SAE TECHNICAL PAPER SERIES

930005

A Gateway for CAN Specification 2.0 Non-Passive Devices

Craig Szydlowski
Intel Corp.

Reprinted from:
Multiplex Technology Applications in Vehicle Electrical Systems (SP-954)

International Congress and Exposition
Detroit, Michigan
March 1-5, 1993

SAE The Engineering Society For Advancing Mobility Land Sea Air and Space®
INTERNATIONAL

SAE routinely stocks printed papers for a period of three years following date of publication. Direct your orders to SAE Customer Sales and Satisfaction Department.

Quantity reprint rates can be obtained from the Customer Sales and Satisfaction Department.

To request permission to reprint a technical paper or permission to use copyrighted SAE publications in other works, contact the SAE Publications Group.

GLOBAL MOBILITY *DATABASE*

All SAE papers, standards, and selected books are abstracted and indexed in the SAE Global Mobility Database.

ISSN 0148-7191

Positions and opinions advanced in this paper are those of the author(s) and not necessarily those of SAE. The author is solely responsible for the content of the paper. A process is available by which discussions will be printed with the paper if it is published in SAE transactions. For permission to publish this paper in full or in part, contact the SAE Publications Group.

Persons wishing to submit papers to be considered for presentation or publication through SAE should send the manuscript or a 300 word abstract of a proposed manuscript to: Secretary, Engineering Activity Board, SAE.

Printed in USA

930005

A Gateway for CAN Specification 2.0 Non-Passive Devices

Craig Szydlowski
Intel Corp.

ABSTRACT

The Controller Area Network (CAN) protocol, developed by ROBERT BOSCH GmbH, offers a comprehensive solution to managing communication between multiple CPUs. In September 1991, the CAN protocol was revised (CAN Specification 2.0) to add an extended message format that increases the number of permitted message identifiers. The CAN protocol now supports 11- and 29-bit message identifiers allowing standard and extended formats, respectively.

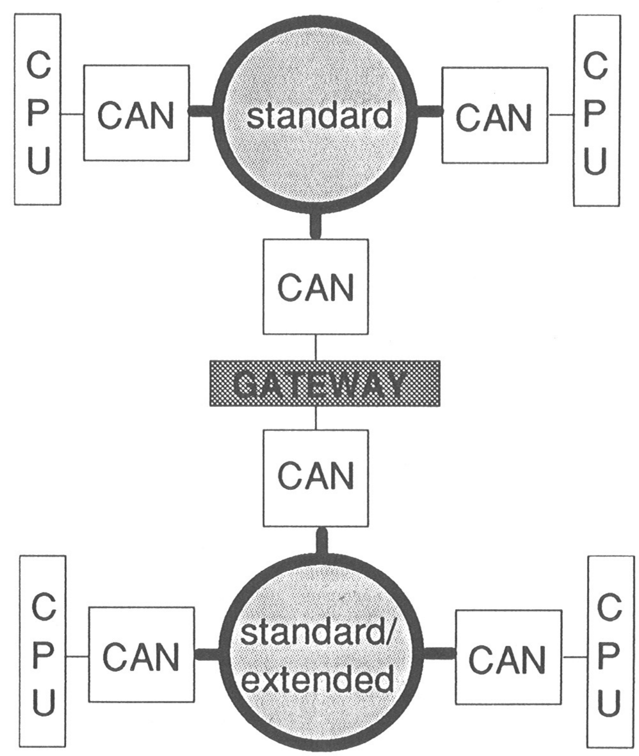
CAN Specification 2.0 implements a new message bit, the identifier extension bit (IDE bit) which allows CAN devices to differentiate standard and extended formats. However, most existing CAN implementations are based on the previous CAN protocol specification and will not recognize extended format messages and will respond with an error message. These chips are CAN Specification 2.0 non-passive.

Until semiconductor makers develop a variety of chips to implement 29-bit message identifiers, CAN users may require a gateway to interconnect networks using 29-bit message identifiers to existing networks that only use CAN 1.2 11-bit message identifiers. In this context, a gateway is a system that translates messages in one protocol to another and vice versa.

A gateway between two networks is typically designed to minimize transmission latency, to minimize lost messages (overruns) and to manage bus error issues. Following a brief description of the CAN message format, general gateway design considerations and performance estimates will be addressed. In addition, a method to send remote frames across the gateway is discussed.

GATEWAY EXAMPLE

The gateway considered in this paper is shown in figure 1. The microcontroller in the center, labeled 'GATEWAY', translates messages between the networks above and below. The network labelled 'standard' transmits CAN 1.2 standard messages (11 -bit identifier) while the network labeled 'standard/extended' transmits CAN 2.0 both standard and extended messages (11- and 29-bit identifiers). The standard messages for the CAN 1.2 and CAN 2.0 networks are bit-wise and electrically identical.



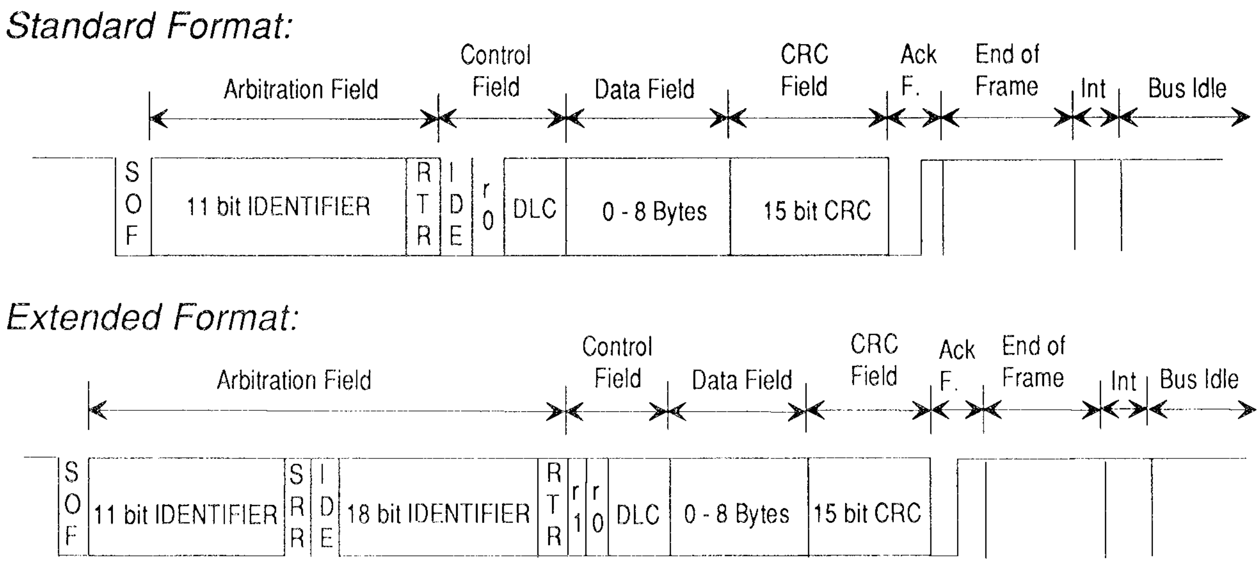


Figure 2: Standard and Extended Formats

The gateway microcontroller communicates with two CAN chips, one from each network. Examples of gateway functions are:

a) bridging standard messages without translation.
b) bridging standard and extended messages with message identifier translation.
c) managing re-transmission issues when errors occur.

CAN PROTOCOL MESSAGE FORMATS

The CAN protocol supports two message formats which differ in the length of the message identifier field. Messages using 11 message identifier bits are called standard and those using 29 message identifier bits are called extended. Both standard and extended message formats support four frame types:

1) Data - carries data
2) Remote - sent when a node requests data from another node
3) Error - sent when a node detects a message error
4) Overload - sent when a node requires extra delay

The following describes the standard and extended message formats for data and remote frames shown in figure 2.

SOF: Start Of Frame (dominant bit) marks the beginning of a data/remote frame.

Arbitration: One or two fields which contain the message identifier bits. The standard format has one 11-bit field and the extended format has two fields, 11- and

RTR: Remote Transmission Request bit is dominant for data frames and recessive for remote frames. This bit is in the arbitration field.

SRR: Substitute Remote Request bit is used in extended messages and is recessive. This bit is a substitute for the RTR bit in the standard format. This bit is in the arbitration field of the extended format.

IDE: Identifier Extension bit is dominant for standard format and recessive for extended format. This bit is in the arbitration field of the extended format and in the control field of the standard format.

Control Field: Reserved bits r0 and r1 are sent as dominant bits. The 4-bit Data Length Code (DLC) indicates the number of bytes in the data field.

Data Field: The data bytes are located in the data frame (0-8 bytes). A remote frame contains zero data bytes.

CRC Field: This field is composed of a 15-bit Cyclical Redundancy Code error code and a recessive CRC delimiter bit.

ACK Field: Acknowledge is a dominant bit sent by nodes receiving the data/remote frame and is followed by a recessive ACK delimiter bit.

INT: Intermission is the three recessive bits which separate data and remote frames.

The minimum message lengths of standard and extended message formats are summarized below for data and remote frames. These bit counts will be used to assess gateway message transmission latency and overrun susceptibility. The actual lengths of these messages may differ because 'stuff' bits are added to the message. Stuff bits assist synchronization by adding transitions to the message. A stuff bit is inserted in the bit stream after five consecutive-equal value bits are transmitted; the stuff bit is the opposite polarity of the five consecutive bits. All message fields are stuffed except the CRC delimiter, the ACK field and the End of Frame.

Standard Format

Message Field		**number of bits**
SOF		1
Arbitration Field		12
identifier	11	
RTR	1	
Control Field		6
IDE	1	
r0	1	
DLC	4	
Data Field		0-64
CRC Field		16
ACK Field		2
End of Frame		7
Total		44-108 bits

Extended Format

Message Field		**number of bits**
SOF		1
Arbitration Field		32
identifier	29	
SRR	1	
IDE	1	
RTR	1	
Control Field		6
r1,r0	2	
DLC	4	
Data Field		0-64
CRC Field		16
ACK Field		2
End of Frame		7
Total		64-128 bits

HARDWARE CONFIGURATION

For this analysis, it is assumed both CAN implementations are stand-alone chips (figure 3). The gateway microcontroller reads and writes to the CAN of the receiving CAN chip. Next, the gateway executes logical instructions to translate the message. Then the gateway executes external writes to the CAN device on the second network to program a transmission. Therefore, the microcontroller executes external read and write operations and translates message identifiers.

READ AND WRITE OPERATIONS - The time required to execute read and write operations is dependent upon the interface-timing characteristics of the gateway microcontroller and the CAN chips. Figure 3 shows a microcontroller interfaced to two CAN devices. This configuration uses a 16-bit address/data bus which doubles the communication throughput between the microcontroller and the CAN chips. The CAN chips drive independent interrupt lines to the microcontroller allowing the interrupting CAN device to be easily identified.

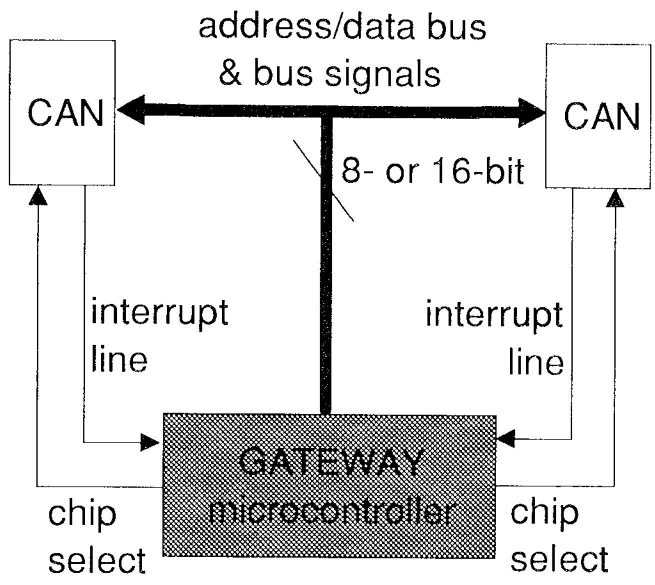


Figure 3: Hardware Example

CAN CHIP MESSAGE CONFIGURATION - The CAN chips must be configured to receive and transmit messages. Typically, CAN chips support 2 to 15 message objects that are configured to receive or transmit. Message objects consist of RAM bytes which store a message identifier, data bytes and control bytes. To receive messages more reliably, a CAN chip must be able to receive an incoming message while it is still processing a previously received message. This may be accomplished 1) by using a buffered receive message object or 2) by implementing a buffered receive using two receive message objects that are alternately validated and invalidated (figure 4). It is assumed that all receive message objects employ acceptance mask filtering which allows all relevant CAN bus messages to be accepted by a single buffered receive message object. Without

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase® Smarter legal research.