INTERNATIONAL STANDARD

ISO 11519-2

First edition
1994-06-15

AMENDMENT 1
1995-04-01

Road vehicles — Low-speed serial data communication —

Part 2:
Low-speed controller area network (CAN)
AMENDMENT 1

Véhicules routiers — Communication en série de données à vitesse basse —
Partie 2: Gestionnaire de réseau de communication à faible vitesse (CAN)

AMENDEMENT 1

Reference number
ISO 11519-2:1994/Amd.1:1995(E)

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Amendment 1 to International Standard ISO 11519-2:1994 was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 3, *Electrical and electronic equipment*.

ISO 11519 consists of the following parts, under the general title *Road vehicles — Low-speed serial data communication* :

— *Part 1: General and definitions*

— *Part 2: Low-speed controller area network (CAN)*

— *Part 3: Vehicle area network (VAN)*

International Organization for Standardization
Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Road vehicles — Low-speed serial data communication —

Part 2:

Low-speed controller area network (CAN)

AMENDMENT 1

Page iv

Insert new page v and the following Introduction.

Introduction

The acceptance and introduction of serial data communication to more and more applications has led to requirements that the assignment of message identifiers to communication functions be standardized for certain applications. These applications can be realized with CAN more comfortably, if the address range that has been defined in ISO 11519-1 by 11 identifier bits is enlarged.

Therefore a second message format ("extended format") is introduced that provides a larger address range defined by 29 bits. This will relieve the system designer from compromises with respect to defining well-structured naming schemes. Users of CAN who do not need the identifier range offered by the extended format can rely on the conventional 11 bit identifier range ("standard format") further on.

In order to distinguish standard and extended format the first reserved bit of the CAN message format, as it is defined in ISO 11519-1, is used. This is done in such a way that the message format in ISO 11519-1 is equivalent to the standard format and therefore is still valid. Furthermore, the extended format has been defined so that messages in standard format and extended format can coexist within the same network.

Amendment 1 to ISO 11519-2 details the necessary changes to the 1994 Standard to include both formats.

Page 10

Replace the wording below item b.2 of 6.1 (Service of the LLC sublayer) by the following.

According to the two different LLC services there are two types of frames from or to the user:

— LLC Data Frame,

— LLC Remote Frame.

There are two formats for both Data Frames and Remote Frames which differ in the length of the identifier (see 6.2): Frames with an 11 bit IDENTIFIER are denoted STANDARD LLC Frames, and Frames containing a 29 bit IDENTIFIER are denoted Extended LLC Frames.

In 6.2, replace table 3 by the following.

Table 3 — List of LLC service primitive parameters

LLC Service Primitive Parameters	
IDE	identifies the IDENTIFIER's length
IDENTIFIER	idendifies the data and its priority
DLC	Data Length Code
DATA	data the user wants to transmit
TRANSFER_STATUS	confirmation parameter

Replace item b) of 6.2.1 L_DATA.request, by the following.

b) Semantics of the L_DATA.request primitive

The primitive shall provide parameters as follows.

```
L_DATA.request (
                IDE
                IDENTIFIER
                DLC
                DATA
                )
```


The parameter DATA is insignificant if the associated LLC Data Frame is of data length zero.

Page 11

Replace item b) of 6.2.2 L_DATA.indication, by the following.

b) Semantics of the L_DATA.indication primitive

The primitive shall provide parameters as follows.

```
L_DATA.indication (
                   IDE
                   IDENTIFIER
                   DLC
                   DATA
                   )
```

The parameter DATA is insignificant if the associated LLC data frame is of data length zero.

Page 12

Replace item b) of 6.2.3 L_DATA.confirm, by the following.

b) Semantics of the L_DATA.confirm primitive

The primitive shall provide parameters as follows.

```
L_DATA.confirm (
                IDE
                IDENTIFIER
                TRANSFER_STATUS
                )
```

The TRANSFER_STATUS is used to indicate the completion of the transaction initiated by the previous L_DATA.request primitive.

```
TRANSFER_STATUS:[COMPLETE,NOT_COMPLETE]
```

Replace item b) of 6.2.4 L_REMOTE.request, by the following.

b) Semantics of the L_REMOTE.request primitive

The primitive shall provide parameters as follows.

```
L_REMOTE.request(
                 IDE
                 IDENTIFIER
                 DLC
                 )
```

The value of DLC equals the length of the data field of the requested data frame.

Page 13

Replace item a) of 6.2.5 L_REMOTE.indication, by the following.

a) Function

The L_REMOTE.indication primitive is passed from the LLC sublayer to the LLC user to indicate the arrival of a request for transmission of a LSDU.

Replace item b) of 6.2.5 L_REMOTE.indication, by the following.

b) Semantics of the L_REMOTE.indication primitive

The primitive shall provide parameters as follows.

```
L_REMOTE.indication (
                     IDE
                     IDENTIFIER
                     DLC
                     )
```

The IDENTIFIER identifies the LSDU to be sent. The value of DLC equals the length of the data field of the requested data frame.

Replace item b) of 6.2.6 L_REMOTE.confirm, by the following.

b) Semantics of the L_REMOTE.confirm primitive

The primitive shall provide parameters as follows.

```
L_REMOTE.confirm (
                  IDE
                  IDENTIFIER
                  TRANSFER_STATUS
                  )
```

The TRANSFER_STATUS is used to indicate the completion of the transaction initiated by the previous L_REMOTE.request primitive.

```
TRANSFER_STATUS:[COMPLETE,NOT_COMPLETE]
```

Replace 6.3 Structure of LLC frames, by the following.

LLC frames are the protocol data units that are exchanged between peer LLC entities (LPDUs). The structure and format of the LLC Data and Remote Frame are specified subsequently.

Page 14

Replace 6.3.1 Specification of the LLC data frame, by the following.

A LLC data frame is composed of four bit fields (see figure 4):

— IDE bit,

— Identifier Field,

— Data Length Code (DLC) Field,

— LLC Data Field.

IDE bit

Standard LLC Data Frames and Extended LLC Data Frames are distinguished by the IDE bit: IDE = '0' indicates the Standard Frame Format, and IDE = '1' indicates the Extended Frame Format.

Identifier Field

Depending on the IDE bit, the identifier is either 11 bit long (Standard Frame Format), or 29 bit long (Extended Frame Format). In case of the Standard Frame Format the 7 most significant bits (ID–10 to ID–4) must not be all '1'.

DLC field

The number of bytes in the data field is indicated by the Data Length Code. This Data Length Code consists of 4 bits. The data field can be of length zero. The admissible number of data bytes for a data frame ranges from 0, ..., 8. Other values may not be used.

Replace figure 4 by the following.

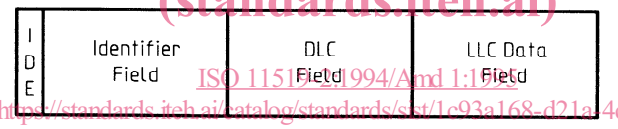

Figure 4 — LLC Data Frame

Page 15

Replace 6.3.2 Specification of the LLC Remote Frame, by the following.

A LLC remote frame is composed of three bit fields:

— IDE bit,

— Identifier field,

— DLC field,

The structures of the bit fields IDE, Identifier Field, and DLC Field are equivalent to the structures of the corresponding bit fields of a LLC Data Frame (see 6.2.1). The only difference between a LLC Remote Frame and a LLC Data Frame is that there is no Data Field in a Remote Frame, independent of the value of the Data Length Code. This value is the Data Length Code of the corresponding Data Frame.

Replace figure 5 by the following.

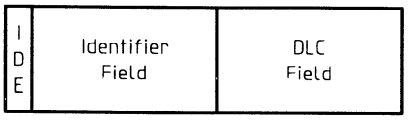

Figure 5 — LLC remote frame

Page 16

Add at the end of 8.1 Services of the MAC sublayer, the following new wording.

According to the two different MAC data transfer services there are two types of frames from or to the user:

— MAC Data Frame,

— MAC Remote Frame.

There are two different formats for both Data Frames and Remote Frames which differ in the length of the IDENTIFIER:

Frames with an 11 bit IDENTIFIER are denoted Standard MAC Frames, and

Frames containing 29 bit IDENTIFIER are denoted Extended MAC Frames.

Page 17

Amend items b) and c) of 8.1.1.1 MA_DATA.request, to read as follows.

b) Semantics of the MA_DATA.request primitive

The primitive shall provide parameters as follows.

```
MA_DATA.request (
                 IDE
                 IDENTIFIER
                 DLC
                 DATA
                 )
```


The parameter DATA is insignificant for MAC data frames of data length zero.

c) Effect on receipt

Receipt of this primitive causes the MAC sublayer to prepare a Protocol Data Unit by adding all MAC specific control information (SOF, RTR bit, reserved bit r0, CRC, 'recessive' bit during ACK Slot, EOF in case of MAC Standard Frame Format, and SOF, SRR bit, RTR bit, reserved bit r1, CRC, 'recessive' bit during ACK Slot, EOF in case of MAC Extended Frame Format) to the MSDU coming from the LLC sublayer. The MAC PDU will be serialized and passed bit by bit as a service data unit to the physical layer for transfer to the peer MAC sublayer entity or entities.

Amend item b) of 8.1.1.2 MA_DATA.indication, to read as follows.

b) Semantics of the MA_DATA.indication primitive

The primitive shall provide parameters as follows.

```
MA_DATA.indication (
                    IDE
                    IDENTIFIER
                    DLC
                    DATA
                    )
```

The parameter DATA is insignificant if the associated MAC data frame is of data length zero. The arrival of a MSDU is indicated to the LLC sublayer only if it has been received correctly.

Amend item b) of 8.1.1.3 MA_DATA.confirm, to read as follows.

b) Semantics of the MA_DATA.confirm primitive

The primitive shall provide parameters as follows.

```
MA_DATA.confirm(
                IDE
                IDENTIFIER
                TRANSMISSION_STATUS
                )
```

The TRANSMISSION_STATUS is used to indicate the success or failure of the previous MA_DATA.request primitive.

```
TRANSMISSION_STATUS:[SUCCESS,NO_SUCCESS]
```

Failures are either errors which occurred during transmission or the loss of arbitration.

Page 18

Amend items b) and c) of 8.1.1.4 MA_REMOTE.request, to read as follows.

b) Semantics of the MA_REMOTE.request primitive

The primitive shall provide parameters as follows.

```
MA_REMOTE.request (
                  IDE
                  IDENTIFIER
                  DLC
                  )
```

The Identifier identifies the MSDU to be sent. The value of DLC equals the length of the data of the requested MSDU.

c) Effect on receipt

Receipt of this primitive causes the MAC sublayer to prepare a Protocol Data Unit by adding all MAC specific control information (SOF, RTR bit, reserved bit r0, CRC, recessive bit during ACK Slot, EOF in case of MAC Standard Frame Format, and SOF, SRR bit, RTR bit, reserved bits r1 and r0, CRC, recessive' bit during ACK Slot, EOF in case of MAC Extended Frame Format). The MAC PDU will be serialized and passed bit by bit as service data unit to the physical layer for transfer to the peer MAC sublayer entity or entities.

Page 19

Amend item b) of 8.1.1.5 MA_REMOTE.indication, to read as follows.

b) Semantics of the MA_REMOTE.indication primitive

The primitive shall provide parameters as follows.

```
MA_REMOTE.indication (
                     IDE
                     IDENTIFIER
                     DLC
                     )
```

The arrival of a MSDU transmission request is indicated to the LLC sublayer only if it has been received correctly.

Amend item b) of 8.1.1.6 MA_REMOTE.confirm, to read as follows.

b) Semantics of the MA_REMOTE.confirm primitive

The primitive shall provide parameters as follows.

```
MA_REMOTE.confirm (
                  IDE
                  IDENTIFIER
                  TRANSMISSION_STATUS
                  )
```

The TRANSMISSION_STATUS is used to indicate the success or failure of the previous MA_REMOTE.request primitive.

```
TRANSMISSION_STATUS:[SUCCESS,NO_SUCCESS]
```

Failures are either errors which occurred during transmission or loss of the arbitration.

Page 21

Change the function 3 of the Frame transmission, Transmit data encapsulation in 8.2 Functional model of the MAC sublayer architecture, to read as follows.

Transmit data encapsulation

1) Acceptance of LLC frames and interface control information.

2) CRC sequence calculation.

3a) Standard Frame Format: Construction of a Standard MAC Frame by adding SOF, RTR bit, reserve bit r0 CRC, ACK, and EOF to the LLC frame.

3b) Extended Frame Format: Construction of an Extended MAC Frame by adding SOF, SRR bit, RTR bit, reserve bits r0 and r1, CRC, ACK, and EOF to the LLC frame.

Page 22

Add to the functions of 8.2.2 Frame reception, a) Receive media access management in 8.2 Functional model of the MAC sublayer architecture:

9) Distinction between Standard and Extended Frame Format.

Amend 8.3.1 Specification of the MAC data frame, to read as follows.

A MAC data frame is composed of seven different bit fields:

— Start of Frame (SOF),
— Arbitration field,
— Control field,
— Data field,

Page 23

— CRC field,
— ACK field,
— End of Frame (EOF).

a) *Start of Frame (SOF)*

marks the beginning of data and remote frames. It consists of a single 'dominant' bit.

A node is only allowed to start transmission when the bus is idle (see bus idle in 8.3.5). All nodes have to synchronize to the leading edge caused by Start of Frame of the node starting transmission first.

Replace sub-clause "*Arbitration field*" and the first paragraph of "*Control field*" to read as follows.

b) *Arbitration field*

The format of the Arbitration field is different for Standard Format and Extended Format Frames, (see figures 7a and 7b):

— In Standard Format the Arbitration Field consists of the 11 bit IDENTIFIER, passed from the LLC sublayer, and the RTR (Remote Transmission Request) bit. The value of the RTR bit in a MAC Data Frame is '0'.