TEXAS INSTRUMENTS

Application Report
SLOA101 - August 2002

Introduction to the Controller Area Network (CAN)

Steve Corrigan *HPL*

ABSTRACT

A controller area network (CAN) is ideally suited to the many high-level industrial protocols embracing CAN and ISO 11898 as their physical layer. Its cost, performance, and upgradeability provide for tremendous flexibility in system design. This paper presents a brief introduction to the CAN operating principles, the implementation of a basic CAN bus using Texas Instrument's CAN transceivers and DSPs, and a discussion of the robust error detection and fault confinement mechanisms. Some of the properties of CAN, especially relating to the electrical layer and features of transceiver products, are then discussed at a tutorial level.

Contents

List of Figures

List of Tables

1 Introduction

The CAN bus was developed by BOSCH[1] as a multi-master, message broadcast system that specifies a maximum signaling rate of 1M bit per second (bps). Unlike a traditional network such as USB or Ethernet, CAN does not send large blocks of data point-to-point from node A to node B under the supervision of a central bus master. In a CAN network many short messages like temperature or RPM are broadcast to the entire network, which allows for data consistency in every node of the system.

This application report explains the CAN message format, message identifiers, and bit-wise arbitration—a major benefit of the CAN signaling scheme. CAN bus implementation is examined and typical waveforms are presented.

2 The CAN Standard

CAN is an International Standardization Organization (ISO) defined serial communications bus originally developed for the automotive industry to replace the complex wiring harness with a two-wire bus. The specification calls for signaling rates up to 1 Mbps, high immunity to electrical interference, and an ability to self-diagnose and repair data errors. These features have led to CAN's popularity in a variety of industries including automotive, marine, medical, manufacturing, and aerospace.

The CAN communications protocol, ISO 11898, describes how information is passed between devices on a network, and conforms to the Open Systems Interconnection (OSI) model that is defined in terms of layers. Actual communication between devices connected by the physical medium is defined by the physical layer of the model. The ISO 11898 architecture defines the lowest two layers of the seven layer OSI/ISO model as the data-link layer and physical layer in Figure 1.

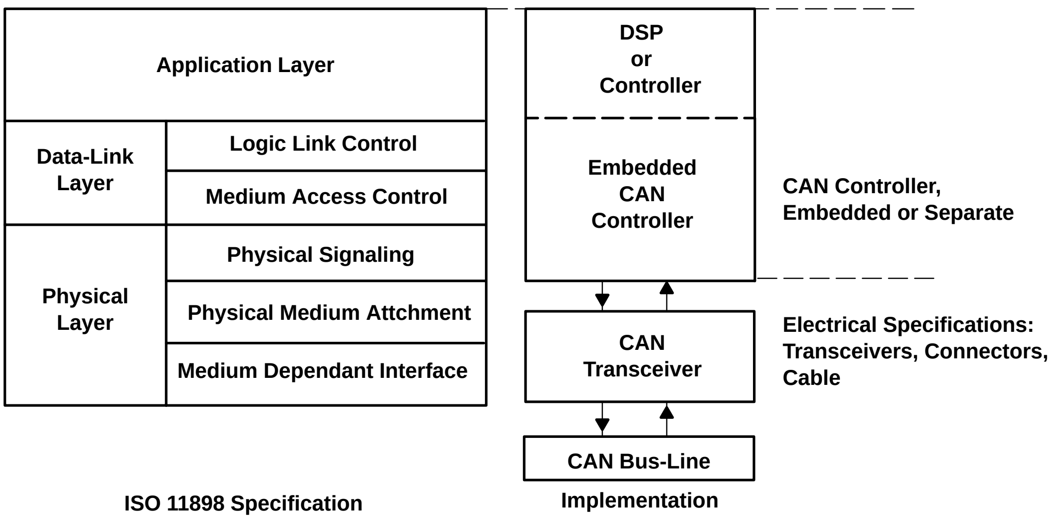

Figure 1. The Layered ISO 11898 Standard Architecture

[1] Robert Bosch GmbH, www.bosch.com

In Figure 1, the application layer establishes the communication link to an upper-level application specific protocol such as the vendor independent CANopen protocol. This protocol is supported by the international users and manufacturers group, CAN in Automation (CiA). Additional CAN information is located at the CiA website, can-cia.de. There are many similar emerging protocols dedicated to particular applications like industrial automation or aviation. Examples of industry-standard CAN-based protocols are KVASER's CAN Kingdom, Allen-Bradley's DeviceNet™ and Honeywell's Smart Distributed System (SDS™).

3 Standard CAN or Extended CAN

The CAN communication protocol is a carrier-sense multiple-access protocol with collision detection and arbitration on message priority (CSMA/CD+AMP). CSMA means that each node on a bus must wait for a prescribed period of inactivity before attempting to send a message. CD+AMP means that collisions are resolved through a bit-wise arbitration, based upon a preprogrammed priority of each message in the identifier field of a message. The higher priority identifier always wins bus access.

The first version of the CAN standards listed in Table 1, ISO 11519 (Low-Speed CAN) is for applications up to 125 kbps with a *standard* 11-bit identifier. The second version, ISO 11898 (1993), also with 11-bit identifiers provides for signaling rates from 125 kbps to 1 Mbps while the more recent ISO 11898 amendment (1995) introduces the *extended* 29-bit identifier. The ISO 11898 11-bit version is often referred to as Standard CAN Version 2.0A, while the ISO 11898 amendment is referred to as Extended CAN Version 2.0B. The Standard CAN 11-bit identifier field in Figure 2 provides for 2^{11}, or 2048 different message identifiers, while the Extended CAN 29-bit identifier in Figure 3 provides for 2^{29}, or 537 million identifiers.

Table 1. CAN Versions

NOMENCLATURE	STANDARD	MAX. SIGNALING RATE	IDENTIFIER
Low–Speed CAN	ISO 11519	125 kbps	11-bit
CAN 2.0A	ISO 11898:1993	1 Mbps	11-bit
CAN 2.0B	ISO 11898:1995	1 Mbps	29-bit

3.1 The Bit Fields of Standard CAN and Extended CAN

3.1.1 Standard CAN

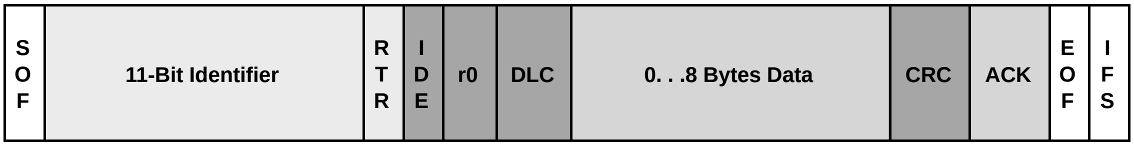

Figure 2. Standard CAN: 11-Bit Identifier

The meaning of the bit fields of Figure 2 are:

- SOF—The single dominant start of frame (SOF) bit marks the start of a message, and is used to synchronize the nodes on a bus after being idle.
- Identifier—The Standard CAN 11-bit identifier establishes the priority of the message. The lower the binary value, the higher its priority.

- RTR—The single remote transmission request (RTR) bit is dominant when information is required from another node. All nodes receive the request, but the identifier determines the specified node. The responding data is also received by all nodes and used by any node interested. In this way all data being used in a system is uniform.
- IDE—A dominant single identifier extension (IDE) bit means that a standard CAN identifier with no extension is being transmitted.
- r0—Reserved bit (for possible use by future standard amendment).
- DLC—The 4-bit data length code (DLC) contains the number of bytes of data being transmitted.
- Data—Up to 64 bits of application data may be transmitted.
- CRC—The 16-bit (15 bits plus delimiter) cyclic redundancy check (CRC) contains the checksum (number of bits transmitted) of the preceding application data for error detection.
- ACK—Every node receiving an accurate message overwrites this recessive bit in the original message with a dominate bit, indicating an error-free message has been sent. Should a receiving node detect an error and leave this bit recessive, it discards the message and the sending node repeats the message after rearbitration. In this way each node acknowledges (ACK) the integrity of its data. ACK is 2 bits, one is the acknowledgement bit and the second is a delimiter.
- EOF—This end-of-frame (EOF) 7-bit field marks the end of a CAN frame (message) and disables bit–stuffing, indicating a stuffing error when dominant. When 5 bits of the same logic level occur in succession during normal operation, a bit of the opposite logic level is *stuffed* into the data.
- IFS—This 7-bit inter-frame space (IFS) contains the amount of time required by the controller to move a correctly received frame to its proper position in a message buffer area.

3.1.2 Extended CAN

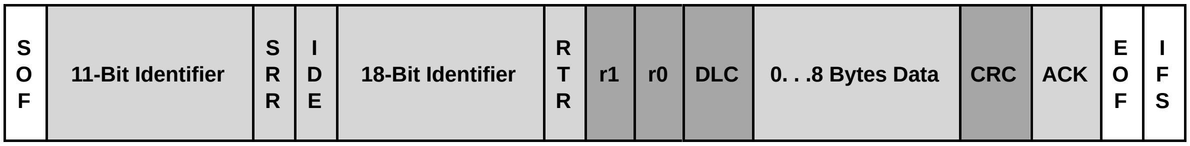

Figure 3. Extended CAN: 29-Bit Identifier

As shown in Figure 3, the Extended CAN message is the same as the Standard message with the addition of:

- SRR—The substitute remote request (SRR) bit replaces the RTR bit in the standard message location as a placeholder in the extended format.
- IDE—A recessive bit in the identifier extension (IDE) indicates that there are more identifier bits to follow. The 18-bit extension follows IDE.
- r1—Following the RTR and r0 bits, an additional reserve bit has been included ahead of the DLC bit.

4 A CAN Message

4.1 Arbitration

A fundamental CAN characteristic shown in Figure 4 is the opposite logic state between the bus, and the driver input and receiver output. Normally a logic high is associated with a one, and a logic low is associated with a zero—but not so on a CAN bus. This is why it is desirable to have the driver input and receiver output pins of a CAN transceiver passively pulled high internally, as in the SN65HVD230. In the absence of any input, the device automatically defaults to a recessive bus state on all input and output pins.

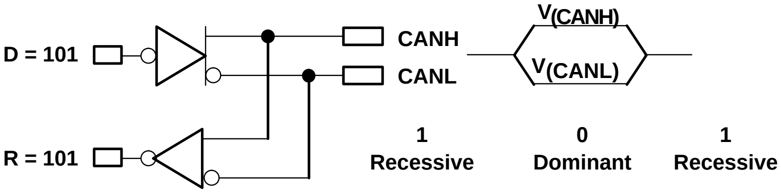

Figure 4. The Inverted Logic of a CAN Bus

Bus access is event-driven and takes place randomly. If two nodes try to occupy the bus simultaneously, access is implemented with a nondestructive, bit-wise arbitration. Nondestructive means that the node winning arbitration just continues on with the message, without the message being destroyed or corrupted by another node.

The allocation of priority to messages in the identifier is a feature of CAN that makes it particularly attractive for use within a real-time control environment. The lower the binary message identifier number, the higher its priority. An identifier consisting entirely of zeros is the highest priority message on a network since it holds the bus dominant the longest. Therefore, if two nodes begin to transmit simultaneously, the node that sends a zero (dominant) while the other nodes send a one (recessive) gets control of the CAN bus and goes on to complete its message. A dominant bit always overwrites a recessive bit on a CAN bus.

Note that a node constantly monitors its own transmission. This is the reason for the transceiver configuration of Figure 4 in which the CANH and CANL output pins of the driver are internally tied to the receiver's input. The propagation delay of a signal in the internal loop from the driver input to the receiver output is typically used as a qualitative measure of a CAN transceiver. This propagation delay is referred to as the loop time (t_{LOOP} in a TI data sheet), but takes on varied nomenclature from vendor to vendor.

Figure 5 displays the arbitration process. Since the nodes continuously monitor their own transmissions, when the node B recessive bit is overwritten by node C's higher priority dominant bit, B detects that the bus state does not match the bit that it transmitted. Therefore, node B halts transmission while node C continues on with its message. Another attempt to transmit the message is made by node B once the bus is released by node C. This functionality is part of the ISO 11898 physical signaling layer, which means that it is contained entirely within the CAN controller and is completely transparent to a CAN user.

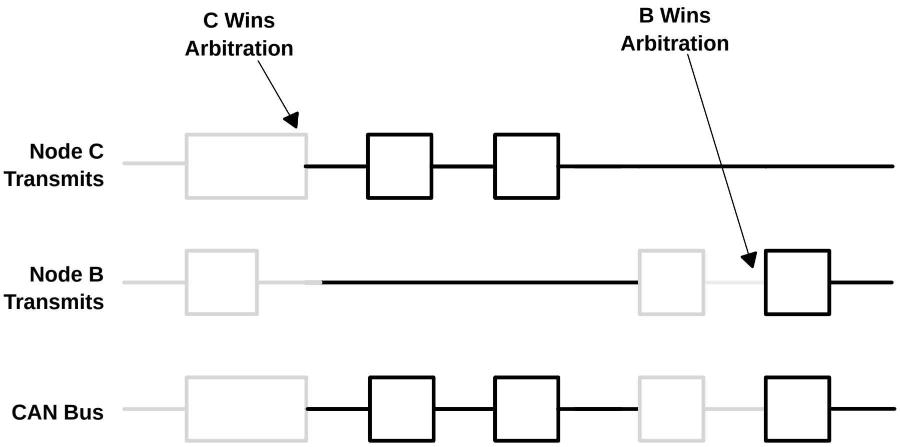

Figure 5. Arbitration on a CAN Bus

The allocation of message priority is up to a system designer, but industry groups mutually agree upon the significance of certain messages. For example, a manufacturer of motor drives may specify that message 0010 is a winding current feedback signal from a motor on a CAN network and that 0011 is the tachometer speed. Since 0010 has the lowest binary identifier, messages relating to current values always have a higher priority on the bus than those concerned with tachometer readings.

In the case of DeviceNet^{TM}, devices from many manufacturers such as proximity switches and temperature sensors can be incorporated into the same system. Since the messages generated by DeviceNet sensors have been predefined by their professional association, the Open DeviceNet Vendors Association (ODVA), a certain message always relates to the specific type of sensor such as temperature, regardless of the actual manufacturer.

4.2 Message Types

There are four different message types, or frames (Figures 2 and 3) that can be transmitted on a CAN bus: the data frame, the remote frame, the error frame, and the overload frame. A message is considered to be error free when the last bit of the ending EOF field of a message is received in the error–free recessive state. A dominant bit in the EOF field causes the transmitter to repeat a transmission.

4.2.1 The Data Frame

The data frame is the most common message type, and is made up by the arbitration field, the data field, the CRC field, and the acknowledgement field. The arbitration field determines the priority of a message when two or more nodes are contending for the bus. The arbitration field contains an 11-bit identifier for CAN 2.0A in Figure 2 and the RTR bit, which is dominant for data frames. For CAN 2.0B in Figure 3 it contains the 29-bit identifier and the RTR bit. Next is the data field which contains zero to eight bytes of data, and the CRC field which contains the 16-bit checksum used for error detection. Lastly, there is the acknowledgement field. Any CAN controller that is able to correctly receive a message sends a dominant ACK bit that overwrites the transmitted recessive bit at the end of correct message transmission. The transmitter checks for the presence of the dominant ACK bit and retransmits the message if no acknowledge is detected.

4.2.2 The Remote Frame

The intended purpose of the remote frame is to solicit the transmission of data from another node. The remote frame is similar to the data frame, with two important differences. First, this type of message is explicitly marked as a remote frame by a recessive RTR bit in the arbitration field, and secondly, there is no data.

4.2.3 The Error Frame

The error frame is a special message that violates the formatting rules of a CAN message. It is transmitted when a node detects an error in a message, and causes all other nodes in the network to send an error frame as well. The original transmitter then automatically retransmits the message. There is an elaborate system of error counters in the CAN controller which ensures that a node cannot tie up a bus by repeatedly transmitting error frames.

4.2.4 The Overload Frame

The overload frame is mentioned here for completeness. It is similar to the error frame with regard to the format, and it is transmitted by a node that becomes too busy. It is primarily used to provide for an extra delay between messages.

4.3 Error Checking and Fault Confinement

The robustness of CAN may be attributed in part to its abundant error checking procedures. The CAN protocol incorporates five methods of error checking: three at the message level and two at the bit level. If a message fails with any one of these error detection methods, it is not accepted and an error frame is generated from the receiving nodes, causing the transmitting node to resend the message until it is received correctly. However, if a faulty node hangs up a bus by continuously repeating an error, its transmit capability is removed by its controller after an error limit is reached.

At the message level are the CRC and the ACK slots displayed in Figures 2 and 3. The 16-bit CRC contains the checksum of the preceding application data for error detection with a 15-bit checksum and 1-bit delimiter. The ACK field is two bits long and consists of the acknowledge bit and an acknowledge delimiter bit. Finally, at the message level there is a form check. This check looks for fields in the message which must always be recessive bits. If a dominant bit is detected, an error is generated. The bits checked are the SOF, EOF, ACK delimiter, and the CRC delimiter bits.

At the bit level each bit transmitted is monitored by the transmitter of the message. If a data bit (not arbitration bit) is written onto the bus and its opposite is read, an error is generated. The only exceptions to this are with the message identifier field which is used for arbitration, and the acknowledge slot which requires a recessive bit to be overwritten by a dominant bit. The final method of error detection is with the bit stuffing rule where after five consecutive bits of the same logic level, if the next bit is not a compliment, an error is generated. Stuffing ensures rising edges available for on-going synchronization of the network, and that a stream of recessive bits are not mistaken for an error frame, or the seven-bit interframe space that signifies the end of a message. Stuffed bits are removed by a receiving node’s controller before the data is forwarded to the application.

With this logic, an active error frame consists of six dominant bits—violating the bit stuffing rule. This is interpreted as an error by all of the CAN nodes which then generate their own error frame. This means that an error frame can be from the original six bits to twelve bits long with all the replies. This error frame is then followed by a delimiter field of eight recessive bits and a bus idle period before the corrupted message is retransmitted. It is important to note that the retransmitted message still has to contend for arbitration on the bus.

5 The CAN Bus

The data link and physical signaling layers of Figure 1, which are normally transparent to a system operator, are included in any controller that implements the CAN protocol, such as Texas Instruments' TMS320LF2407 3.3-V DSP with integrated CAN controller. Connection to the physical medium is then implemented through a line transceiver such as TI's SN65HVD230 3.3-V CAN transceiver to form what the ISO–11898 standard refers to as an electronic control unit (ECU) in Figure 6.

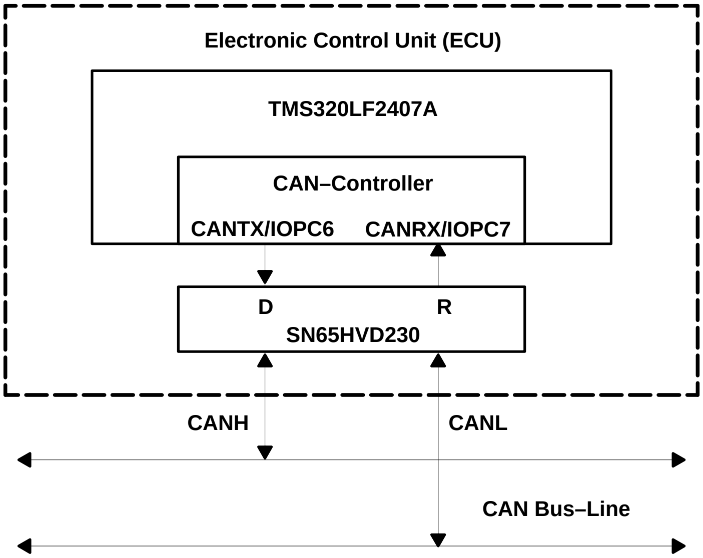

Figure 6. Details of an Electronic Control Unit

Signaling is differential which is where CAN derives its robust noise immunity and fault tolerance. Balanced differential signaling reduces noise coupling and allows for high signaling rates over twisted-pair cable. Balanced means that the current flowing in each signal line is equal but opposite in direction, resulting in a field-canceling effect that is a key to low noise emissions. The use of balanced differential receivers and twisted-pair cabling enhance the common-mode rejection and high noise immunity of a CAN bus.

The two signal lines of the bus, CANH and CANL, in the quiescent recessive state, are passively biased to ≈ 2.5 V. The dominant state on the bus takes CANH ≈ 1 V higher to ≈ 3.5 V, and takes CANL ≈ 1 V lower to ≈ 1.5 V creating a typical 2-V differential signal as displayed in Figure 7.

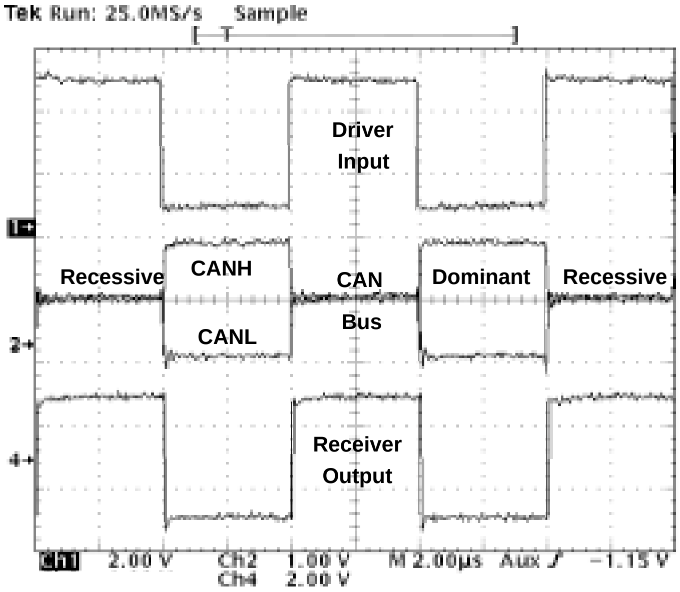

Figure 7. CAN Dominant and Recessive Bus States of the SN65HVD230

The CAN standard defines a communication network that links all the nodes connected to a bus and enables them to talk with one another. There may or may not be a central control node, and nodes may be added at any time, even while the network is operating (hot-plugging).

The nodes could theoretically be sending messages from smart sensing technology and a motor controller. An actual application may include a temperature sensor sending out a temperature update that is used to adjust the motor speed of a fan. If a pressure sensor node wants to send a message at the same time, arbitration assures that the message is sent. For instance, Node A in Figures 8 and 9 finishes sending its message as nodes B and C acknowledge a correct message being received. Nodes B and C then begin arbitration—node C wins the arbitration and sends its message. Nodes A and B acknowledge C’s message, and node B then continues on with its message. Again note the opposite polarity of the driver input and output on the bus.

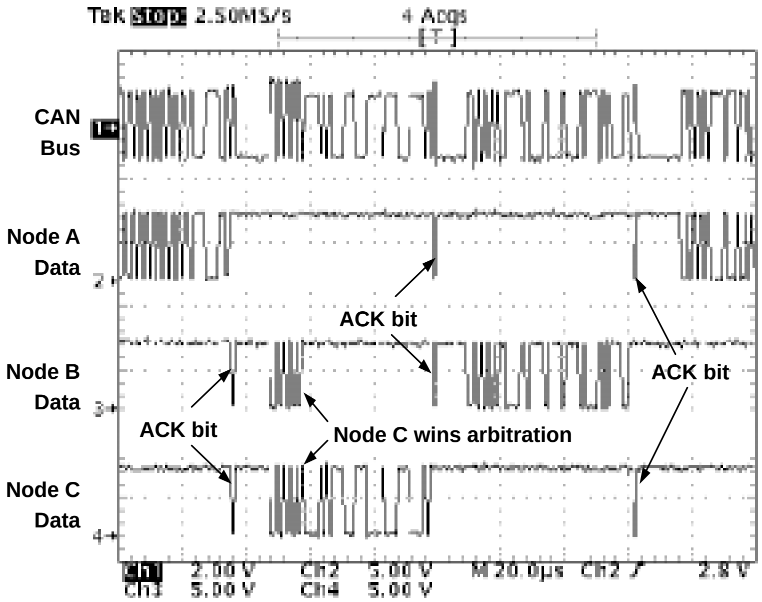

Figure 8. CAN Bus Traffic

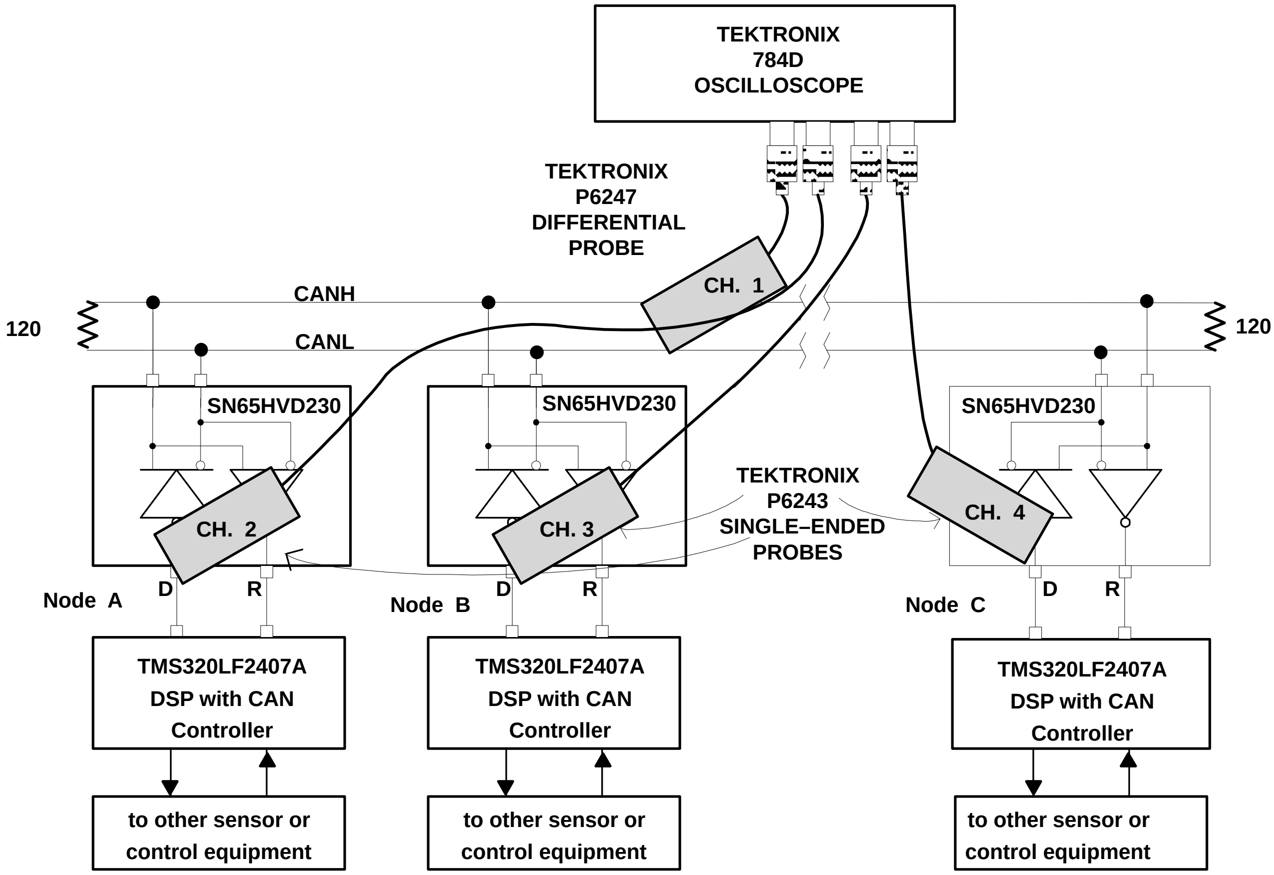
TEKTRONIX 784D OSCILLOSCOPE
TEKTRONIX P6247 DIFFERENTIAL PROBE
CH. 1
CANH
CANL
120
120
SN65HVD230
SN65HVD230
SN65HVD230
CH. 2
CH. 3
CH. 4
TEKTRONIX P6243 SINGLE–ENDED PROBES
Node A
Node B
Node C
D
R
D
R
D
R
TMS320LF2407A DSP with CAN Controller
TMS320LF2407A DSP with CAN Controller
TMS320LF2407A DSP with CAN Controller
to other sensor or control equipment
to other sensor or control equipment
to other sensor or control equipment

Figure 9. CAN Test Bus

5.1 CAN Transceiver Features

5.1.1 *Supply Voltage*

Most CAN transceivers like the SN65HVD251 require a 5-V power supply to reach the signal levels required by the ISO 11898 standard. By superior attention to high-efficiency circuit design, the Texas Instruments SN65HVD23x CAN transceiver family operates with a 3.3-V power supply, and is fully interoperable with 5-V CAN transceivers on the same bus. This allows designers to reduce total node power by 50% or more (Figure 10).

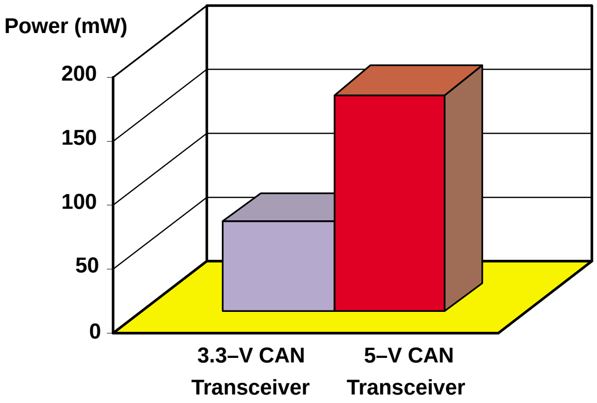

Figure 10. 3.3-V CAN Transceiver Power Savings

For applications using 3.3-V technology, such as the Texas Instruments TMS320C240x™ family of DSPs with integrated CAN controllers, the need for a 5 V power supply can be eliminated. In addition to the inherent power savings of using a 3.3-V transceiver, this lowers the overall part count for the node, thereby reducing system cost and increasing system reliability.

For designers with an existing system design requiring a 5-V powered transceiver, the SN65HVD251 is available as an improved alternate for the commonly-used PCA82C250 and PCA82C251. While the 'HVD251 is a drop-in replacement for the '82C250 and '82C251, it offers the additional features of higher ESD protection, shorter loop delay, and wider common-mode range.

5.1.2 *High Short-Circuit Protection*

The CAN standard recommends that a transceiver survive bus wire short-circuits to each other, to the power supply, and to ground. This ensures that transceivers are not damaged by bus cable polarity reversals, cable crush, and accidental shorts to high power supplies. While the 'HVD23x can survive short-circuits to voltages in the range of –4 V to 16 V, the 'HVD251 provides an extended short-circuit protection to voltages in the –36 V to 36 V range.

5.1.3 *High ESD Protection*

Static charge is an unbalanced electrical charge at rest, typically created by the physical contact of different materials. One surface gains electrons, while the other surface loses electrons. This results in an unbalanced electrical condition known as a static charge. When a static charge moves from one surface to another, it is referred to as an electrostatic discharge (ESD). It can occur only when the voltage differential between the two surfaces is sufficiently high to break down the dielectric strength of the medium separating the two surfaces.

ESD can occur in any one of four ways: a charged body can touch an IC, a charged IC can touch a grounded surface, a charged machine can touch an IC, or an electrostatic field can induce a voltage across a dielectric sufficient to break it down. It becomes readily apparent that a high ESD rating not only indicates a robust transceiver, but a robust circuit design as well.

While the earliest CAN transceivers on the market have only a 2 kV or 4 kV ESD maximum on the bus pins, the HVD2xx families of CAN transceivers have ESD ratings as high as 16 kV on the bus pins when tested in accordance with the human body model (HBM) of JEDEC Standard 22, Test Method A114-A. With these ESD ratings, the TI HVD2xx CAN transceivers are much better suited to the harsh electrical environments of automotive and industrial applications than the earlier transceiver versions of other vendors.

5.1.4 Wide Common-Mode Range

Common-mode noise is the difference in potential between grounds of sending and receiving nodes on a bus. This is often the case in the networked equipment typically found in a CAN application. Possible effects of this problem are intermittent reboots, lockups, bad data transfer, or physical damage to the transceiver.

Network interface cards, parallel ports, serial ports, and especially transceivers are prime targets for some form of failure if not designed to accommodate high levels of ground noise and power supply imbalance between typical CAN nodes.

With this in mind the ‘HVD230 family is designed to operate with complete safety over the bus voltage range of –2 V to 7 V required by the standard. However, the ‘HVD251 extends this safe operating range even farther with operation from –7 V to 12 V.

5.1.5 Common-Mode Rejection

Common-mode noise of varied magnitudes exists within the networks associated with CAN applications. Noise from pulsing motor controllers, switch-mode power supplies, or from fluorescent lighting are the typical sources of noises that couple onto bus lines (displayed in Figure 11).

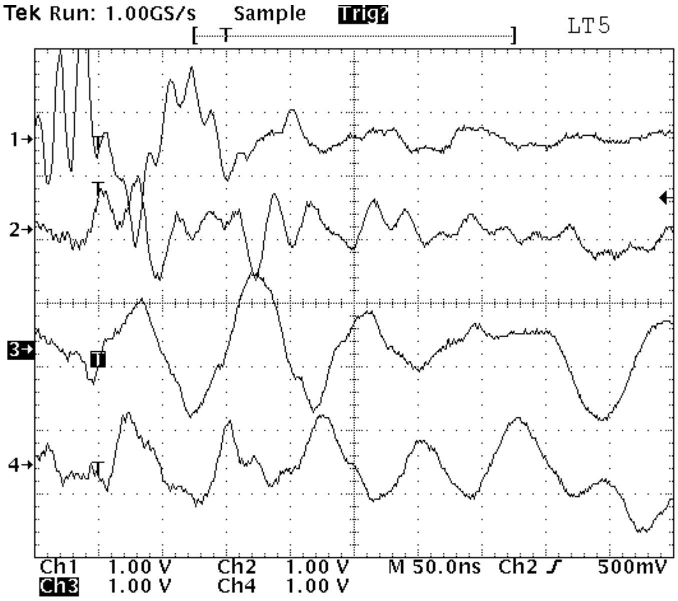

Figure 11. Common-Mode Noise Coupled onto 4 Twisted-Pair Bus Lines

A CAN transceiver not specifically designed to reject this coupled noise can respond to common-mode noise as if it were data on a bus and send meaningless data to a controller. The ‘HVD23x and ‘HVD251 are specifically designed and tested for their ability to reject this common-mode noise.

5.1.6 High Input Impedance

A high bus input impedance increases the number of nodes that can be added to a bus above the standard’s 30 nodes. The high impedance restricts the amount of current that a receiver sinks or sources onto a bus over common-mode voltage conditions. This ensures that a driver transmitting a message into such a condition is not required to sink or source an excessive amount of current from the sum of the receiver currents on the bus.

5.1.7 Controlled Driver Output Transition Times

Controlling driver output slew-rate dampens the rise-time of a dominant bit to improve signal quality and provides for longer stub lengths and a better bit-error rate.

5.1.8 Low Current Standby and Sleep Modes

Many applications are looking to lower-power opportunities as more electronics are added to designs. The standby mode of the ‘HVD230 is generally referred to as the *listen only* mode, since in standby, the ‘HVD230’s driver circuitry is switched off while the receiver continues to monitor bus activity. In the occurrence of a dominant bit on the bus, the receiver passes this information along to its DSP/CAN controller which in turn activates the circuits that are in standby.

The difference between the ‘HVD230 and ‘HVD231 is that both driver and receiver circuits can be switched off in the ‘HVD231 to create an extremely low-power sleep mode.

5.1.9 Thermal Shutdown Protection

Another desirable safety feature for a CAN transceiver is the thermal shutdown circuitry of the ‘HVD23x and ‘HVD251. This feature protects a device against the destructive currents and resulting heat that can occur in a short–circuit condition. Once thermal shutdown is activated, the device remains shut down until the circuitry is allowed to cool.

5.1.10 Glitch Free Power Up and Power Down

This feature provides for hot–plugging onto a powered bus without disturbing the network. The ‘HVD23x’s and ‘HVD251’s driver and receiver pins are passively pulled high internally while the bus pins are biased internally to a high-impedance recessive state. This provides for a power-up into a known recessive condition without disturbing ongoing bus communication.

5.1.11 Unpowered Node Does Not Disturb the Bus

Several CAN transceivers on the market today have very low output impedance when un–powered. This low impedance causes the device to sink any signal present on the bus and shuts down all data transmission. The ‘HVD23x and ‘HVD251 have a very high output impedance in powered and unpowered conditions, and maintain the integrity of the bus when power or ground is removed from the circuit.

5.2 The Relationship Between Bus Length and Signaling Rate

The basics of arbitration require that the front wave of the first bit of a message travel to the most remote node on a network and back again before the bit is designated by the receiver of the sending node as dominant or recessive (typically this *sample* is made at about two–thirds the bit width). With this limitation, the maximum bus length and signaling rate are determined by network parameters.

Factors to be considered in network design include the ≈ 5 ns/m propagation delay of typical twisted-pair bus cable, signal amplitude loss due to the loss mechanisms of the cable, and the input impedance of the bus transceivers. Under strict analysis, variations among the different oscillators in a system also need to be accounted for with adjustments in signaling rate and bus length. Maximum signaling rates achieved with the SN65HVD230 in high-speed mode with several bus lengths are listed in Table 2.

Table 2. Maximum Signaling Rates for Various Cable Lengths

BUS LENGTH (m)	SIGNALING RATE (kbps)
30	1000
100	500
250	250
500	125
1000	62.5

The ISO 11898 standard specifications are given for a maximum bus length of 40 m and maximum stub length of 0.3 m with a maximum of 30 nodes. However, with careful design, longer cables, longer stub lengths, and many more nodes can be added to a bus—always with a trade-off in signaling rate. A transceiver with high input impedance such as the HVD230 is needed to increase the number of nodes on a bus.

The cable is specified to be a shielded or unshielded twisted-pair with a 120-Ω characteristic impedance (Zo). The standard defines the interconnection to be a single twisted-pair cable. The interconnection is terminated at both ends with a resistor equal to the characteristic impedance of the line to prevent signal reflections. Nodes are then connected to the bus with unterminated drop cables, or stubs, which should be kept as short as possible to minimize signal reflections.

Connectors, while not specified by the standard should not affect standard operating parameters such as the minimum VOD. Although unshielded cable is used in many applications, data transmission circuits employing CAN transceivers are used in applications requiring a rugged interconnection with a wide common-mode voltage range. Therefore, shielded cable is recommended in these electronically harsh environments, and when coupled with the standard's –2 V to 7 V common-mode range of tolerable ground noise, helps to ensure data integrity.

6 Conclusion

CAN is ideally suited in applications requiring a high number of short messages in a short period of time with high reliability in rugged operating environments. Since CAN is message based and not address based, it is especially suited when data is needed by more than one location and system-wide data consistency is mandatory.

Fault confinement is also a major benefit of CAN. Faulty nodes are automatically dropped from the bus, which prevents any single node from bringing a network down, and assures that bandwidth is always available for critical message transmission. This error containment also allows nodes to be added to a bus while the system is in operation, otherwise known as hot-plugging.

The many features of the 'HVD23x and 'HVD251 make them ideally suited for the many rugged applications to which the CAN protocol is being adapted. Among the applications finding solutions with CAN are automobiles, trucks, trains, busses, airplanes, and agriculture, construction, mining, and marine vehicles. CAN-based control systems are being used in factory and building automation and embedded control systems for machines, medical devices, domestic appliances and many other applications.

7 Additional Reading

1. Controller Area Network, Basics Protocols, Chips and Applications; Dr. Konrad Etschberger; ISBN 3–00–007376–0 (www.ixxat.com)

2. CAN Systems Engineering, From Theory to Practical Applications; Wolfhard Lawrenz, ISBN 0–387–94939–9

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third–party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265