Downloaded from SAE International by Iobd xtool, Thursday, February 13, 2014 08:49:27 PM

SAE TECHNICAL PAPER SERIES

940361

CAN Controller Architecture Optimized for SAE-J1939 Applications

Peter Buehring
Philips Semiconductors

Reprinted from: Multiplexing and Fiberoptics (SP-1012)

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

**International Congress & Exposition
Detroit, Michigan
February 28-March 3, 1994**

400 Commonwealth Drive, Warrendale, PA 15096-0001 U.S.A. Tel: (412)776-4841 Fax:(412)776-5760

Downloaded from SAE International by Iobd xtool, Thursday, February 13, 2014 08:49:27 PM

The appearance of the ISSN code at the bottom of this page indicates SAE's consent that copies of the paper may be made for personal or internal use of specific clients. This consent is given on the condition, however, that the copier pay a $5.00 per article copy fee through the Copyright Clearance Center, Inc. Operations Center, 222 Rosewood Drive, Danvers, MA 01923 for copying beyond that permitted by Sections 107 or 108 of the U.S. Copyright Law. This consent does not extend to other kinds of copying such as copying for general distribution, for advertising or promotional purposes, for creating new collective works, or for resale.

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 Global Mobility Database.

No part of this publication may by reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

ISSN 0148-7191
Copyright 1994 Society of Automotive Engineers, Inc.

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 90-1203C/PG

Downloaded from SAE International by Iobd xtool, Thursday, February 13, 2014 08:49:27 PM

940361

CAN Controller Architecture Optimized for SAE-J1939 Applications

Peter Buehring
Philips Semiconductors

ABSTRACT

An electronic module in a J1939-compatible network requires to be equipped with a CAN protocol controller, which might be integrated on the main CPU or is attached to the CPU as a separate chip. Besides the fundamental functions required to transmit and receive CAN-compatible messages, the CAN hardware can support the CPU effectively, if the typical communication structure of J1939 was respected during the design of the CAN controller. It is shown that the CPU's management of, especially, received messages can be accelerated, if the CAN controller's architecture reflects the internal structure of the J1939 recommendations.

1. Introduction

The communication protocol CAN ("Controller Area Network") has been standardized at the ISO committee for in-vehicle networks /1,2/. The ISO standard specifies the layers 1 and 2 (Physical Layer and Data Link Layer) /3/ only, with the upper layers to be defined by the car manufacturer individually. Mercedes Benz was the first company whose passenger cars had been equipped with electronic control based on CAN communication.

In 1991 the CAN protocol had been extended with a second message format providing a message identifier of 29 bits length (the first format uses an 11-bit identifier). Now the SAE accepted CAN as a basis for definition of the in-vehicle network J1939 /4/ for use in trucks and busses (figure 1). J1939 defines all relevant layers of the communication, thus enabling different manufacturers to build electronic modules performing compatible bus communication.

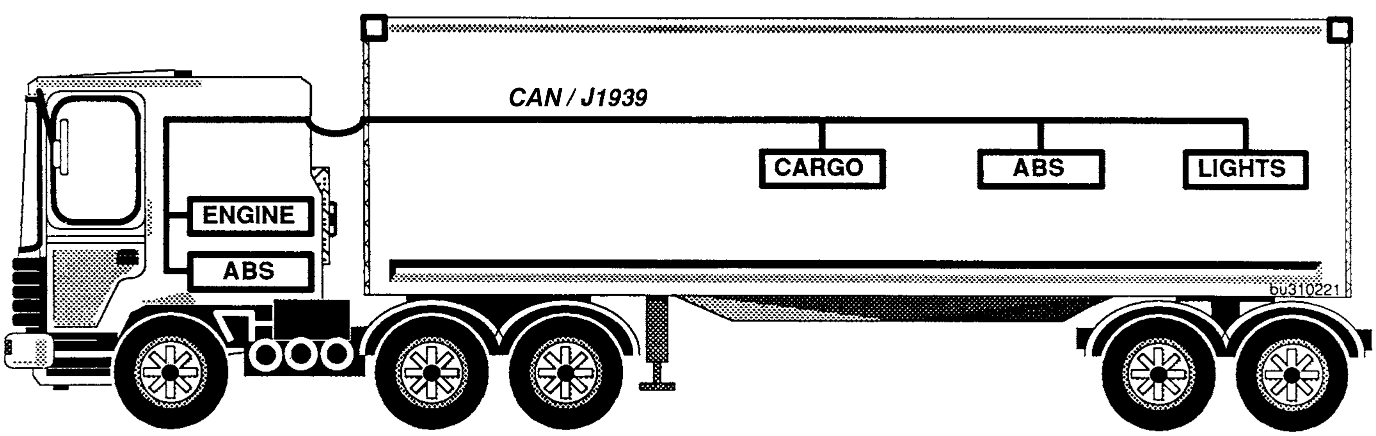

***Figure** 1 Example of communication between electronic modules via CAN / J1939.*

Downloaded from SAE International by Iobd xtool, Thursday, February 13, 2014 08:49:27 PM

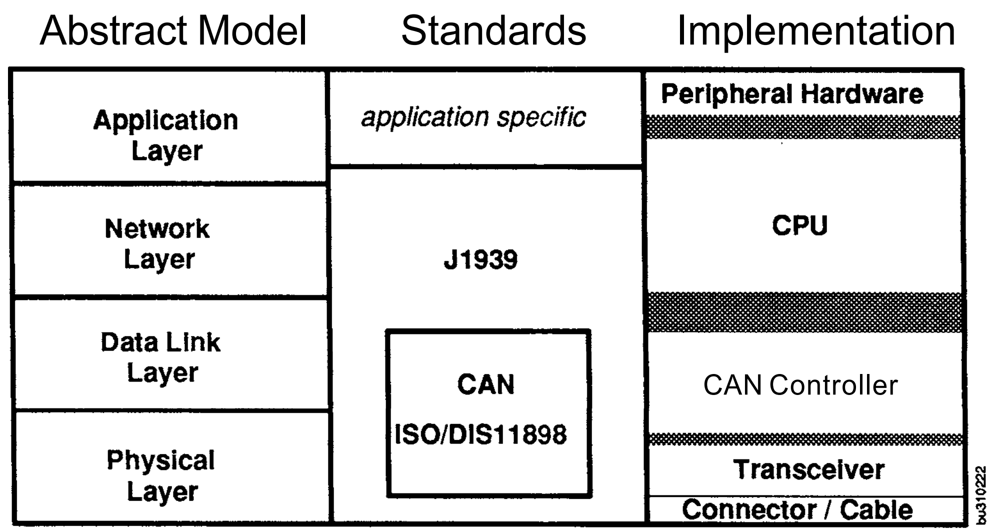

Figure 2 *The communication tasks of the different layers are distributed to the module's components.*

The functionality of an electronic module, which is connected to the bus, comprises the control of the local application as well as the communication functions required by the network. For the application usually a microcontroller is used, and at least for the control of the lower communication layers dedicated hardware is required, which might be integrated on the microcontroller or is a separate chip. Figure 2 illustrates the relation of the different layers to the module's components. The location of the borderlines between the devices can vary according to different implementation possibilities.

In the past, when no standards for the higher layers were available, a CAN controller's performance was mainly restricted to the original CAN protocol functions. All other communication tasks had to be executed by the microcontroller thus reducing the processing power being left for the application.

With J1939 there is a specification available describing the communication tasks of all layers. Therefore it was worth thinking about integration of J1939-specific functions into the CAN controller's hardware.

For determination of the additional functions to be integrated, an analysis of J1939 has been made. The functions which were appropriate for hardware implementation have been identified. By checking the requirement, that the CPU's remaining overhead for communication should be as low as possible, a decision could be made what the CAN controller's new architecture should look like.

2. Processing CAN messages according to J1939

Within J1939 a message is called a Protocol Data Unit (PDU), of which two formats exist. The one format is used for transmission of messages to specific destinations and therefore the destination address is part of the CAN identifier (see figure 3). The other

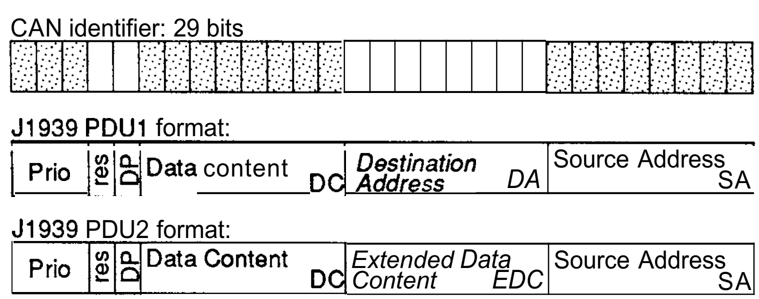

Figure 3 *Partitioning of the CAN identifier according to the two formats of J1939.*

Downloaded from SAE International by Iobd xtool, Thursday, February 13, 2014 08:49:27 PM

format is taken for broadcasted information without specifying a destination. The abbreviation "prio" stands for "priority", "res" is "reserved", and "DP" is "Data Page", which may double the range of Data Contents if set to 1.

Whether a message has to be received by a node or not can be determined by checking its identifier. PDU1 format messages are treated in a different way than those of PDU2 format. But even messages with the same PDU format require different reactions of the module depending on the identifier (see figure **4**):

case A: the reception of data has to be confirmed by replying an acknowledge message, or a not-acknowledge message has to be sent if the Data Content is unknown, respectively.

case B: the reception of data has not to be confirmed.

case C: a global request for data requires reply, if the data is available

case D: it is not data that is received but a message for communication control (e.g. an acknowledge is received or a multipacket message is started). This requires special reaction depending on the history.

case E: all other messages don't have to be received.

In Table 1 the identifier specifications relating to the cases A-E are listed.

The transmission of a message is more uniform. It may be caused by the application itself, or by reception of a message requiring reply.

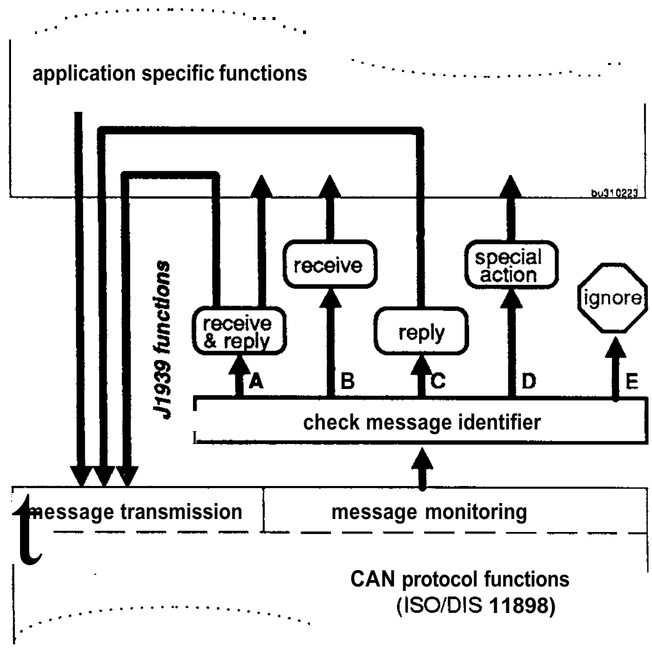

Figure 4 *Different cases for handling received messages.*

3. Requirements on the CAN controller

There is the requirement that the module's microcontroller should not be burdened too much with the control of the communication. This implies that at least the following is supported by the CAN controller:

» The CAN controller should not pass on messages to the CPU which are not of interest for the module.

» The CPU's effort for determination of the contents of received messages should be as low as possible. But the identifier specifications for the different cases of reception are not trivial, as it was shown in the previous section. Obviously the sorting of the received messages requires support by the CAN controller.

Case	Mode	PDU Format	Identifier Specification	Procedure
A	point-to-point (data)	PDU1 (DA=specific)	$0 \le DC \le 231 \wedge DA=$node address	if DC known then receive data and send ack, else send nack
B	broadcast (data)	PDU1 (DA=global), or PDU2	$DC \in \{list\} \wedge DA=255 \vee DC,EDC \in \{list\}$	receive data
C	global request	PDU1	$DC=234 \wedge DA=255$	respond with data, if available
D	point-to-point (management)	PDU1 (DA=specific)	$232 \le DC \le 239 \wedge DA=$node address	special actions according to history
E	don't care	don't care	all other identifiers	no action

Acronyms: DA = Destination Address, DC = Data Content, EDC = Extended Data Content, PDU = Protocol Data Unit, refer to figure 3.

Table 1 *A node can determine the relevance of a received message by checking the identifier.*

Downloaded from SAE International by Iobd xtool, Thursday, February 13, 2014 08:49:27 PM

» Some application tasks might not be interruptable. Hence the CPU would not be able to fetch the received data immediately after arrival. Of course this should not mean that data is lost.

» Parts of the identifier contain information which has to be read. This can be the Source Address which has to be taken for the Destination Address of a response, for example. The interesting identifier parts are all of 1-byte size. Of course the CAN controller should store the messages' identifiers byte-wise accordingly.

Even if the above said is supported by the CAN controller, there will remain some CPU overhead needed for communication. But in most cases the application will leave some percent of processing power, which is sufficient for this purpose. In the other cases it might be desirable that the CAN controller gives even considerably more support, i.e. automatically sends back acknowledge messages, for instance.

4. Philips' new CAN controller architecture

It was the result of investigations, that the basic requirements on a CAN controller as listed in section 3, could be implemented with dedicated hardware. On the other hand, the additional features mentioned in the last section would lead to a quite complex "J1939 communication processor" which is associated with software running on an additional microcontroller inside. But again, the latter would include the same CAN hardware, which is presented in this section.

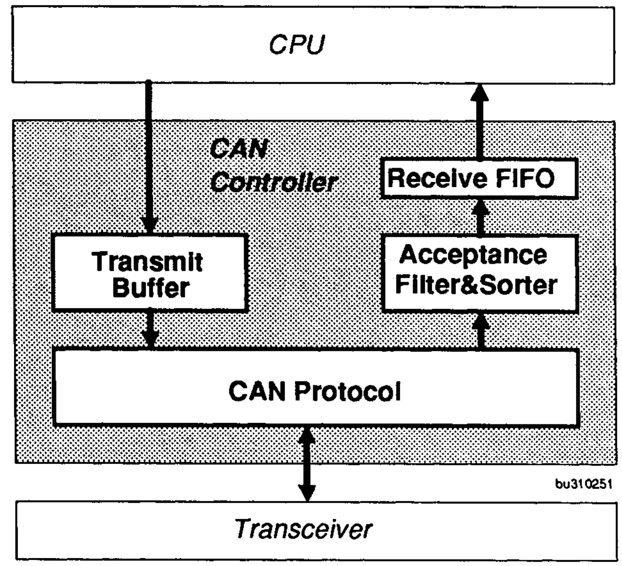

Figure 5 *Block diagram of the new CAN controller.*

The architecture of the CAN controller satisfying the needs of J1939 is shown in figure 5. While the Transmit Buffer and the Receive Buffer (FIFO) are not elements which could be contemplated as being J1939-specific, the Acceptance Filter & Sorter (AFS) is the part whose design was strongly influenced by the J1939 specification.

The task of the AFS is to suppress reception of unwanted messages, and to sort the received messages according to the cases described in section 2. The AFS provides 16 "channels" with different configuration possibilities each (see figure 6)

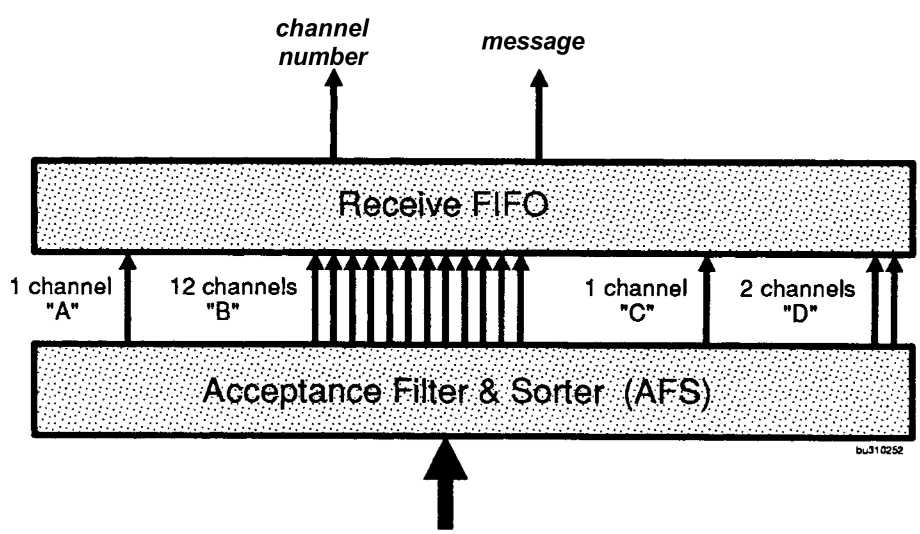

Figure 6 *The Acceptance Filter rejects unwanted messages and sorts relevant ones according to the kind of processing **by** the CPU.*

Downloaded from SAE International by Iobd xtool, Thursday, February 13, 2014 08:49:27 PM

according to the different identifier specifications for these cases.

Channel " A recognizes all messages related to "case A , similarly the other channels detect the cases B, C and D. With the 12 channels "B" a list of 12 Data Contents can be specified for reception. The 2 channels "D" allow for the defining of 2 subgroups relating to "case D", for even more flexibility.

Simple acceptance filters known from former CAN controllers were only allowed to specify the single identifier bits to "0","1" or "don't care". This was not sufficient in all cases, as for recognition of PDU1-formats ("case A) , for instance, the filter has to check if the Data Content is lower than 232. Therefore the design of a more complex acceptance filter was required.

In the FIFO the received messages (identifier+data) are stored together with the channel number serving as index for the different ways of data processing by the CPU. This enables the effective use of jump tables, for instance. The FIFO can store up to 10 messages, allowing the CPU the delayed reading of the received data.

In fact the new CAN controller meets all the basic requirements for an effective support of the CPU's J1939 communication.

5. Integration on Microcontrollers

As the CAN controller relieves the CPU of time-consuming communication control, the requirements on the CPU's size can now be reduced. There are many in-vehicle functions where the power of 8-bit microcontrollers is sufficient. In conjunction with the new CAN controller the use of an 80C51 family microcontroller can be a very cost-effective solution, especially if the CAN controller is integrated on the µC.

Philips Semiconductors is currently developing an 80C51 derivative including the new CAN controller. Together with the CAN transceiver 82C251 a compact node in a J1939 network can be designed, which mainly consists of two chips, only (see figure 7).

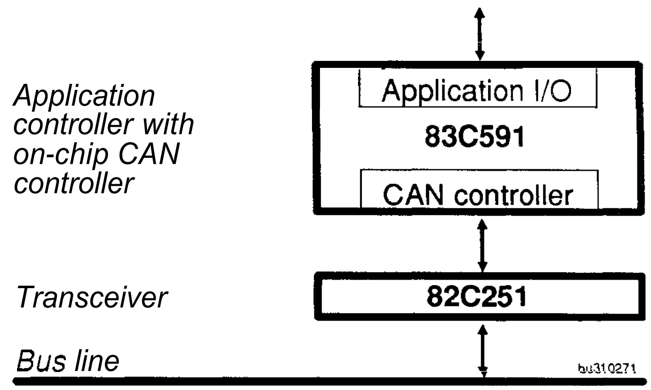

Figure 7 *Two-chip* module set-up.

6. Summary

In the CAN controllers, which had been developed in the past, there was no support of higher communication layers incorporated. The lack of standards for these layers prevented such implementation. J1939 is a specification covering all applicable communication layers of an in-car network. Philips' new architecture for CAN controllers has been introduced, which provides J1939-specific features reducing significantly the CPU's software overhead for compliance with J1939.

References

/1/ Road Vehicles - Interchange of Digital Information - Controller Area Network (CAN) for High Speed Communication. Document ISOIDIS 11898, ISO Draft for International Standard, 1991

/2/ Road Vehicles - Serial Data Communication for Automotive Application, Part 1: Controller Area Network (CAN). Document ISOIDIS 11519, ISO Draft for International Standard, 1991

/3/ Information processing systems - Open Systems Interconnection - Basic reference model. Document ISO 7498, ISO International Standard, 1983

/4/ Recommended Practice for a Serial Control and Communications Vehicle Network (Class C). Draft J1939, SAE, 1993