INTERNATIONAL STANDARD

ISO 11898

First edition
1993-11-15

Road vehicles — Interchange of digital information — Controller area network (CAN) for high-speed communication

Véhicules routiers — Échange d'information numérique — Gestionnaire de réseau de communication à vitesse élevée (CAN)

Reference number
ISO 11898:1993(E)

Contents

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

Printed in Switzerland

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.

International Standard ISO 11898 was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Sub-Committee SC 3, *Electrical and electronic equipment*.

Road vehicles — Interchange of digital information — Controller area network (CAN) for high-speed communication

1 Scope

This International Standard specifies characteristics of setting up an interchange of digital information between Electronic Control Units (ECUs) of road vehicles equipped with the Controller Area Network at transmission rates above 125 kbit/s up to 1 Mbit/s.

The Controller Area Network (CAN) is a serial communication protocol which supports distributed real-time control and multiplexing.

This specification of CAN describes the general architecture of CAN in terms of hierarchical layers according to the ISO reference model for Open Systems Interconnection (OSI) specified in ISO 7498. The data link layer and physical layer are specified according to ISO 8802-2 and ISO 8802-3. This International Standard contains detailed specifications of aspects of CAN belonging to the

a) physical layer;

b) data link layer

— Logical Link Control (LLC) sublayer,

— Medium Access Control (MAC) sublayer.

All other layers of the OSI model do not have counterparts within this specification of CAN protocol but are part of the user's level.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 11898. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO 11898 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 7498:1984, *Information processing systems — Open Systems Interconnection — Basic Reference Model.*

ISO 7637-3:—[1], *Road vehicles — Electrical disturbance by conduction and coupling — Part 3: Passenger cars and light commercial vehicles with nominal 12 V supply voltage and commercial vehicles with 24 V supply voltage — Electrical transient transmission by capacitive and inductive coupling via lines other than supply lines.*

ISO 8802-2:1989, *Information processing systems — Local area networks — Part 2: Logical link control.*

1) To be published.

ISO/IEC 8802-3:1993, *Information technology — Local and metropolitan area networks — Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.*

3 Definitions and abbreviations

For the purposes of this International Standard, the following definitions apply.

3.1 Data link layer definitions

3.1.1 bit rate: Number of bits per time during transmission, independent of bit representation.

3.1.2 bit stuffing: Technique used in bit-oriented protocols in order

— to achieve data transparency (arbitrary bit patterns may not be interpreted as protocol information), and

— to provide "dominant" to "recessive" edges, and vice versa, which are necessary for correct resynchronization when using a Non-Return-to-Zero bit representation.

Whenever the transmitting logic encounters a certain number (stuff width) of consecutive bits of equal value in the data, it automatically stuffs a bit of complementary value — a stuff bit — into the outgoing bit stream. Receivers destuff the frame, i.e. the inverse procedure is carried out.

3.1.3 bus: Topology of a communication network, where all nodes are reached by passive links which allow transmission in both directions.

3.1.4 bus value: One of two complementary logical values: "dominant" or "recessive". The "dominant" value represents the logical "0", and the "recessive" represents the logical "1". During simultaneous transmission of "dominant" and "recessive" bits, the resulting bus value will be "dominant".

3.1.5 contention-based arbitration: Carrier Sense Multiple Access (CSMA) arbitration procedure where simultaneous access of multiple nodes results in a contention. One frame will survive the contention uncorrupted.

3.1.6 frame: Data link protocol data unit specifying the arrangement and meaning of bits or bit fields in the sequence of transfer across the transmission medium.

3.1.7 multicast: Addressing where a single frame is addressed to a group of nodes simultaneously. Broadcast is a special case of multicast, whereby a single frame is addressed to all nodes simultaneously.

3.1.8 multi-master: System partitioned into several nodes where every node may temporarily control the action of other nodes.

3.1.9 node: Any assembly, linked to a communication line, capable of communicating across the network according to a communication protocol specification.

3.1.10 non-return-to-zero: Method of representing binary signals. Within one and the same bit time, the signal level does not change, i.e. a stream of bits having the same logical value provides no edges.

3.1.11 priority: Attribute to a frame controlling its ranking during arbitration. A high priority increases the probability that a frame wins the arbitration process.

3.1.12 protocol: Formal set of conventions or rules for the exchange of information between nodes, including the specification of frame administration, frame transfer and physical layer.

3.1.13 receiver: Device that converts physical signals used for transmission back into logical information or data signals.

3.1.14 transmitter: Device that converts information or data signals to electrical or optical signals so that these signals can be transferred across the communication medium.

3.2 Physical layer definitions

3.2.1 common mode bus voltage range: Boundary voltage levels of V_{CAN_L} and V_{CAN_H}, for which proper operation is guaranteed if up to the maximum number of ECUs are connected to the bus line.

3.2.2 differential internal capacitance, C_{diff} (of an ECU): Capacitance seen between CAN_L and CAN_H during the recessive state when the ECU is disconnected from the bus line. (See figure 1.)

3.2.3 differential internal resistance, R_{diff} (of an ECU): Resistance which is seen between CAN_L and CAN_H during the recessive state when the ECU is disconnected from the bus line. (See figure 1.)

3.2.4 differential voltage, V_{diff}: value

$$V_{diff} = V_{CAN\_H} - V_{CAN\_L}$$

with the voltages V_{CAN_L} and V_{CAN_H} denoting the voltages of CAN_L and CAN_H relative to ground of each individual ECU.

3.2.5 internal capacitance, C_{in} (of an ECU): Capacitance seen between CAN_L (or CAN_H) and ground during the recessive state when the ECU is disconnected from the bus line. (See figure 1.)

3.2.6 internal delay time, t_{ECU} (of an ECU): Sum of all asynchronous delay times occurring on the transmitting and receiving path relative to the bit timing logic unit of the protocol IC of each individual ECU disconnected from the bus line.

3.2.7 internal resistance, R_{in} (of an ECU): Resistance which is seen between CAN_L (or CAN_H) and ground during the recessive state when the ECU is disconnected from the bus line. (See figure 1.)

3.2.8 physical layer: Electrical circuit realization that connects an ECU to a bus. The total number of ECUs connected on a bus is limited by electrical loads on the bus line.

3.2.9 physical media (of the bus): Pair of parallel wires, shielded or unshielded, dependent on EMC requirements. The individual wires are designated as CAN_L and CAN_H. The names of the corresponding pins of ECUs are also denoted by CAN_L and CAN_H respectively.

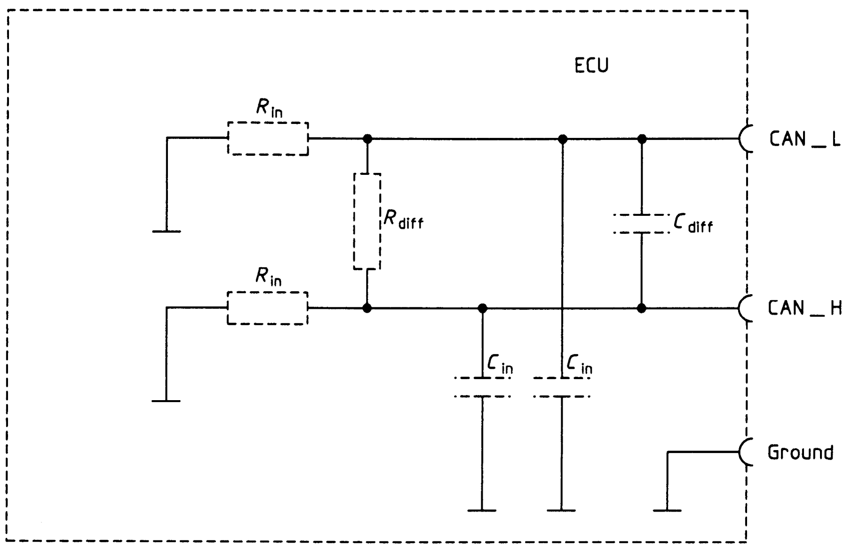

Figure 1 — Definitions of internal capacitances and internal resistances of ECU

3.3 List of abbreviations

ACK	Acknowledgement
ECU	Electronic Control Unit
EOF	End of Frame
CAN	Controller Area Network
CRC	Cyclic Redundancy Check
DLC	Data Length Code
IC	Integrated Circuit
FCE	Fault Confinement Entity
LAN	Local Area Network
LLC	Logical Link Control
LME	Layer Management Entity
LPDU	LLC Protocol Data Unit
LSB	Least Significant Bit
LSDU	LLC Service Data Unit
MAC	Medium Access Control
MAU	Medium Access Unit
MDI	Medium Dependent Interface
MPDU	MAC Protocol Data Unit
MSB	Most Significant Bit
MSDU	MAC Service Data Unit
NRZ	Non-Return-to-Zero
OSI	Open System Interconnection
PL	Physical Layer
PLS	Physical Signalling
PMA	Physical Medium Attachment
RTR	Remote Transmission Request
SOF	Start of Frame

4 Basic concepts of CAN

CAN has the following properties:

— multi-master priority-based bus access;

— non-destructive contention-based arbitration;

— multicast frame transfer by acceptance filtering;

— remote data request;

— configuration flexibility;

— system-wide data consistency;

— error detection and error signalling;

— automatic retransmission of frames that have lost arbitration or have been destroyed by errors during transmission;

— distinction between temporary errors and permanent failures of nodes and autonomous switching-off of defective nodes.

4.1 Frames

Information on the bus is sent in fixed format frames of different but limited length. When the bus is idle, any connected node may start to transmit a new frame.

4.2 Bus access method

When the bus is idle, any node may start to transmit a frame. If two or more nodes start to transmit frames at the same time, the bus access conflict is resolved by contention-based arbitration using the identifier. The mechanism of arbitration guarantees that neither information nor time is lost. The transmitter with the frame of highest priority gains bus access.

4.3 Information routing

In CAN systems a node does not make use of any information about the system configuration (e.g. node address). Instead, receivers accept or do not accept information based upon a process called "Frame Acceptance Filtering", which decides whether the received information is relevant or not. There is no need for receivers to know the transmitter of the information and vice versa.

4.4 System flexibility

Nodes may be added to the CAN network without requiring any change in the software or hardware of any node, if the added node is not the transmitter of any data frame or if the added node does not require any additional transmitted data.

4.5 Data consistency

Within a CAN network it is guaranteed that a frame is simultaneously accepted either by all nodes or by no node. Thus data consistency is a property of the system achieved by the concepts of multicast and by error handling.

4.6 Remote data request

By sending a remote frame, a node requiring data may request another node to send the corresponding data frame. The data frame and the corresponding remote frame are named by the same identifier.

4.7 Error detection

For dectecting errors, the following measures are provided:

— monitoring (transmitters compare the bit levels to be transmitted with the bit levels detected on the bus);

— 15-bit cyclic redundancy check;

— variable bit stuffing with a stuff width of 5;

— frame check.

4.8 Error signalling and recovery time

Corrupted frames are flagged by any transmitting node and any normallly operating (error-active) receiving node. Such frames are aborted and will be retransmitted according to the implemented recovery procedure (see 6.3.3). The recovery time from detecting an error until the possible start of the next frame is typically 17 to 23 bit times (in the case of a heavily disturbed bus, up to 29 bit times), if there are no further errors.

4.9 Acknowledgement

All receivers check the consistency of the received frame and will acknowledge a consistent frame and flag an inconsistent frame.

4.10 Automatic retransmission

Frames that have lost arbitration and frames that have been disturbed by errors during transmission will be retransmitted automatically when the bus is idle again. A frame that will be retransmitted is handled like any other frame. This means that it participates in the arbitration process in order to gain bus access.

4.11 Fault confinement

CAN nodes are able to distinguish short disturbances from permanent failures. Defective transmitting nodes are switched off. "Switched off" means that a node is logically disconnected from the bus line, so that it can neither send nor receive any frames.

4.12 "error-active"

An "error-active" node can normally take part in bus communication and send an active error flag when an error has been detected. The active error flag consists of six (6) dominant consecutive bits and violates the rule of bit stuffing and all fixed formats appearing in a regular frame (see 11.1.5).

4.13 "error-passive"

An "error-passive" node shall not send an active error flag. It takes part in bus communication, but when an error has been detected a passive error flag is sent. The passive error flag consists of six (6) recessive consecutive bits. After transmission, an "error-passive" node will wait some additional time before initiating a further transmission (see suspend transmission in 8.4.5 and 11.1.5).

4.14 "bus off"

A node is in the state "bus off" when it is switched off from the bus due to a request of fault confinement entity. In the "bus off" state, a node can neither send nor receive any frames. A node can leave the "bus off" state only upon a user request.

5 Layered architecture of CAN

5.1 Reference to OSI model

According to the OSI reference model, the CAN architecture represents two layers:

— data link layer,

— physical layer.

This International Standard specifies the data link and the physical layer of CAN (see figure 2).

According to ISO 8802-2 and ISO 8802-3 (LAN standards), the data link layer is subdivided into:

— Logic Link Control (LLC);

— Medium Access Control (MAC).

The physical layer is subdivided into:

— Physical Signalling (PLS);

— Physical Medium Attachment (PMA);

— Medium Dependent Interface (MDI).

The MAC sublayer operations are supervised by a management entity called the "Fault Confinement Entity (FCE)". Fault confinement is a self-checking mechanism that makes it possible to distinguish short disturbances from permanent failures (fault confinement: see 11.1).

The physical layer may be supervised by an entity that detects and manages failures of the physical medium (for example, shorted or interrupted bus lines, bus failure management: see 11.2).

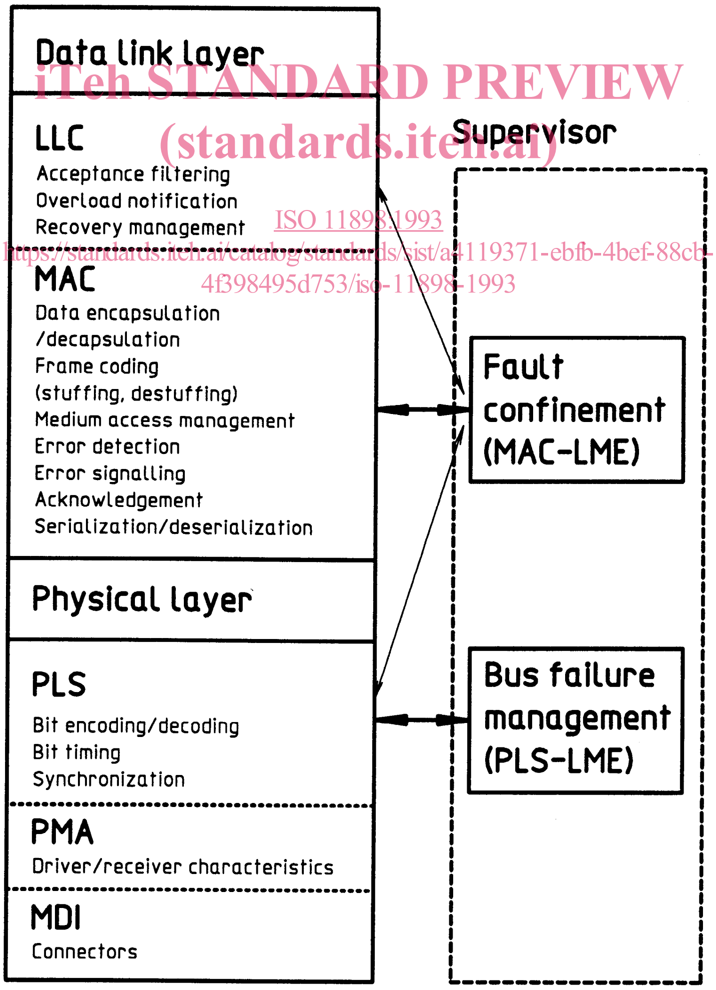

Figure 2 — Layered architecture of CAN

5.2 Protocol specification

Two peer protocol entities communicate with each other by exchanging frame or Protocol Data Units (PDUs). An (N)-layer Protocol Data Unit (NPDU) consists of N-layer specific Protocol Control Information (N-PCI) and (N)-user data. In order to transfer a NPDU it must be passed to a (N-1)-layer entity via a (N-1)-Service Access Point [(N-1)-SAP]. The NPDU is passed by means of the (N-1)-layer Service Data Unit [(N-1)-SDU] to the (N-1)-layer, the services of which allow the transfer of the NPDU. The service data unit is the interface data whose identity is preserved between (N)-layer entities, i.e. it represents the logical data unit transferred by a service. The data link layer of the CAN protocol does not provide means for mapping one SDU into multiple PDUs nor means for mapping multiple SDUs into one PDU, i.e. a NPDU is directly constructed from the associated NSDU and the layer specific control information N-PCI. Figure 3 illustrates the data link sublayer interactions.

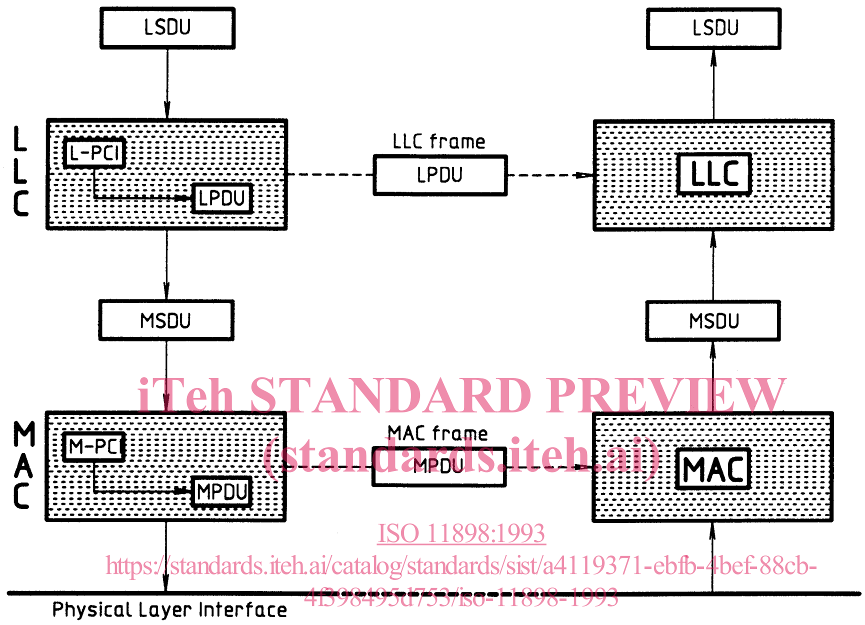

Figure 3 — Protocol layer interactions

5.3 Format description of services

5.3.1 Format description of service primitives

Service primitives are written in the form:

```
service.type(
            [parameter1,...]
            )
```

"service" indicates the name of the service, e.g. L_DATA for data transfer service provided by the LLC sublayer.

"type" indicates the type of the service primitives (see 5.3.2).

"[parameter1,...]" is the list of values passed to the service primitives.

The brackets indicate that this parameter list may be empty.

5.3.2 Types of service primitives

Service primitives are of three generic types:

service. request

The request primitive is passed from the (N)-user (service user) to the (N)-layer (service provider) to request initiation of the service.

service. indication

The indication primitive is passed from the (N)-layer to the (N)-user to indicate an internal (N)-layer (or sublayer) event which is significant to the (N)-user. This event may be logically related to a remote service request, or may be caused by an event internal to the (N)-layer (or sublayer).

service. confirm

The confirm primitive is passed from the (N)-layer (or sublayer) to the (N)-user to convey the results of one or more associated previous service request(s). This primitive may indicate either failure to comply or some level of compliance. It does not necessarily indicate any activity at the remote peer interface.

5.4 LLC interface

The LLC sublayer offers two types of connectionless transmission services to the LLC user:

— unacknowledged data transfer service,

— unacknowledged remote data request service.

The interface service data from or to the user is described in 6.1.2. The messages that can be sent between LLC user and LLC sublayer are shown in table 1 a) and b).

Table 1 — Messages between LLC user and LLC sublayer

a) Message sent from LLC user to LLC sublayer	
User to LLC message	**Meaning**
Reset_Request	Request to set the node into an initial state
b) Messages sent from LLC sublayer to LLC user	
LLC to user message	**Meaning**
Reset_Response	Response to the Reset_Request
Node_Status	Indicates the current status of the node, i.e. it signals whether or not the node is in the state "bus off".

The LLC interface messages from and to the supervisor fault confinement entity are described in 11.1.3.1.

6 Description of LLC sublayer

The LLC (Logical Link Control) sublayer describes the upper part of the OSI data link layer. It is concerned with those protocol issues that are independent of the type of medium access method.

6.1 Services of LLC sublayer

6.1.1 LLC sublayer

The LLC sublayer offers two types of connectionless-mode transmission services:

Unacknowledged data transfer service

This service provides means by which LLC users can exchange Link Service Data Units (LSDU) without the establishment of a data link connection. The data transfer can be point-to-point, multicast or broadcast.

Unacknowledged remote data request service

This service provides means by which a LLC user can request another remote node to transmit a Link Service Data Unit (LSDU) without the establishment of a data link connection.

The way in which the remote node serves the data request is not specified here. Basically, there are two ways:

a) The requested data is prepared by the remote user for transmission. In this case the data is located in a remote node buffer and will be transmitted by the LLC entity upon reception of the remote request frame.

b) The requested data will be transmitted by the remote user upon reception of the remote request frame.

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

— LLC Data Frame,

— LLC Remote Frame.

The LLC Data Frame carries data from a transmitter to a receiver. The LLC remote frame is transmitted to request the transmission of a data frame (with the same identifier) from a (single) remote node. In both cases, the LLC sublayer notifies the successful transmission or reception of a frame to the user.

6.1.2 Service primitive specification

This subclause describes in detail the LLC service primitives and their associated parameters. The complete list of LLC service primitives is given in table 2.

Table 2 — LLC service primitives overview

Unacknowledged Data Transfer Service	
L_Data.request	Request for data transfer
L_Data.indication	Indication of data transfer
L_Data.confirm	Confirm data transfer
Unacknowledged Remote Data Request Service	
L_Remote.request	Request for remote data request
L_Remote.indication	Indication of remote data request
L_Remote.confirm	Confirmation remote data request

The parameters that are associated with the different LLC service primitives are listed in table 3.

Table 3 — List of LLC service primitive parameters

LLC Service Primitive Parameters	
IDENTIFIER	Identifies the data and its priority
DLC	Data Length Code
DATA	Data the user wants to transmit
TRANSFER_STATUS	Confirmation parameter

6.1.2.1 L_DATA.request

a) **Function**

The L_DATA.request primitive is passed from the LLC user to the LLC sublayer to request that a LSDU be sent to one or more remote LLC entities.

b) **Semantics of L_DATA.request primitive**

The primitive shall provide parameters as follows:


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

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

c) **Effect on receipt**

Receipt of this primitive causes the LLC sublayer to initiate the transfer of a LLC data frame by use of the data transfer service provided by the MAC sublayer (see table 5).

6.1.2.2 L_DATA.indication

a) **Function**

The L_DATA.indication primitive is passed from the LLC sublayer to the LLC user to indicate the arrival of a LSDU.

b) **Semantics of L_DATA.indication primitive**

The primitive shall provide parameters as follows:

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

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