

# Application Note

## **Extended Frame Format - A New Option of the CAN Protocol**

Hans-Christian Reuss

Product Concept & Application Laboratory Hamburg, F. R. Germany

### **Keywords**

CAN Protocol Specification Vers. 2.0 A+B, Standard Frame Format, Extended Frame Format, Compatibility, Bus Performance, Products

**Report No** : HAI/AN 92 002  
**Date** : 93-05-04  
**Pages** : 9

©Philips Export B.V.

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner. The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

**Philips Semiconductors**



**PHILIPS**

**Summary:**

In addition to the CAN "standard frame format", which is specified with an 11 bit identifier, Bosch introduced in 1991 the CAN "extended frame format", which operates with a 29 bit identifier. This paper contains a description and a comparison of both frame formats. The result of the comparison shows, that both formats have their own advantages; the advantage of the extended frame format is the higher number of different message types, the advantage of the standard frame format are the lower bus access time, the higher bus throughput and the availability of products. Therefore it is recommended to use only standard frames, as long as the network communication can be managed with 2032 different message types.

Revision history:

92-12-15: 1st release

93-05-04: 2nd release: pages 0,8(,9) revised

Table of Contents:

1.	Introduction .....	2
2.	History of the CAN specifications .....	2
3.	Standard and Extended Frame Format .....	3
4.	Comparison .....	5
5.	Compatibility Aspects .....	7
6.	Available Products .....	8
7.	Conclusion .....	8
	References .....	9

## 1. Introduction

Once specified by Bosch, CAN ('Controller Area Network') is an advanced serial communication protocol, which efficiently supports distributed real-time control and multiplexing with a very high safety level. Typical applications of CAN-based networks can be found in automotive and industrial environments.

Originally CAN message frames have contained 11 bit identifiers. With the CAN protocol specification vers. 2.0 Bosch has introduced in 1991 a second CAN message frame format: the extended frame format, which is based on a 29 bit identifier increasing the number of different identifiers considerably. This new format is only optional, that means that message frames with the 11 bit identifier will stay the 'standard message frames'.

The objective of this paper is the comparison of the standard and the extended frame format with their advantages and disadvantages. Criteria being considered for this comparison are the bus access time, the bus throughput, the CPU-load, and the available products.

The reader should be familiar with CAN fundamentals. Information is available e.g. in the CAN protocol specification vers. 2.0 /1/ and in the Philips application notes and data sheets.

In accordance to the CAN protocol specification vers. 2.0 frames with the number of 11 identifier bits are denoted with standard frame, and frames with the number of 29 identifier bits are denoted with extended frame.

## 2. History of the CAN specifications

Since 1985, when the CAN protocol specification was published first, it has become the basis for several integrated circuits being produced by the semiconductor manufacturers. Table 1 shows the history of the CAN protocol specification with all versions being provided until today. It is important to mention that every new version is compatible to all older ones (not vice versa).

Version	Year	Changes
1.0	1985	
1.1	1987	Respecification of bit timing requirements
1.2	1990	Increased oscillator tolerance
2.0	1991	Part A - Equal to vers. 1.2
		Part B - Introduction of optional second format "extended frame" for data and remote frames

Table 1 CAN Protocol Specification Versions

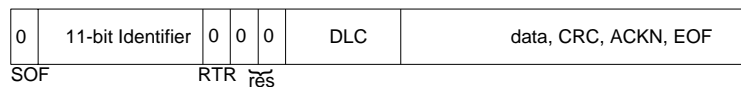
Between version 1.0 and 1.2 only slight modifications have been implemented: in version 1.1 the bit timing requirements have been respecified in order to increase the performance and the understandability; in version 1.2 few modifications at the interpretation of dominant bits during intermission or during the error/overload delimiters have been introduced, thus the oscillator tolerance has been increased and the use of ceramic oscillators instead of crystal driven ones has been allowed.

Version 2.0 consists of two parts: 2.0 A is equal to 1.2, and 2.0 B is again the same specification but it supports also the new optional extended frame format. The reason for the introduction of the extended frame format was to adapt CAN to the requirements of the American car-manufacturers on class C networks ( $\geq 100$  Kbit/s). With the 29 bit identifier a message strategy can be realized, which is similar to the J1850-protocol published by the American Society of Automotive Engineers (SAE) as a recommended practice for class A and B networks ( $< 100$  Kbit/s). Consequently the SAE-J1939 committee (bus & trucks) voted for CAN as network for busses, trucks, agriculture and construction vehicles (class C, 250 Kbit/s). J1939 has made use of the advantage, that the extended frame format with its huge addressing space eases the standardization of the message identifiers.

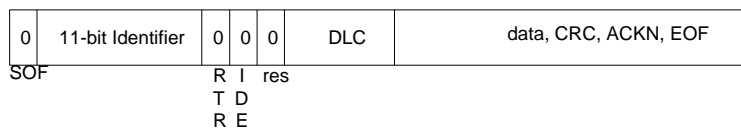
Nevertheless the majority of all CAN applications will continue to work with standard frames only due to some reasons presented in the following chapters.

### 3. Standard and Extended Frame Format

(a) Versions 1.0, 1.1, 1.2, 2.0A



(b) Version 2.0B (Standard Frame Format)



(c) Version 2.0B (Extended Frame Format)

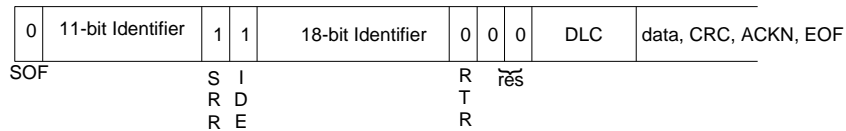


Fig. 1 CAN Data Frame Types

Fig. 1 gives an overview of the different CAN data frame types: All CAN messages start with the identifier (arbitration) field. There are one or three control bits coming along with the identifier. These bits define,

whether it is a standard or an extended frame and whether it is a data or a remote frame. Fig. 1 (a) shows the data frame format, how it is specified in the CAN protocol specification versions 1.0, 1.1, 1.2 and 2.0 A. Fully compatible to that is the standard data format (Fig. 1 (b)), how it is specified in version 2.0 B. In contrast to that Fig. 1 (c) describes the extended data format with the 11+18=29 bit identifier (version 2.0 B). The meaning of the three control bits is as follows:

**RTR bit - Remote Transmit Request**

The RTR bit differentiates between data and remote frames. In data frames this bit is dominant ('0'), in remote frames this bit is recessive ('1').

**SRR bit - Substitute Remote Request**

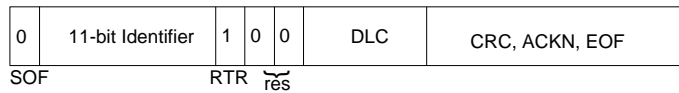
The SRR bit is a recessive bit. It is transmitted in extended frames at the position of the RTR bit of standard frames.

**IDE bit - Identifier Extension**

The IDE bit differentiates between standard and extended frames. In standard frames this bit is dominant, whereas in extended frames this bit is recessive.

Similar to Fig. 1 showing the standard and extended frame formats of CAN data messages, Fig. 2 shows the frame formats of CAN remote messages. Here the RTR bit is set to recessive ('1').

(a) Versions 1.0, 1.1, 1.2, 2.0A



(b) Version 2.0B (Standard Frame Format)



(c) Version 2.0B (Extended Frame Format)

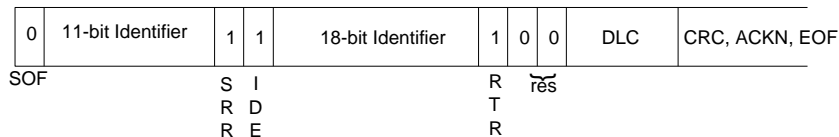


Fig. 2 CAN Remote Frame Types

In case that several nodes within one system start a simultaneous transmission of message frames with the same identifier, the following rules are valid: Data frames have a higher priority than remote frames, and standard frames have a higher priority than extended frames. That means that e.g. a standard remote frame wins arbitration against an extended data frame, if the 11 most significant bits of the identifiers are equal.

## 4. Comparison

### Number of message identifiers

Since the identifier of the standard frame is 11 bits wide, 2048 different message types are possible. Due to implementation reasons only 2032 of them can be used. The 29 bit identifier of the extended frame however supports more than 500 million different message types.

### Bus access time

In worst case a message gets bus access after the completion of the preceding message, if its priority is high enough. When using a transfer rate of 1 Mbit/s, the maximum bus access time at standard frames is 110/134  $\mu$ s, whereas the max. bus access time at extended frames is 130/159  $\mu$ s (without/with stuff bits). In Table 2 the formulas are given how to calculate the number of bit times of CAN message frames.

Condition	Number of bits per message	
	Standard Frame Format	Extended Frame Format
No Stuff Bits	$47 + 8 \cdot \text{DLC} *$	$67 + 8 \cdot \text{DLC} *$
Max. number of Stuff Bits	$55 + 10 \cdot \text{DLC} *$	$80 + 10 \cdot \text{DLC} *$

\*) DLC: Data Length Code

Tab. 2 CAN Message Length (incl. 3 Bits for Intermission)

### Bus throughput

The maximum data rate of CAN is specified at 1 Mbit/s. This results in a max. duration of a standard frame of 111  $\mu$ s and a max. duration of an extended frame of 131  $\mu$ s (without stuff bits). If these message are transmitted cyclicly, max. net transfer rates of 577 kbit/s (standard frame) and 488 kbit/s (extended frame) can be reached. Fig. 3 shows a diagram, where the max. net transfer rates when using standard and extended frames are shown as a function of the DLC (Data Length Code, that is the number of transmitted data bytes per message).

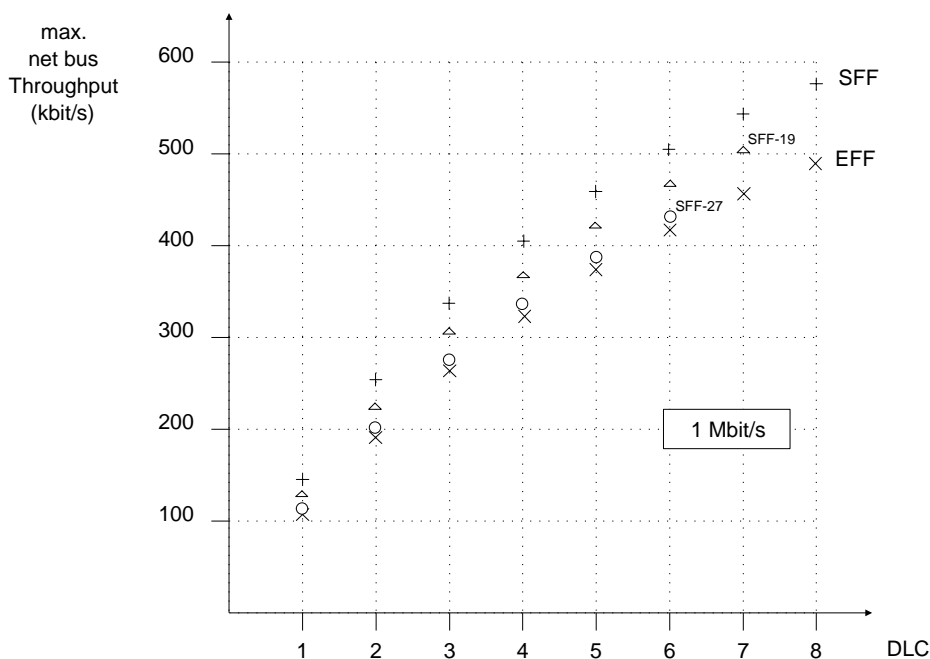


Fig. 3 Net Bus Throughput of Standard (SFF) and Extended Frames (EFF) at a Data Rate of 1 Mbit/s

The reason why extended frames have been introduced is, that a much bigger number of different identifiers are at the user's disposal. An alternative approach is, to use standard frames and to claim one or two data bytes in order to extend the identifier. That is as far possible as not all of the 8 data bytes are used for the application. At arbitration however only the 11 bit identifier is regarded. In Fig. 3 the max. bus throughput of messages, at which one (SFF-19) or two (SFF-27) data bytes are used for the purpose to increase the identifier, are sketched additionally.

#### CPU-load caused by CAN

All Controller Area Networks contain at least one node, which is microcontroller driven. Since at CAN the OSI-layers 1 and 2 are realized almost completely by hardware, the microcontroller is only responsible to provide messages for transmission, to process messages being received, and to handle parts of the network management (e.g. initialization). Comparing now standard and extended frames, the length of the identifier has only a very small impact to the CPU-load, if the following conditions are fulfilled:

- ° optimal distribution of the message identifiers within the system and
- ° optimal programming of the acceptance filters.

All CAN controllers available today have any kind of acceptance filter, which has the function to reject messages not being interesting for the certain node. Thus the CPU-load at reception can be reduced considerably. For the next CAN controller generation, which will be able to handle also extended frames, enhanced acceptance filters are announced. These filters will compensate a possibly higher CPU-load at the reception of extended frames by additional features. These features are a better programming flexibility



and a higher number of acceptance mask and code entries. In addition increased memory buffers will reduce the CPU peak burden caused by message bursts.

Summarizing it can be said that at optimal use of the acceptance filter there is hardly any difference between standard and extended frames concerning the CPU-load. For more details please refer to /2/.

## 5. Compatibility Aspects

As a matter of principle, standard and extended frames coexist in the same network. Error handling and fault confinement work in an uniform way for both formats. As mentioned above the arbitration priorities of standard and extended identifiers are well defined.

All CAN controllers support the standard frame format. Thus new CAN controllers, which support the extended frame format, can communicate with today's CAN controllers, when they use standard frames.

Concerning the compatibility of standard and extended frames three types of CAN controllers exist:

- CAN controllers according to CAN protocol specification vers. 2.0 A, which are able to receive and transmit standard frames
- CAN controllers according to CAN protocol specification vers. 2.0 B, which are able to receive and transmit standard frames **and** to tolerate extended frames without detecting an error ('**extended frame passive**')
- CAN controllers according to CAN protocol specification vers. 2.0 B, which are able to receive and transmit standard **and** extended frames ('**extended frame active**').

In Tab. 3 a standard/extended frame compatibility matrix is shown. An error is only detected, if a CAN controller, which does not tolerate extended frames, meets an extended frame at its bus terminals.

Receiver Transmitter	Spec. 2.0 Part A Standard Frame	Spec 2.0 Part B "Extended Frame <u>Passive</u> "	Spec. 2.0 Part B "Extended Frame <u>Active</u> "
Spec. 2.0 Part A Standard Frame	Compatible	Compatible	Compatible
Spec. 2.0 Part B Standard Frame	Compatible	Compatible	Compatible
Spec. 2.0 Part B Extended Frame	not Compatible	Compatible	Compatible

Tab. 3 Standard/Extended Frame Compatibility Matrix

## 6. Available Products

If standard frames are used exclusively in an application, then both kinds of CAN controllers - those according to version 2.0 B ("passive" or "active") as well as those according to version 2.0 A (or even older versions) - can be used. That means that for these CAN networks the full range of available CAN controllers can be used. All future CAN products will still perform standard frame communication.

When extended frames are used in a CAN network, the number of usable products is only an extract of the full range of available CAN controllers. Because of the aim to offer very cheap CAN controllers, it is likely that even some future CAN controllers will be created which do not support extended frame "actively". For most applications the cheaper price will be more beneficial than the additional feature of extended frame communication.

## 7. Conclusion

Tab. 4 tries a summarizing valuation of the standard and extended frame formats regarding the number of different identifiers, the bus access time, the bus throughput, the CPU-load, the availability of products and the chip size/cost. The result is, that it is advantageous to use the standard frame format as long as the application allows to do so. From today's point of view only the american automotive manufacturers have applications needing extended frames. Therefore it should be recommended for all other applications to use only standard frames.

	Standard Frame Format (SFF)	Extended Frame Format (EFF)
Number of IDs	o	+
Bus Access Time	+	o
Bus Throughput	+	o
CPU - Load	+	+
Availability of Products	+	-
Chip size / Cost	+	o

+ better than average  
o average  
- below average

Tab.4 Comparison of CAN Standard and Extended Frame Products

## References

- /1/ CAN Specification Version 2.0, Philips Semiconductors, 1991
- /2/ Application of the P8xC592 Microcontroller with CAN-Interface, Application Note HKI/AN 91014, Philips Semiconductors Hamburg, 1992