply 5 V to the bus side of the device from 24 V bus power. An isolation rating of 5 kV rms is achieved with the [ADM3052](#).

For applications where an isolated supply is discretely provided, or bus power is used for various circuit elements, the [ADM3054](#) can be used with a 5 V isolated power supply on the bus side of the device.

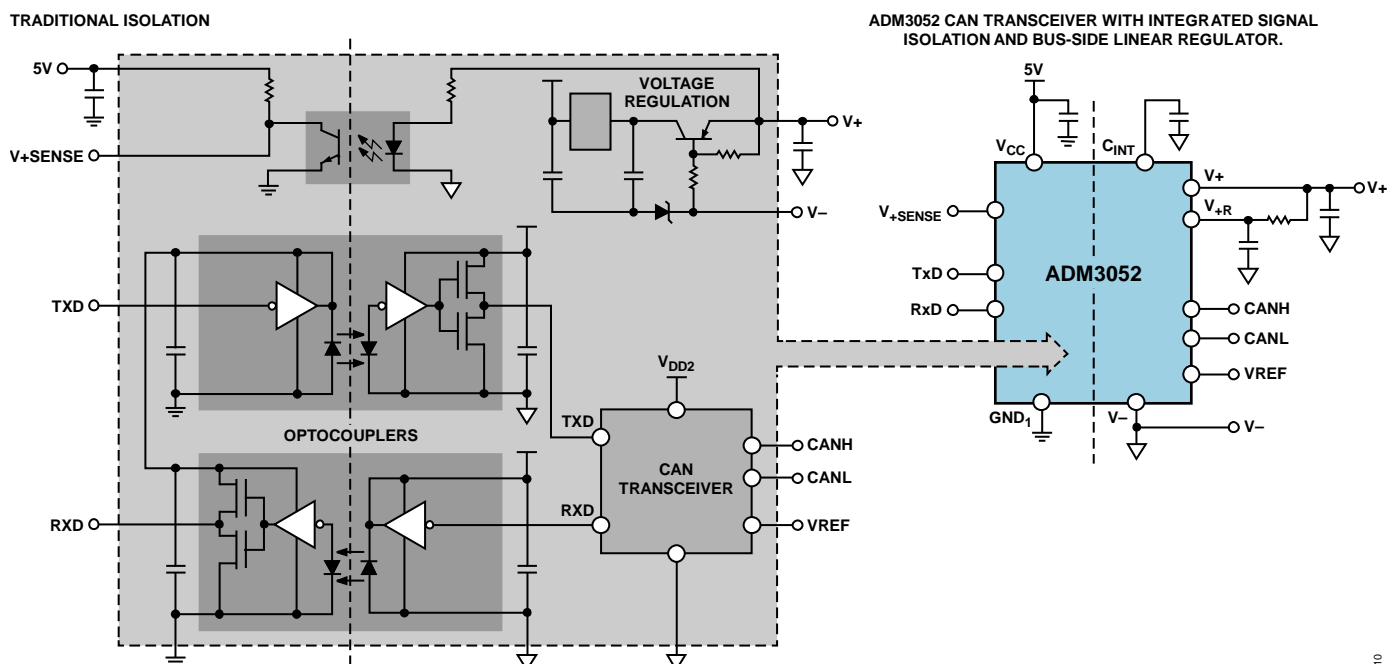


Figure 10. Comparison of Traditional Isolation with Bus Power vs. [ADM3052](#)

In other applications, power from a supply on the logic side of the circuit can be transferred across the isolation barrier. Traditionally, this is achieved by use of a dc-to-dc converter implemented with an oscillator, transformer, and regulator using discrete components.

The [ADM3053](#) integrates power isolation in addition to signal isolation. A single 5 V supply can supply power to the bus side

via an integrated isolated dc-to-dc converter using Analog Devices *isoPower* technology. An isolation rating of 2.5 kV rms is achieved with the [ADM3053](#). Figure 11 compares the traditional solution using discrete devices to the [ADM3053](#). The internal blocks of the [ADM3053](#) are shown in the functional block diagram in Figure 12.

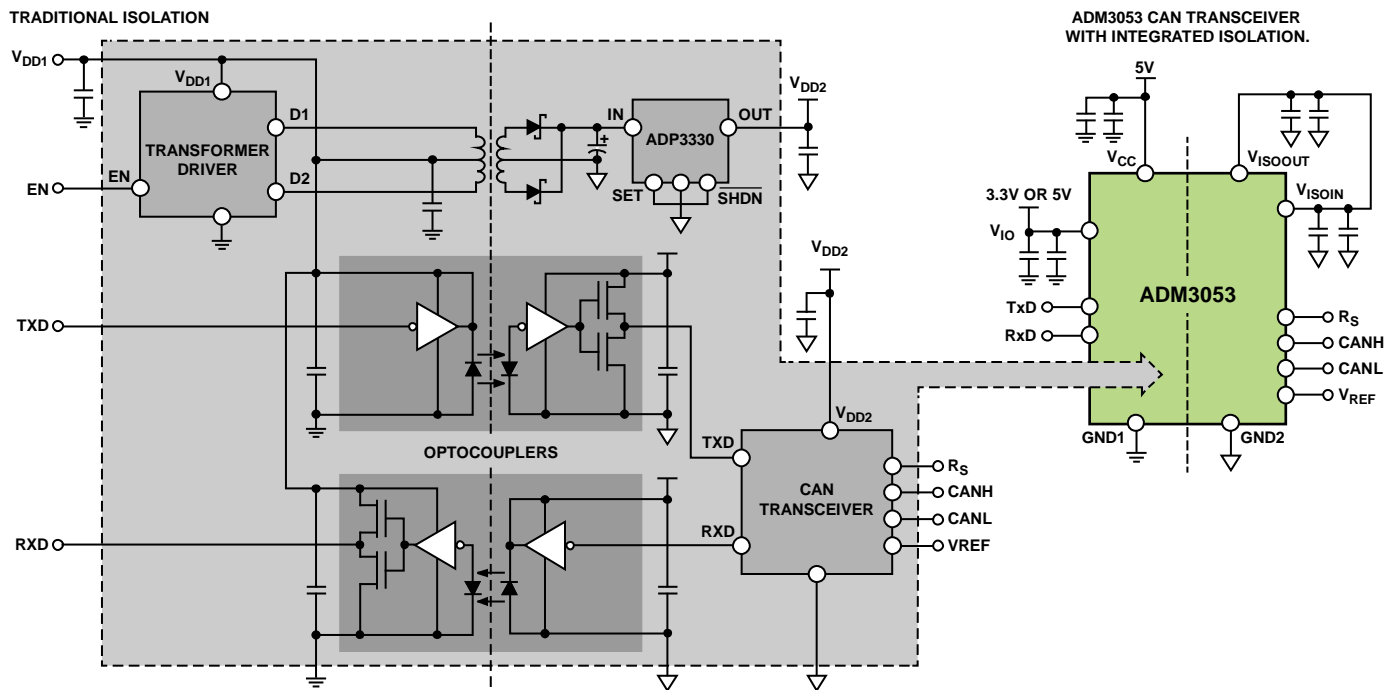


Figure 11. Comparison of Traditional Isolation (Signal and Power) vs. [ADM3053](#)

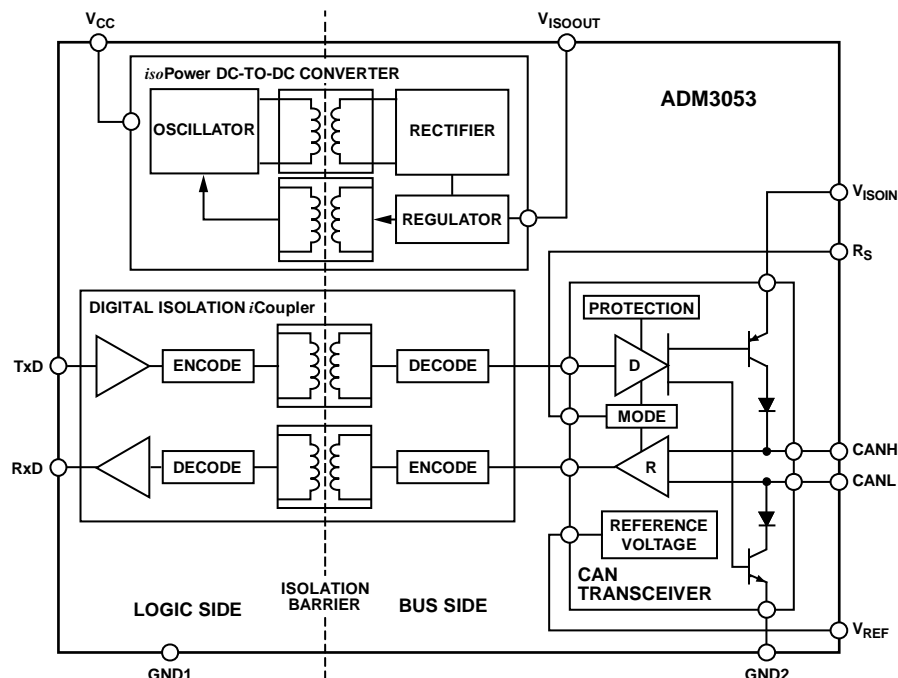


Figure 12. [ADM3053](#) Functional Block Diagram

STRESS PROTECTION

MISWIRE AND SHORT-CIRCUIT

Some CAN applications, such as DeviceNet, in addition to carrying data on the bus lines, CANH and CANL, also distribute power along the bus. In such systems, bus power of typically 24 V is routed along the bus, along with a common ground.

In such systems, the connector for each CAN node has four wires, CANH, CANL, 24 V, and ground. To prevent damage in the case of miswiring of these signals, CAN nodes using 24 V bus power require protection on all bus lines (CANH, CANL, power, and ground). Miswire protection is a requirement for the DeviceNet protocol. The [ADM3052](#) isolated CAN transceiver with integrated bus power regulator incorporates ± 36 V miswire protection on the CANH, CANL, V+, and V- pins.

Other CAN nodes that do not use power from the bus may still require this protection. In such cases, the CAN node still requires protection against shorting of CANH or CANL by connection to a power or ground line. For this reason, the [ADM3051](#), [ADM3053](#), and [ADM3054](#) CAN transceivers also incorporate ± 36 V protection on CANH and CANL.

TRANSIENT OVERVOLTAGE

In I&I applications, lightning strikes, power source fluctuations, inductive switching, and electrostatic discharge can cause damage to CAN transceivers by generating large transient voltages. The following electrostatic discharge (ESD) protection, electrical fast transients (EFT) protection, and surge protection specifications are relevant to industrial applications:

- IEC 61000-4-2 ESD protection
- IEC 61000-4-4 EFT protection
- IEC 61000-4-5 surge protection

The [ADM3051](#), [ADM3052](#), and [ADM3053](#) CAN transceivers offered by Analog Devices include basic ESD protection on all pins. The level of protection can be further enhanced on the bus pins when using external clamping devices, such as transient voltage suppression (TVS) diodes. TVS diodes are normally used to protect silicon devices, like CAN transceivers, from transients.

The protection is accomplished by clamping the voltage spike to a limit, by the low impedance avalanche breakdown of a PN junction. TVS diodes are ideally open-circuit devices. A TVS diode can be modeled as a large resistance in parallel with some capacitance while working below its breakdown voltage. When a transient is generated, and the surge voltage is larger than the breakdown voltage of the TVS, the resistance of the TVS decreases to keep the clamping voltage constant. The selection of TVS diode is such that the clamping voltage is less than the voltage rating of the device that it is protecting. The transients are clamped instantaneously (<1 ns), and the damaging current is diverted away from the protected device (see Figure 13).

The function of a TVS in CAN applications is to protect the CAN transceiver by clamping voltages on the bus that are outside the maximum ratings. Some TVS devices have been specifically designed for CAN applications. An example circuit using bidirectional Zener diodes to implement TVS on a CAN bus is shown in Figure 14.

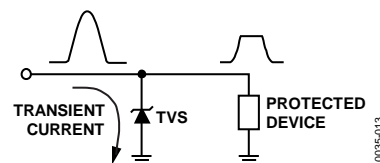


Figure 13. Transient Voltage Suppressor

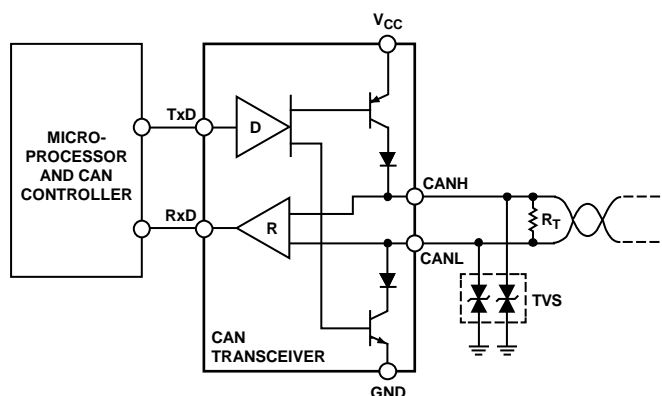


Figure 14. TVS Application Circuit

REFERENCES

- ¹ ISO 11898-1:2003, “Road Vehicles — Controller Area Network (CAN) — Part 1: Data Link Layer and Physical Signalling,” (ISO International Standard, 2003).
- ² ISO 11898-2:2003, “Road Vehicles — Controller Area Network (CAN) — Part 2: High Speed Medium Access Unit,” (ISO International Standard, 2003).
- ³ Hein Marais, [Application Note AN-960](#), “S-485/RS-422 Circuit Implementation Guide,” (Analog Devices, Inc., 2008).
- ⁴ CAN Specification 2.0, Part B, (CAN in Automation, 1991).
- ⁵ Boaxing Chen, “[iCoupler® Products with isoPower™ Technology: Signal and Power Transfer Across Isolation Barrier Using Micro-Transformers](#),” Technical Article, (Analog Devices, Inc., 2006).
- ⁶ DeviceNet™ Technical Overview, (Open DeviceNet™ Vendor Association, Inc., 2001), X to XI.
- ⁷ EN 50325-4:2002, “Industrial Communication Subsystem Based on ISO 11898 (CAN) for Controller-Device Interfaces, CANopen,” (CAN in Automation, 2002).

RELATED LINKS

Resource	Description
ADM3051	Product Page, High Speed Industrial CAN Transceiver with Bus Protection for 24 V Systems
ADM3052	Product Page, Isolated CAN Transceiver with Integrated High Voltage, Bus-Side, Linear Regulator
ADM3053	Product Page, Signal and Power isolated CAN Transceiver with Integrated Isolated DC-to-DC Converter
ADM3054	Product Page, 5 kV rms Signal Isolated High Speed CAN Transceiver with Bus Protection
ADSP-BF548	Product Page, High Performance Convergent Multimedia Blackfin Processor
AN-960	RS-485/RS-422 Circuit Implementation Guide