INTERNATIONAL STANDARD

ISO 11898-1

Second edition
2015-12-15

Road vehicles — Controller area network (CAN) —

Part 1: Data link layer and physical signalling

Véhicules routiers — Gestionnaire de réseau de communication (CAN) —

Partie 1: Couche liaison de données et signalisation physique

Reference number
ISO 11898-1:2015(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT), see the following URL: Foreword — Supplementary information.

The committee responsible for this document is ISO/TC 22, *Road vehicles*, Subcommittee SC 31, *Data communication*.

This second edition cancels and replaces the first edition (ISO 11898-1:2003), which has been technically revised. It also incorporates the Corrigendum ISO 11898-1:2003/Cor 1:2006.

ISO 11898 consists of the following parts, under the general title *Road vehicles — Controller area network (CAN)*:

— *Part 1: Data link layer and physical signalling*

— *Part 2: High-speed medium access unit* [1)]

— *Part 3: Low-speed, fault-tolerant, medium-dependent interface*

— *Part 4: Time-triggered communication*

— *Part 5: High-speed medium access unit with low-power mode* [1)]

— *Part 6: High-speed medium access unit with selective wake-up functionality* [1)]

1) Parts 2, 5, and 6 are being revised. They will be merged under a new edition of Part 2.

Introduction

ISO 11898 was first published as one document in 1993. It covered the CAN data link layer, as well as the high-speed physical layer.

In the reviewed and restructured ISO 11898 series:

— Part 1 defines the data link layer including the logical link control (LLC) sub-layer and the medium access control (MAC) sub-layer, as well as the physical signalling (PHS) sub-layer;

— Part 2 defines the high-speed physical medium attachment (PMA);

— Part 3 defines the low-speed fault-tolerant physical medium attachment (PMA);

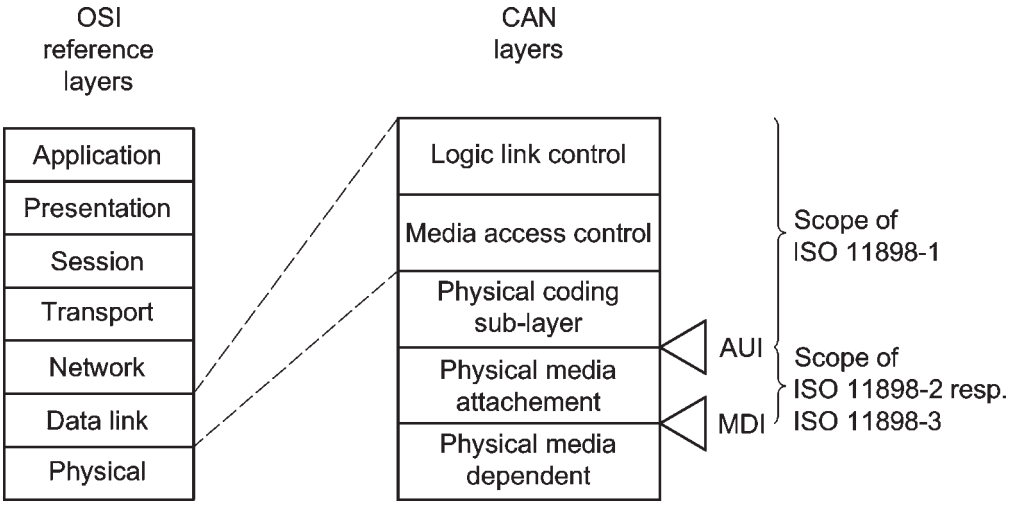
— Part 4 defines the time-triggered communication;

— Part 5 defines the power modes of the high-speed physical medium attachment (PMA);

— Part 6 defines the selective wake-up functionality of the high-speed physical medium attachment (PMA).

NOTE ISO 11898-2 is updated in parallel to the update of this part of ISO 11898 to combine the functions described in ISO 11898-2, ISO 11898-5 and ISO 11898-6. (The future edition of ISO 11898-2 will cancel and replace the current ISO 11898-2:2003, ISO 11898-5:2007 and ISO 11898-6:2013)

Figure 1 shows the relations between the OSI reference layers and the parts of the ISO 11898 series.



NOTE ISO 11898-2 refers to the future edition that will cancel and replace the current ISO 11898-2:2003, ISO 11898-5:2007 and ISO 11898-6:2013.

Figure 1 — CAN data link and physical sub-layers relation to the OSI model

Road vehicles — Controller area network (CAN) —

Part 1:
Data link layer and physical signalling

1 Scope

This part of ISO 11898 specifies the characteristics of setting up an interchange of digital information between modules implementing the CAN data link layer. Controller area network is a serial communication protocol, which supports distributed real-time control and multiplexing for use within road vehicles and other control applications.

This part of ISO 11898 specifies the Classical CAN frame format and the newly introduced CAN Flexible Data Rate Frame format. The Classical CAN frame format allows bit rates up to 1 Mbit/s and payloads up to 8 byte per frame. The Flexible Data Rate frame format allows bit rates higher than 1 Mbit/s and payloads longer than 8 byte per frame.

This part of ISO 11898 describes the general architecture of CAN in terms of hierarchical layers according to the ISO reference model for open systems interconnection (OSI) according to ISO/IEC 7498-1. The CAN data link layer is specified according to ISO/IEC 8802-2 and ISO/IEC 8802-3.

This part of ISO 11898 contains detailed specifications of the following (see Figure 2):

— logical link control sub-layer;

— medium access control sub-layer;

— physical coding sub-layer.

There are three implementation options. They are the following:

— support of the Classical CAN frame format only, not tolerating the Flexible Data Rate frame format;

— support of the Classical CAN frame format and tolerating the Flexible Data Rate frame format;

— support of the Classical CAN frame format and the Flexible Data Rate frame format.

The last option is recommended to be implemented for new designs.

NOTE Implementations of the first option can communicate with implementations of the third option only as long as the Flexible Data Rate frame format is not used; otherwise, Error Frames are generated. There are opportunities to run implementations of the first option also in CAN networks using the Flexible Data Rate frame format, but these are not in the scope of this part of ISO 11898.

2 Conformance

The data link layer conformance test plan is not in the scope of this part of ISO 11898. For an implementation to be compliant with this part of ISO 11898, the logical link control sub-layer and the medium access control sub-layer shall comply with all mandatory specifications and values given in this part of ISO 11898. If optional specifications and values are implemented, they shall comply, too.

3 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 7498-1, *Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model — Part 1*

ISO/IEC/IEEE 8802-3:2014, *Standard for Ethernet — Part 3*

4 Terms and definitions

For the purpose of this part of ISO 11898, the following terms and definitions apply.

4.1
arbitration phase
phase where the nominal bit time is used

4.2
bit stuffing
frame coding method providing bus state changes required for periodic resynchronization when using an NRZ bit representation

Note 1 to entry: 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 de-stuff the Data Frames and the Remote Frames, i.e. the inverse procedure is carried out.

4.3
bus
topology of a communication network, where all nodes are reached by passive links which allow transmission in both directions

4.4
bus comparator
electronic circuit converting physical signals used for transfer across the communication medium back into logical information or data signals

4.5
bus driver
electronic circuit converting information or data signals into physical signals so that these signals can be transferred across the communication medium

4.6
bus state
one of two complementary logical states: dominant or recessive

Note 1 to entry: The dominant state represents the logical 0, and the recessive state represents the logical 1. During simultaneous transmission of dominant and recessive bits, the resulting bus state is dominant. When no transmission is in progress, the bus is idle. During idle time, it is in recessive state

4.7
Classical Base Frame Format
format for Data Frames or Remote Frames using an 11-bit identifier, which are transmitted with one single bit rate and up to and including 8 data bytes

4.8
Classical Extended Frame Format
format for Data Frames or Remote Frames using a 29-bit identifier, which are transmitted with one single bit rate and up to and including 8 data bytes

4.9
Classical Frame
Data Frame or Remote Frame using the Classical Base Frame Format or the Classical Extended Frame Format

4.10
content-based arbitration
CSMA arbitration procedure resolving bus-contention when multiple nodes simultaneously access the bus

4.11
data bit rate
number of bits per time during data phase, independent of bit encoding/decoding

4.12
data bit time
duration of one bit in data phase

4.13
Data Frame
frame containing user data (e.g. one or more signals or one or more suspect parameters of one or more process data)

4.14
data phase
phase where the data bit time is used

4.15
edge
difference in bus-states between two consecutive time quanta

4.16
Error Frame
frame indicating the detection of an error condition

4.17
FD enabled
able to receive and to transmit FD Frames, as well as Classical Frames

4.18
FD Base Frame Format
format for Data Frames using an 11-bit identifier, which are transmitted with a flexible bit rate and up to and including 64 data bytes

4.19
FD Extended Frame Format
format for Data Frames using a 29-bit identifier, which are transmitted with a flexible bit rate and up to and including 64 data bytes

4.20
FD Frame
Data Frame using the FD Base Frame Format or FD Extended Frame Format

4.21
FD intolerant
only able to receive or to transmit Classical Frames, disturbing FD Frames

4.22
FD tolerant
not able to receive or to transmit FD Frames but not disturbing them

4.23
frame
Protocol Data Unit of the data link layer specifying the arrangement and meaning of bits or bit fields in the sequence of transfer across the transmission medium

4.24
handle
hardware object label of one or multiple LLC frames (LPDU)

4.25
higher-layer protocol
protocol above the Data Link Layer protocol according to the Open System Interconnection model

[SOURCE: ISO/IEC 7498-1]

4.26
identifier
does not indicate the destination of the frame but reflects the priority of a particular frame and denotes the meaning of the data

4.27
idle
state of the network, when there is recessive state after the completion of a frame

4.28
idle condition
detection of a sequence of eleven consecutive sampled recessive bits

4.29
integrating
status of a node waiting on an idle condition after it has started the protocol operation during bus-off recovery or after a protocol exception event

4.30
minimum time quantum
smallest time quantum that can be configured for the specific implementation

4.31
multicast
addressing where a single frame is addressed to a group of nodes simultaneously

Note 1 to entry: Broadcast is a special case of multicast, whereby a single frame is addressed to all nodes simultaneously.

4.32
multi master
network with several nodes where every node is able to temporarily control the action of other nodes

4.33
node
assembly, linked to a communication network, capable of communicating across the network according to a communication protocol specification

Note 1 to entry: A CAN node is a node communicating across a CAN network.

4.34
node clock
time base to coordinate the bit-time-related state machines in CAN implementations

4.35
nominal bit rate
number of bits per time during arbitration phase, independent of the bit encoding/decoding

4.36
nominal bit time
duration of one bit in arbitration phase

4.37
Non-Return-to-Zero
method of representing binary signals, i.e. within one and the same bit time, the signal level does not change, where a stream of bits having the same logical value provides no edges

4.38
Overload Frame
frame indicating an overload condition

4.39
priority
attribute to a frame controlling its ranking during the arbitration

Note 1 to entry: A high priority increases the probability that a frame wins the arbitration process.

4.40
protocol
formal set of conventions or rules for the exchange of information between nodes, including the specification of frame administration, frame transfer and PL

4.41
protocol exception event
exception from the formal set of conventions or rules to be able to tolerate future new frame formats

4.42
receiver
any node that is not transmitter or integrating when the bus is not idle

4.43
Remote Frame
frame that requests the transmission of a dedicated Data Frame

4.44
stuff bit count
number of stuff bits in a frame before the CRC field, not including fixed stuff bits

4.45
time-triggered communication
option where a frame can be transmitted in a defined time slot, which also provides a network-wide synchronization of clocks, as well as disabling of the automatic retransmission of frames, so that dedicated data and remote frames avoid collisions with data and remote frames transmitted by other nodes

4.46
transceiver
electronic circuit that connects a CAN node to a CAN network, consisting of a bus comparator and a bus driver

4.47
transmitter
node originating a data frame or remote frame, and stays transmitter until the bus is idle again or until the node loses arbitration

5 Symbols and abbreviated terms

ACK Acknowledgement

AUI	Attachment Unit Interface
BCH	Bose-Chaudhuri-Hocquenghem
BRS	Bit Rate Switch
CAN	Controller area network
CBFF	Classical Base Frame Format
CEFF	Classical Extended Frame Format
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
DF	Data Frame
DLC	Data Length Code
DLL	Data Link Layer
EF	Error Frame
EOF	End Of Frame
ESI	Error State Indicator
FBFF	FD Base Frame Format
FCE	Fault Confinement Entity
FD	Flexible Data Rate
FDF	FD Format indicator
FEFF	FD Extended Frame Format
HLP	Higher-Layer Protocols
IDE	IDentifier Extension
IPT	Information Processing Time
LAN	Local Area Network
LLC	Logical Link Control
LME	Layer Management Entity
LPDU	LLC Protocol Data Unit
LSDU	LLC Service Data Unit
MAC	Medium Access Control
MAU	Medium Attachment Unit
MDI	Medium Dependent Interface
MPDU	MAC Protocol Data Unit

MSB	Most Significant Bit
MSDU	MAC Service Data Unit
NRZ	Non-Return-to-Zero
OF	Overload Frame
OSI	Open Systems Interconnection
OVLD	Overload
PCI	Protocol Control Information
PCS	Physical Coding Sub-layer
PDU	Protocol Data Unit
PL	Physical Layer
PMA	Physical Medium Attachment
r0	Reserved bit in Classical Extended Frame Format
res	Reserved bit in FD Frames
RF	Remote Frame
RRS	Remote Request Substitution
RTR	Remote Transmission Request
SAP	Service Access Point
SDU	Service Data Unit
SJW	Synchronization Jump Width
SOF	Start Of Frame
SP	Sample Point
SRR	Substitute Remote Request
SSP	Secondary Sample Point
TDC	Transmitter Delay Compensation

6 Basic concepts of CAN

6.1 CAN properties

CAN has the following properties:

— multi-master priority-based bus access;

— non-destructive content-based arbitration;

— all frame transfer is done as broadcast;

— multicast frame transfer by acceptance filtering;

— remote data request;

— configuration flexibility;

— network-wide data consistency;

— error detection and error signalling;

— automatic retransmission of frames that have lost arbitration, have not been acknowledged, or have been destroyed by errors during transmission;

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

6.2 Frames

Information on the bus is sent in fixed format frames of different but limited length. When the bus is idle, any connected node is allowed to start the transmission of a DF or RF. The bus is idle when no frames are transmitted. Additionally, any connected node may start the indication of error and overload information by means of dedicated frames (EF and OF, respectively).

6.3 Bus access method

If two or more nodes start to transmit DFs or RFs at the same time, the bus access conflict is resolved by content-based arbitration using the identifier. The mechanism of arbitration ensures that neither information nor time is lost. The transmitter with the DF or RF of the highest priority gains the bus access. A DF with the same ID as an RF wins bus arbitration.

6.4 Information routing

A node does not make use of any information about the network 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.

6.5 Network flexibility

Nodes can 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 DF and if the added node does not require any additional transmitted data.

6.6 Data consistency

Within a CAN network, a frame can be simultaneously accepted as a valid frame either by all nodes or by no node. Thus data consistency is a property of the CAN network achieved by the concepts of broadcast and by error handling.

6.7 Remote data request

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

NOTE 1 The node having the message with the requested ID decides whether new data are produced or data in a transmit buffer will be sent.

NOTE 2 The node having the message with the requested data decides how to respond to an RF with mismatching DLC.

6.8 Error detection

For detecting errors, the following measures are provided:

— monitoring (transmitters compare the transmitted bit levels with the bit levels detected on the network);

— 15-bit CRC for Classical Frames, 17-bit CRC for FD Frames with up to 16 data-field bytes, 21-bit CRC for FD Frames with 20 to 64 data-field bytes;

— stuff bit count check for FD Frames;

— variable bit stuffing with a stuff width of five (except in the CRC field of FD Frames);

— frame format check;

— ACK check.

6.9 Error signalling and recovery time

Corrupted frames are flagged by any transmitting node and any normally operating (error-active) receiving node. Such frames are aborted and retransmitted according to the implemented recovery procedure (see 8.3.4). The recovery time from detecting an error until the possible start of the next frame is typically 17 to 23 nominal bit times (in the case of nodes in error passive mode up to 31 nominal bit times), if there are no further errors.

6.10 ACK

All receivers check the consistency of the received DFs and RFs, acknowledge a consistent frame, and flag an inconsistent frame by means of an EF. A transmitting node regards a DF or RF that is not acknowledged as corrupted.

6.11 Automatic retransmission

Frames that have lost arbitration, frames that have not been acknowledged, and frames that have been disturbed by errors during transmission are retransmitted automatically until their transmission has been successfully completed or until their transmission is no longer requested (see 8.3.4 and 10.9.6). Optionally, the automatic retransmission may be disabled (see 9.3). Optionally, the automatic retransmission may be limited to a certain number of attempts (see 10.9.6).

6.12 Fault confinement

CAN nodes are able to distinguish short disturbances from permanent failures. Defective transmitting nodes are switched off. Switched off means a node is logically disconnected from the bus, so that it can neither send nor receive any frames (see 12.1.4.4).

6.13 Error-active

An error-active node normally takes part in bus communication and sends an active error flag when an error has been detected. The active error flag consists of 6 consecutive dominant bits and violates the rule of bit-stuffing and all fixed formats appearing in a DF and RF (see 12.1.4.2).

6.14 Error-passive

An error-passive node sends no 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 6 consecutive recessive bits. After transmission, an error-passive node waits some additional time before initiating a further transmission (see suspend transmission in 10.4.6.4 and 12.1.4.2).

6.15 Bus-off

A node is in the bus-off state when it is switched off from the bus due to a request of FCE. In the bus-off state, a node neither sends nor receives frames. In the bus-off state, a node does not send any dominant bits.

7 Layered architecture of CAN

7.1 Reference to OSI model

According to the OSI reference model (see ISO/IEC 7498-1), the CAN architecture of this part of ISO 11898 represents two layers (see Figure 2),

— DLL, and

— PCS of PL.

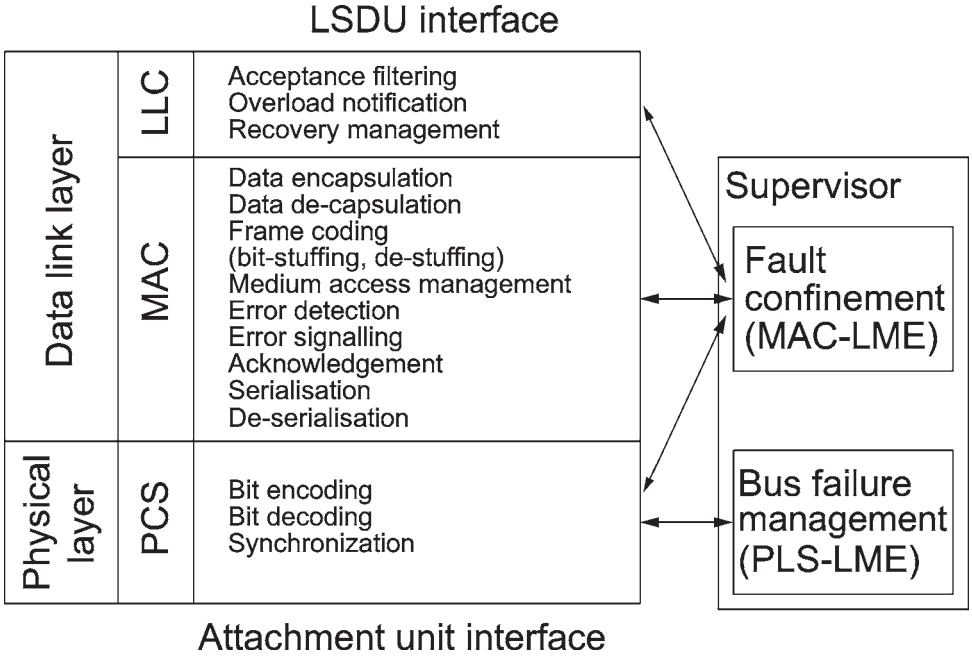


Figure 2 — Layered architecture of CAN

According to ISO/IEC 8802-2 and ISO/IEC 8802-3, the DLL is subdivided into

— LLC, and

— MAC.

The PL is subdivided into

— PCS,

— PMA, and

— MDI.

The MAC sub-layer operations are supervised by the FCE. Fault confinement is a self-checking mechanism that distinguishes short disturbances from permanent failures (see 12.1).

Optionally, the PL is supervised by an entity that detects and manages failures of the physical medium (for example, shorted or interrupted bus lines, see 12.2).

7.2 Protocol specification

Two peer protocol entities shall communicate with each other by exchanging frames or PDUs.

An (*N*)-layer Protocol Data Unit (PDU_N) consists of (*N*)-layer specific protocol control information (PCI_N) and (*N*)-layer user data. PDU_N shall be passed to a (*N*-1)-layer entity via an SAP_{N-1}. The PDU_N shall be passed by means of the SDU_{N-1} to the (*N*-1)-layer, the services of which allow the transfer of the PDU_N. The SDU shall be the interface data whose identity is preserved between (*N*)-layer entities, i.e. it represents the logical data unit transferred by a service. The DLL of the CAN protocol shall provide neither means for mapping one SDU into multiple PDUs nor means for mapping multiple SDUs into one PDU, i.e. a PDU_N is directly constructed from the associated SDU_N and the layer specific control information PCI_N. Figure 3 illustrates the data link sub-layer interactions.

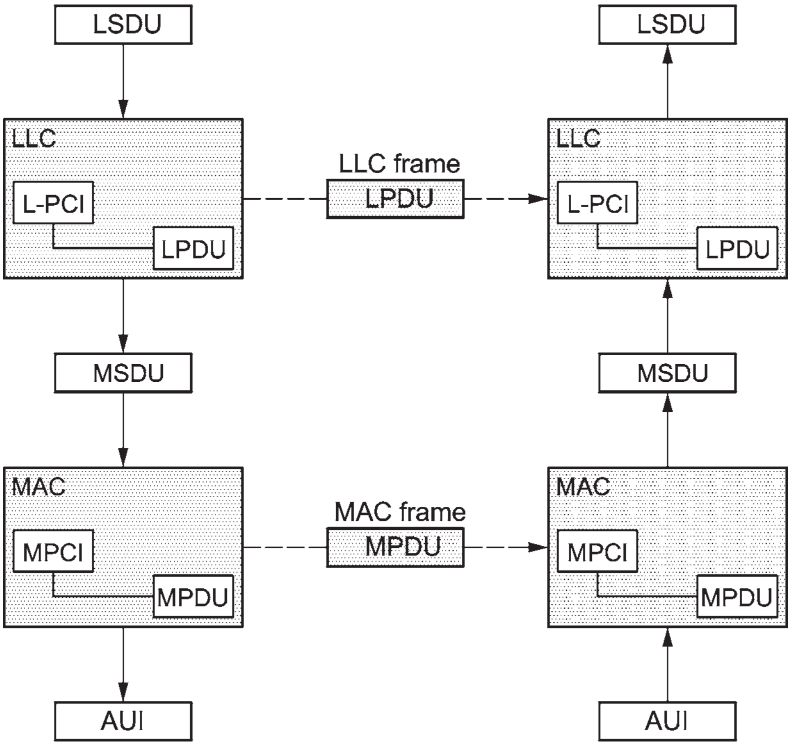


Figure 3 — Protocol layer interactions

7.3 Format description of services

7.3.1 Format description of service primitives

Service primitives shall be written as:

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

where

`service`	indicates the name of the service, e.g. L_Data for data transfer service provided by the LLC sub-layer;
`type`	indicates the type of the service primitives (see 7.3.2);
`[parameter1,…]`	is the list of values passed to the service primitives.

The brackets indicate that this parameter list may be empty.

7.3.2 Types of service primitives

Service primitives shall be of three generic types.

a) Service.Request

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

b) Service.Indication

The indication primitive shall be passed from the (*N*)-layer to the (*N*)-user to indicate an internal (*N*)-layer (or sub-layer) 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 sub-layer).

c) Service.Confirm

The confirm primitive shall be passed from the (*N*)-layer (or sub-layer) 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 shall not necessarily indicate any activity at the remote peer interface.

7.4 LLC interface

The LLC sub-layer shall offer two types of connectionless transmission services to the LLC user:

— unacknowledged data transfer service;

— unacknowledged remote data request service.

The interface service data sent from or to the user shall be as given in 8.2.2. The messages sent between LLC user and LLC sub-layer are specified in Table 1 and Table 2.

Table 1 — Message sent from LLC user to LLC sub-layer

Message	Description
Reset_Request	Request to set the node into an initial state

Table 2 — Message sent from LLC sub-layer to LLC user

Message	Description
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 bus-off state.

The LLC interface messages sent from and to the supervisor FCE shall be as specified in 12.1.3.

8 Description of LLC sub-layer

8.1 General

The LLC sub-layer describes the upper part of the DLL according to ISO/IEC 8802-2. It is associated with those protocol issues that are independent of the type of the medium access method.

8.2 Services of LLC sub-layer

8.2.1 Types of connectionless-mode transmission services

The LLC sub-layer shall offer two types of connectionless-mode transmission services:

Unacknowledged data transfer service

This service shall provide means by which LLC users exchange LSDU without establishing a data link connection. The data transfer may be point-to-point, multicast or broadcast.

Unacknowledged remote data request service

This service shall provide means used by an LLC user to request a remote node for an LSDU transmission without establishing a data link connection.

The remote node shall basically serve the data request in the following two ways.

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

— The requested data shall be transmitted by the remote user upon reception of the remote request frame.

According to the two different LLC services, six types of frames may be used for the communication between transmitting node and receiving nodes.

— LLC Data Frame in Classical Base Frame Format;

— LLC Data Frame in Classical Extended Frame Format;

— LLC Data Frame in FD Base Frame Format;

— LLC Data Frame in FD Extended Frame Format;

— LLC Remote Frame in Classical Base Frame Format;

— LLC Remote Frame in Classical Extended Frame Format.

The LLC DFs shall carry data from a transmitter to a receiver. The LLC RFs shall be transmitted to request transmission of a DF (with the same identifier) from a (single) remote node. In both cases, the LLC sub-layer shall notify the successful transmission or reception of a DF or RF to the LLC user.

8.2.2 Service primitive specification

8.2.2.1 General

The service primitive specification of this sub-clause describes in detail the LLC service primitives and their associated parameters. The complete list of LLC service primitives shall be as given in Table 3.

Table 3 — LLC service primitives overview

Service	Service primitives	Description
Unacknowledged data transfer	L_Data.Request	Request for data transfer
	L_Data.Indication	Indication of data transfer
	L_Data.Confirm	Confirm data transfer
	L_Data.AbortRequest (optional)	Request abortion of data transfer
Unacknowledged remote data request	L_Remote.Request	Request for remote data request
	L_Remote.Indication	Indication of remote data request
	L_Remote.Confirm	Confirmation remote data request
	L_Remote.AbortRequest(optional)	Request abortion of remote data request

The parameters associated with the different LLC service primitives shall be as given in Table 4.

Table 4 — List of LLC service primitive parameters

Parameter	Description
Identifier	Identifies the content of the frame
Format	Frame format (CBFF, CEFF, FBFF, FEFF, optionally giving ESI and BRS bit values)
DLC	Length of the data
Data	Data the user wants to transmit
Transfer_Status	Confirmation parameter
Handle (optional)	Identifies hardware element used for transaction

8.2.2.2 L_Data.Request

Function

The L_Data.Request primitive shall be passed from the LLC user to the LLC sub-layer to request that an LSDU be sent to one or more remote LLC entities.

Semantics of L_Data.Request primitive

The primitive shall provide parameters as follows:

```
L_Data.Request(
     Identifier
     Format
     DLC
     Data
     Handle
     )
```

The parameter Data shall be insignificant if the associated LLC DF is of data length zero. Optionally, the hardware element (message storage unit) to be used for the transmission is identified by the Handle.

Effect on receipt

Receipt of this primitive shall cause the LLC sub-layer to initiate the transfer of an LLC DF by use of the data transfer service provided by the MAC sub-layer (see Table 6). Any L_Data.Request shall be processed not later than the second SOF after the request when there are no error frames present during this time.

8.2.2.3 L_Data.indication

Function

The L_Data.Indication primitive shall be passed from the LLC sub-layer to the LLC user to indicate the arrival of an LSDU.

Semantics of L_Data.Indication primitive

The primitive shall provide parameters as follows:

```
L_Data.Indication(
     Identifier
     Format
     DLC
     Data
     )
```

The parameter Data shall be insignificant if the associated LLC DF is of data length zero.

Effect on receipt

The effect on receipt of this primitive by the LLC user is unspecified.

8.2.2.4 L_Data.Confirm

Function

The L_Data.Confirm primitive shall be passed from the local LLC sub-layer to the LLC user to convey the results of the previous L_Data.Request primitive. This primitive shall be a local confirmation, i.e. it shall not imply that the remote LLC entity or entities have passed the associated indication primitive to the corresponding LLC user(s).

Semantics of L_Data.Confirm primitive

The primitive shall provide parameters as follows:

```
L_Data.Confirm(
     Identifier
     Transfer_Status
     Handle
     )
```

The Transfer_Status shall be used to indicate the completion of the transaction status

— initiated by the previous L_Data.Request primitive (if optional Handle is not present), or

— of the optionally referred hardware element given by Handle.

Transfer_Status:[Complete, Not_Complete, Aborted]

The transfer status Aborted is available if the optional service L_Data.AbortRequest is provided and supported.

Effect on receipt

The effect on receipt of this primitive by the LLC user is unspecified.

8.2.2.5 L_Data.AbortRequest (optional)

Function

The optional L_Data.AbortRequest primitive shall be passed from the LLC user to the LLC sub-layer to abort a request of transmission of an LSDU, which had been requested before.

Semantics of L_Data.AbortRequest primitive

The primitive shall provide parameters as follows:

```
L_Data.AbortRequest(
     Handle
     )
```

By Handle, the hardware element (message storage unit) for which the transmission shall be aborted is identified.

Effect on receipt

Receipt of this primitive shall cause the LLC sub-layer to abort the transfer of an LLC DF in the specified message storage unit. Pending transmissions, which already have been passed to the MAC sub-layer, shall only be aborted if

— an error in the MAC sub-layer during transmission occurs, or

— arbitration was lost in the MAC sub-layer,

which causes the LSDU to wait for another transmission attempt.

This means that an abortion request must be kept pending in the LLC layer until either one of the above situations occur, or until the transmission was completed.

The LLC sub-layer cannot immediately abort transmissions which have already been submitted to the MAC sub-layer (due to priority scheduling of the requested hardware element indicated by Handle). Any L_Data.AbortRequest shall be processed prior to the second SOF after the request when there are no error frames present during this time. Any L_Data.AbortRequest shall be processed prior to the third SOF after the request when there are error frames present during this time.

8.2.2.6 L_Remote.Request

Function

The L_Remote.Request primitive shall be passed from the LLC user to the LLC sub-layer to request a single remote LLC entity to transmit an LSDU.

Semantics of L_Remote.Request primitive

The primitive shall provide parameters as follows:

```
L_Remote.Request(
     Identifier
     Format
     DLC
     Handle
     )
```

The value of DLC equals the length of the data field of the requested DF. By the optional Handle, the hardware element (message storage unit) to be used for the transmission is identified.

Effect on receipt

Receipt of this primitive shall cause the LLC sub-layer to initiate the transfer of an LSDU in the optionally specified message storage unit by use of the remote data transfer service provided by the MAC sub-layer (see Table 6).

8.2.2.7 L_Remote.Indication

Function

The L_Remote.Indication primitive shall be passed from the LLC sub-layer to the LLC user to indicate the arrival of a request for transmission of an LSDU.

Semantics of L_Remote.Indication primitive

The primitive shall provide parameters as follows:

```
L_Remote.Indication(
     Identifier
     Format
     DLC
     )
```

The identifier shall identify the LSDU to be sent. The value of DLC equals the length of the data field of the requested DF.

Effect on receipt

The effect on receipt of this primitive by the LLC user is unspecified.

8.2.2.8 L_Remote.Confirm

Function

The L_Remote.Confirm primitive shall be passed from the local LLC sub-layer to the LLC user to convey the results of the previous L_Remote.Request primitive. This primitive shall be a local confirmation, i.e. it does not imply that the remote LLC entity has passed the associated indication primitive to the corresponding LLC user.

Semantics of L_Remote.Confirm primitive

The primitive shall provide parameters as follows:

```
L_Remote.Confirm(
     Identifier
     Transfer_Status
     Handle
     )
```

The Transfer_Status shall be used to indicate the transaction status

— of the previous L_Remote.Request primitive (if optional Handle is not present), or

— of the optionally referred hardware element given by Handle.

The Transfer_Status shall be used to indicate the completion of the transaction initiated by the previous L_Remote.Request primitive.

Transfer_Status:[Complete, Not_Complete, Aborted]

The transfer status Not_Complete shall be given when either a transmission request is still pending or ongoing, or if an optional abortion request is pending.

If the optional service L_Remote.AbortRequest is provided and supported, the transfer status Aborted is available.

Effect on receipt

The effect on receipt of this primitive by the LLC user is unspecified.

8.2.2.9 L_Remote.AbortRequest (optional)

Function

The optional L_Remote.AbortRequest primitive shall be passed from the LLC user to the LLC sub-layer to abort a request to a single remote LLC entity to transmit an LSDU (i.e. an RF), which had been requested before.

Semantics of L_Remote.AbortRequest primitive

The primitive shall provide parameters as follows:

```
L_Remote.AbortRequest(
     Handle
     )
```

By Handle, the hardware element (message storage unit) for which the transmission shall be aborted is identified.

Effect on receipt

Receipt of this primitive shall cause the LLC sub-layer to abort the transfer of an LLC RF in the specified message storage unit. Pending transmissions, which already have been passed to the MAC sub-layer, can only be aborted if

— an error in the MAC sub-layer during transmission occurs, or

— arbitration was lost in the MAC sub-layer,

which causes the LSDU to wait for another transmission attempt.

This means that an abortion request must be kept pending in the LLC layer, until either one of the above situations occur, or until the transmission was completed.

The LLC sub-layer cannot immediately abort transmissions which have already been submitted to the MAC sub-layer (due to priority scheduling of the requested hardware element indicated by the Handle).

8.3 Functions of LLC sub-layer

8.3.1 General

The LLC sub-layer shall provide the following functions:

— frame acceptance filtering;

— overload notification;

— recovery management.

8.3.2 Frame acceptance filtering

A frame transaction initiated at the LLC sub-layer shall be a single, self-contained operation independent of previous frame transactions. The content of a frame shall be named by its identifier. Each receiver shall decide by frame acceptance filtering whether the frame is relevant or not.

8.3.3 Overload notification

The transmission of a MAC OF shall be initiated by the LLC sub-layer if internal conditions of a receiver require delay of the next LLC DF or LLC RF. If there are internal conditions of a CAN implementation that cause a MAC OF to be initiated, these conditions shall be documented for that CAN implementation.

At most, two MAC OF may be generated to delay the next DF or RF.

8.3.4 Recovery management

The LLC sub-layer shall provide means for automatic retransmission of frames that lost arbitration, that have not been acknowledged, or that have been disturbed by errors during transmission. However, the automatic retransmission can be aborted by the optional LLC AbortRequests. The frame transmission service shall not be reported as confirmed or (optionally) aborted to the user before the transmission has been successfully completed or (optionally) aborted. The automatic retransmission of a frame shall be disabled when the transmission of that frame is no longer requested. The automatic retransmission may be disabled for all frames.

8.4 Structure of LLC frames

8.4.1 General

LLC frames shall be the data units exchanged between peer LLC entities (LPDU). The structure and format of the LLC DF and RF shall be specified subsequently. The optional Handle of an LLC frame is kept within the LLC itself as a label for use with upper layer communications. Because of this, its representation is invisible on the MAC layer and thus invisible on the CAN bus. In the following, the LLC frame Handle is no longer considered.

8.4.2 Specification of LLC DF

8.4.2.1 General

An LLC DF shall be composed of four bit fields (see Figure 4):

— identifier field;

— format field;

— DLC field;

— LLC data field.

Figure 4 — LLC DF

8.4.2.2 Identifier field

The identifier field shall be composed of two segments, the base identifier and the identifier extension. The length of the base identifier shall be 11 bit (ID-28 to ID-18) and the length of the identifier extension shall be 18 bit (ID-17 to ID-0). The identifier extension shall be ignored for frames in CBFF and in FBFF.

8.4.2.3 Format field

This field distinguishes between frames in CBFF, CEFF, FBFF, and FEFF. In FBFF and FEFF, it includes the ESI bit and the BRS bit (see Table 4).

8.4.2.4 DLC field

The number of bytes in the data field shall be indicated by the DLC; see Table 5. This DLC shall consist of 4 bit. The admissible number of data bytes for Classical Frames shall range from 0 to 8. DLCs in the range of 0 to 7 shall indicate data fields of length of 0 byte to 7 byte. In Classical Frames, all other DLCs shall indicate that the data field is 8 byte long. In FD Frames, DLCs in the range of 0 to 8 shall indicate data fields of length of 0 byte to 8 byte, while other DLCs code longer data fields, according to Table 5.

Table 5 — Coding of the numbers of data bytes by the DLC

Frames	Data length code				Number of data bytes
	DLC3	DLC2	DLC1	DLC0	
Classical Frames and FD Frames	0	0	0	0	0
	0	0	0	1	1
	0	0	1	0	2
	0	0	1	1	3
	0	1	0	0	4
	0	1	0	1	5
	0	1	1	0	6
	0	1	1	1	7
	1	0	0	0	8
Classical Frames	1	0 or 1	0 or 1	0 or 1	8
FD Frames	1	0	0	1	12
	1	0	1	0	16
	1	0	1	1	20
	1	1	0	0	24
	1	1	0	1	32
	1	1	1	0	48
	1	1	1	1	64

8.4.2.5 Data field

The data field shall consist of the data to be transferred within a DF. It may contain from 0 byte to 8 byte in Classical Frames or it may contain from 0 byte to 64 byte in FD Frames, where each byte shall contain 8 bit.

8.4.3 Specification of LLC RF

An LLC RF shall be composed of three bit fields (see Figure 5):

— identifier field;

— format field;

— DLC field.

Figure 5 — LLC RF

The formats of both the LLC RF identifier field and DLC field shall be identical to the formats of the LLC DF identifier field (see 8.4.2.2) and DLC field (see 8.4.2.4). There shall be no data field, independent of the value of the DLC. The format field of an LLC RF distinguishes only between frames in CBFF and CEFF. There shall be no LLC RF in FBFF or FEFF.

RFs shall only be transmitted with a network-wide determined DLC, which is the DLC of the corresponding DF (see 10.9.9).

8.5 Limited LLC frames

It is not required to implement the full range of possible identifiers or DLCs.

If an LLC sub-layer is restricted to the use of a sub-range of identifiers (e.g. only 11-bit identifiers), then it shall be limited to LLC DFs and LLC RFs with identifiers of that sub-range (e.g. identifiers with their IDE bit set to logic 0 and their identifier extension is ignored).

If an LLC sub-layer is restricted to the use of less than the maximum number of data bytes, then it shall be limited to LLC DFs with a number of data bytes of that restricted range. If the DLC indicates a higher number of data bytes, the data bytes beyond the restricted range shall be replaced in the LLC DF by bytes with the value of CC_{HEX} ("padding") both for received and for transmitted frames. Optionally, the CAN implementation may support a configuration where the CAN node will not transmit a frame whenever the frame's DLC indicates a number of data bytes beyond the restricted range.

NOTE The padding for received messages does not need to be implemented in the LLC layer.

9 Interface between LLC and MAC

9.1 Services

The MAC sub-layer shall provide services to the local LLC for

— (MAC-) acknowledged transfer of LLC frames, and

— transmission of MAC OFs.

The interface service data from or to the LLC sub-layer shall be as described in 8.3.

9.2 Time and time triggering

9.2.1 Description

Optionally, CAN enabled implementations may support network-wide synchronization of clocks and if so, they may additionally support time-triggered communication. The clock synchronization option describes the prerequisites needed for the alignment of node clocks in a network. In order to synchronize the clocks of the nodes within the network, a common reference point is needed. The SOF bit or the sample point of the last bit of EOF of any message shall be used as the reference point. Synchronization of node clocks facilitates the establishment of a network-wide time base in higher-layer protocols. If implemented, the time triggered communication option facilitates frames to be transmitted in defined time slots.

The hardware needed to establish a network-wide time base shall be included between LLC and MAC.

9.2.2 Time base

Any node that supports time and time triggering option shall provide a time base. The time base is a cyclic up counter of at least 16 bit fed with clock ticks generated by an internal or an external tick generator.

9.2.3 Time reference point

Any message received or transmitted shall invoke a capture of the time base taken at the SOF recognition of the respective message or at the sample point of the last bit of EOF. After successful message reception, the capture value shall be provided to the CPU for at least one message and shall be readable until the next message is received.

9.2.4 Event generation

It shall be possible to generate at least one programmable event trigger from the above-mentioned time base. The trigger shall be freely programmable by the CPU in the range of at least zero to $(2^{16} - 1)$ × time ticks.

9.3 Disabling automatic retransmission

9.3.1 Retransmission of frames

Automatic retransmission may be disabled (see 6.11).

9.4 Message time stamping

Optionally, CAN FD enabled or not FD enabled implementations may support a message time stamping function for received and transmitted data frames.

The time-stamp shall have a width of 8 bit, 16 bit, or 32 bit. The clock source for the time-base shall be generated internally in the node or shall be provided by the LLC user. The time-base counter shall be incremented and shall overrun to zero.

The time-base counter may be readable by the LLC user at any time. The time-base value shall be captured at the reference point of each data frame. For Classical Frames, the reference point is the sample point of SOF of the respective frame or the point-in-time when the frame is taken to be valid according to 10.7. For FD Frames, meaning when FDF is recessive and the subsequent res bit is dominant, the reference point is the sample point of SOF, or the sample point of the res bit, or the point-in time when the frame is taken to be valid according to 10.7.

After EOF, the captured time-stamp value shall be readable by the LLC user.

10 Description of MAC sub-layer

10.1 General

The MAC sub-layer represents the lower part of the OSI DLL. It shall service the interface to the LLC sub-layer and the PL, and comprises the functions and rules related to

— encapsulation/de-capsulation of the transmit/receive data,

— error detection and signalling, and

— management of the transmit/receive medium access.

10.2 Services of MAC sub-layer

10.2.1 Service description

The services provided by the MAC sub-layer shall allow the local LLC sub-layer entity to exchange MSDU with the peer LLC sub-layer entities. The MAC sub-layer services shall be the following:

a) **acknowledged data transfer**

This service shall provide means by which LLC entities exchange MSDUs without the establishment of a data link connection. The data transfer may be point-to-point, multicast, or broadcast.

b) **acknowledged remote data request**

This service shall provide means by which an LLC entity requests another remote node to transmit an LSDU without the establishment of a data link connection. The remote LLC entity shall use the

MAC service “acknowledged data transfer” for the transmission of the requested data. ACK of a service shall be generated by the MAC sub-layer(s) of the remote node(s). ACK shall not contain any data of the remote node user.

c) **OF transfer**

This service shall provide means by which an LLC entity initiates the transmission of an OF, a special fixed format LPDU, causing the delay of the next DF or RF.

10.2.2 Service primitives specification

10.2.2.1 General

The service primitives of the MAC sub-layer provided to the LLC sub-layer shall be as given in Table 6.

Table 6 — MAC sub-layer service primitives

Service	Service primitives
Acknowledged data transfer	MA_Data.Request MA_Data.Indication MA_Data.Confirm
Acknowledged remote data request	MA_Remote.Request MA_Remote.Indication MA_Remote.Confirm
OF transfer	MA_OVLD.Request MA_OVLD.Indication MA_OVLD.Confirm

10.2.2.2 MA_Data.Request

Function

The MA_Data.Request primitive shall be passed from the LLC sub-layer to the MAC sub-layer to request that an MSDU be sent to one or more remote MAC entities.

Semantics of MA_Data.Request primitive

The primitive shall provide parameters as follows:

```
MA_Data.Request(
     Identifier
     Format
     DLC
     Data
     )
```

The parameter Data is insignificant for MAC DFs of data length zero.

Effect on receipt

Receipt of this primitive shall cause the MAC sub-layer to prepare a PDU by including all MAC specific control information (SOF, SRR bit, IDE bit, RTR (or RRS) bit, FDF bit, res (or r0) bit, BRS bit, ESI bit, CRC, recessive bits during ACK field, EOF) to the MSDU coming from the LLC sub-layer. For details, see 10.4.2.3 and 10.4.2.4. The MPDU shall be serialized and passed bit by bit as an SDU to the PL for transfer to the peer MAC sub-layer entity or entities.

10.2.2.3 MA_Data.Indication

Function

The MA_Data.Indication primitive shall be passed from the MAC sub-layer to the LLC sub-layer to indicate the arrival of an MSDU.

Semantics of MA_Data.Indication primitive

The primitive shall provide parameters as follows:

```
MA_Data.Indication(
     Identifier
     Format
     DLC
     Data
     )
```

The parameter Data is insignificant if the associated MAC DF is of data length zero. The arrival of an MSDU shall be indicated to the LLC sub-layer only if it has been received correctly.

Effect on receipt

The effect on receipt of this primitive by the LLC sub-layer is unspecified.

10.2.2.4 MA_Data.Confirm

Function

The MA_Data.Confirm primitive shall be passed from the local MAC sub-layer to the LLC sub-layer to convey the results of the previous MA_Data.Request primitive. This primitive is a remote confirmation, i.e. it shall indicate that the remote MAC entity or entities have passed the associated indication primitive to the corresponding user(s).

Semantics of MA_Data.Confirm primitive

The primitive shall provide parameters as follows:

```
MA_Data.Confirm(
     Identifier
     Transmission_Status
     )
```

The Transmission_Status shall be 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 loss of arbitration.

Effect on receipt

The effect on receipt of this primitive by the LLC sub-layer is unspecified.

10.2.2.5 MA_Remote.Request

Function

The MA_Remote.Request primitive shall be passed from the LLC sub-layer to the MAC sub-layer to request a single remote MAC entity to transmit an MSDU.

Semantics of MA_Remote.Request primitive

The primitive shall provide parameters as follows:

```
MA_Remote.Request(
     Identifier
     Format
     DLC
     )
```

The identifier shall identify the MSDU to be sent. The value of DLC shall be equal to the length of the requested MSDU data.

Effect on receipt

Receipt of this primitive shall cause the MAC sub-layer to prepare a PDU by including all MAC specific control information (SOF, SRR bit, IDE bit, RTR bit, FDF bit, r0 bit, CRC, recessive bits during ACK field, EOF) to the MSDU coming from the LLC sub-layer. For details, see 10.4.2.3 and 10.4.2.4. The MPDU shall be serialized and passed bit by bit as an SDU to the PL for transfer to the peer MAC sub-layer entity or entities.

10.2.2.6 MA_Remote.Indication

Function

The MA_Remote.Indication primitive shall be passed from the MAC sub-layer to the LLC sub-layer to indicate the arrival of a request for transmission of an MSDU.

Semantics of MA_Remote.Indication primitive

The primitive shall provide parameters as follows:

```
MA_Remote.Indication(
     Identifier
     Format
     DLC
     )
```

The arrival of an MSDU transmission request shall be indicated to the LLC sub-layer only if it has been received correctly.

Effect of receipt

The effect of receipt on this primitive by the LLC sub-layer is unspecified.

10.2.2.7 MA_Remote.Confirm

Function

The MA_Remote.Confirm primitive shall be passed from the local MAC sub-layer to the LLC sub-layer to convey the results of the previous MA_Remote.Request. This primitive is a remote confirmation, i.e. it shall indicate that the remote MAC entity or entities have passed the associated indication primitive to the corresponding user(s).

Semantics of MA_Remote.Confirm primitive

The primitive shall provide parameters as follows:

```
MA_Remote.Confirm(
     Identifier
     Transmission_Status
     )
```

The Transmission_Status shall be 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 arbitration.

Effect on receipt

The effect on receipt of this primitive by the LLC sub-layer is unspecified.

10.2.2.8 MA_OVLD.Request

Function

The MA_OVLD.Request primitive shall be passed from the LLC sub-layer to the MAC sub-layer to request transmission of a MAC OVLD frame (see 10.4.5). The OVLD frame shall be a fixed format frame and completely constructed in the MAC sub-layer.

Semantics of MA_OVLD.Request primitive

The primitive shall not provide any parameter:

```
MA_OVLD.request(
     )
```

Effect on receipt

Receipt of this primitive shall cause the MAC sub-layer to form an OF. The OF shall be passed to the lower protocol layers for transfer to the peer MAC sub-layer entities.

10.2.2.9 MA_OVLD.Indication

Function

The MA_OVLD.Indication primitive shall be passed from the MAC sub-layer to the LLC sub-layer to indicate that an OF has been received (see 10.4.5).

Semantics of MA_OVLD.Indication primitive

The primitive shall not provide any parameters:

```
MA_OVLD.Indication(
     )
```

Effect on receipt

The effect on receipt of this primitive by the LLC sub-layer is unspecified.

10.2.2.10 MA_OVLD.Confirm

Function

The MA_OVLD.Confirm primitive shall be passed from the MAC sub-layer to the LLC sub-layer to indicate that an OF has been sent. This confirmation shall be local, i.e. it shall not imply that the remote peer protocol entities have received the OF correctly.

Semantics of MA_OVLD.Confirm primitive

The primitive shall provide parameters as follows.

```
MA_OVLD.Confirm(
     Transmission_Status
     )
```

The Transmission_Status shall be used to indicate the success or failure of the previous MA_OVLD. request primitive.

Transmission_Status:[Success, No_Success]

Effect on receipt

The effect on receipt of this primitive by the LLC sub-layer is unspecified.

10.3 Functional model of MAC sub-layer architecture

10.3.1 Capability

The functional capabilities of the MAC sub-layer are described by use of the functional model specified in ISO/IEC 8802-3 (see also Figure 6). In this model, the MAC sub-layer is divided into two fully independently-operating parts, i.e. the transmit part and the receive part. The functions of both transmit and receive parts shall be as given in this clause and Figure 6.

10.3.2 Frame transmission

Frame transmission shall fulfil the following requirements:

a) Transmit data encapsulation

- acceptance of LLC frames and interface control information;
- CRC sequence calculation including stuff bit count for FD Frames;
- construction of MAC frame by adding SOF, SRR bit (if used in frame format), IDE bit, RTR (or RRS) bit, FDF bit, res bit, BRS bit (if used in frame format), ESI bit (if used in frame format), CRC, ACK and EOF to the LLC frame (restricted LLC sub-layers may not request the transmission of MAC frames with identifiers or data fields outside their restrictions; see 8.5).

b) Transmit medium access management

- initiation of the transmission process after recognizing bus idle (compliance with inter-frame space);
- serialization of the MAC frame;
- insertion of stuff bits (bit stuffing);
- arbitration and passing into receive mode in case of loss of arbitration;
- error detection (monitoring, format check);
- ACK check;
- recognition of an overload condition;
- OF construction and initiation of transmission;
- EF construction and initiation of transmission;
- presentation of a serial bit stream to the PL for transmission.

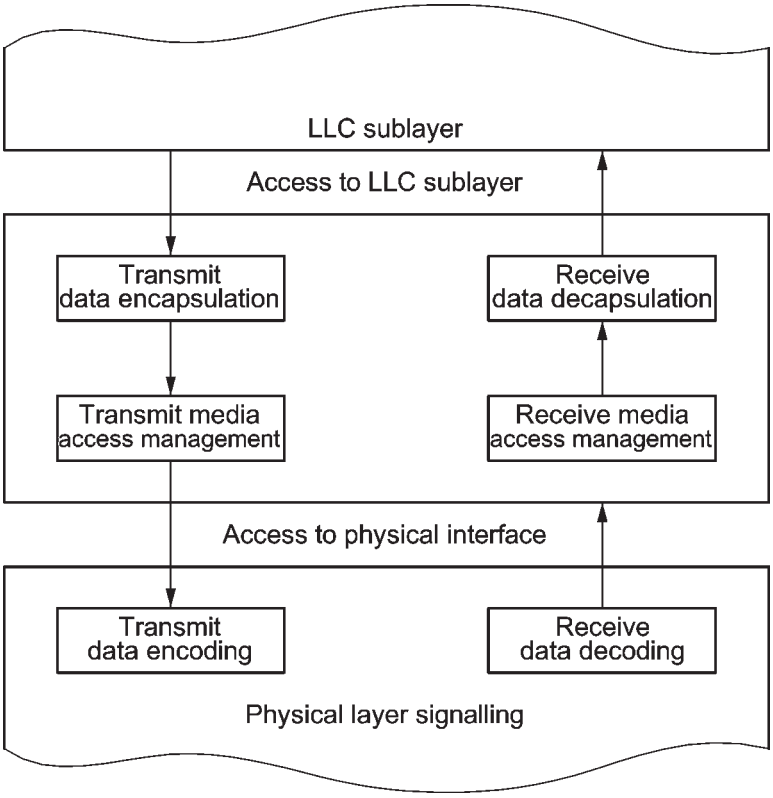


Figure 6 — MAC functions

10.3.3 Frame reception

Frame reception shall fulfil the following requirements:

a) Receive medium access management:

- reception of a serial bit stream from the PL;
- deserialization and recompiling of the frame structure;
- deletion of stuff bits (bit de-stuffing);
- error detection (CRC, stuff bit count check, format check, stuff rule check);
- transmission of ACK;
- EF construction and initiation of transmission;
- recognition of an overload condition;
- reactive OF construction and initiation of transmission.

b) Receive data de-capsulation

- removing the MAC specific information from the received frame;
- presenting the LLC frame and interface control information to the LLC sub-layer (for restricted LLC sub-layers only limited parts of the LLC frame are presented; see 8.5).

10.4 Structure of MAC frames

10.4.1 Description

Data transmission and reception between nodes in a CAN network shall be performed and controlled by four different frame types:

— a DF that carries data from a transmitter to all receivers;

— an RF transmitted by a node for requesting transmission of the DF with the same identifier;

— an EF transmitted by any node (transmitter or receiver) in case of a bus error detected;

— an OF used for providing an extra delay between the preceding and succeeding DFs or RFs.

DFs and RFs shall arbitrate for bus access and shall be separated from preceding frames by an inter-frame space.

There are four different DFs in CAN:

— DF in Classical Base Frame Format,

— DF in Classical Extended Frame Format,

— DF in FD Base Frame Format,

— DF in FD Extended Frame Format.

There are two different RFs in CAN:

— RF in Classical Base Frame Format,

— RF in Classical Extended Frame Format.

10.4.2 Specification of MAC DF

10.4.2.1 Description

On transmission, an LLC DF (see Figure 4) shall be converted into a MAC DF. On reception, a MAC DF shall be converted into an LLC DF. MAC DFs shall be composed of seven different bit fields; see also Figure 7.

— SOF;

— arbitration field (contains identifier field and part of format field);

— control field (contains DLC field and part of format field);

— data field (contains LLC data field);

— CRC field;

— ACK field;

— EOF.

Figure 7 — MAC DF

10.4.2.2 SOF

SOF shall mark the beginning of DFs and RFs. It shall consist of a single dominant bit.

A node shall send a SOF only when the bus is idle (see 10.4.6.3). A node sampling a dominant bit during its suspend transmission time or at the third bit of intermission shall accept it as SOF.

If a node samples a dominant bit at the third bit of intermission, this node shall, if it has a pending transmission and if the node is error active or has been receiver of the previous frame, start transmitting its message at the next bit with the first bit of its identifier, without first transmitting a SOF bit and without becoming receiver.

All nodes shall synchronize to the leading edge caused by SOF of the first node starting to transmit.

10.4.2.3 Arbitration field

The arbitration field shall be composed of the identifier field (passed from the LLC sub-layer) and the RTR bit (in CBFF and in CEFF) or the RRS bit (in FBFF and in FEFF). The value of the RTR bit, as well as the RRS bit, shall be dominant in a MAC DF. The structure of the arbitration field is, depending on the IDE bit and the FDF bit in the control field, different for the four formats.

— In the CBFF (where the IDE flag is dominant), the arbitration field shall consist of the 11-bit base identifier and the RTR bit. The identifier bits shall be denoted ID-28 to ID-18.

— In the CEFF (where the IDE flag is recessive), the arbitration field shall consist of the base identifier (ID-28 to ID-18), the SRR and IDE bits (both bits recessive), the identifier extension (ID-17 to ID-0), and the RTR bit.

— In the FBFF (where the IDE flag is dominant), the arbitration field shall consist of the 11-bit base identifier and the dominant RRS bit. The identifier bits shall be denoted ID-28 to ID-18.

— In the FEFF (where the IDE flag is recessive), the arbitration field shall consist of the base identifier (ID-28 to ID-18), the SRR and IDE bits (both bits recessive), the identifier extension (ID-17 to ID-0), and the dominant RRS bit.

SRR bit [only in CEFF and FEFF]

The SRR bit shall be transmitted in CEFF and in FEFF after bit ID-18, at the position of the RTR bit in CBFF or of the RRS bit in FBFF. The SRR bit shall be transmitted recessive, but receivers shall accept recessive and dominant SRR bits.

RTR bit [only in CBFF and CEFF]

The value of the RTR bit shall be dominant in a MAC DF.

RRS bit [only in FD Frames]

The RRS bit shall be transmitted in FD Frames at the position of the RTR bit in Classical Frames. The RRS bit shall be transmitted dominant, but receivers shall accept recessive and dominant RRS bits.

NOTE Receivers accepting both states for SRR and RRS bits means that neither state is treated as a form error.

IDE bit

The IDE bit shall distinguish either between CBFF and CEFF or between FBFF and FEFF, i.e. whether it belongs to

— the arbitration field for the CEFF and the FEFF, or

— the control field for the CBFF and the FBFF.

The IDE bit in the CEFF and in the FEFF shall be transmitted recessive, whereas in the CBFF and in the FBFF the IDE bit shall be transmitted dominant. It shall be transmitted after the base identifier and the

RTR bit (CBFF), after the base identifier and the SRR bit (CEFF), after the base identifier and the RRS bit (FBFF), or after the base identifier and the SRR bit (FEFF).

Arbitration shall occur over the IDE bit such that collisions between a frame in CBFF or FBFF and a frame in CEFF or FEFF, with both frames having the same base identifier, shall be resolved such that the frame in CBFF or FBFF prevails over the frame in CEFF or FEFF.

10.4.2.4 Control field

The control field shall consist of 6 bit (CBFF and CEFF) or 8 bit (FEFF) or 9 bit (FBFF), where the last four bits shall be the DLC, passed from the LLC sub-layer (see 8.4.2.4).

CBFF

In this format, the first two bits of the control field shall be the IDE bit and the FDF bit, both shall be transmitted dominant. This frame format is backwards compatible to previous versions of this part of ISO 11898.

CEFF

In this format, the first two bits of the control field shall be the FDF bit and the r0 bit, both transmitted dominant. This frame format is backwards compatible to previous versions of this part of ISO 11898.

FBFF

In this format, the first five bits shall be the IDE bit (transmitted dominant), the FDF bit (transmitted recessive), and the reserved res bit (transmitted dominant), followed by the BRS bit and the ESI bit.

FEFF

In this format, the first four bits shall be the FDF bit (transmitted recessive), and the res bit (transmitted dominant), followed by the BRS bit and the ESI bit.

FDF bit

This bit distinguishes between Classical Frames and FD Frames. It is recessive in FD Frames and it is dominant in Classical Frames. In frames with 11-bit identifiers, FDF comes after the IDE bit. In frames with 29-bit identifiers, it comes as the first bit of the control field. An FD tolerant receiver shall detect a protocol exception event (see 10.9.5) when it detects the recessive FDF bit of an FD Frame instead of the dominant FDF bit of a Classical Frame.

The FDF bit corresponds to the r0 bit in frames with 11-bit identifiers and to the r1 bit in frames with 29-bit identifiers as specified in previous versions of this part of ISO 11898. FD intolerant CAN nodes are not able to correctly decode FD Frames.

Arbitration does not occur over the FDF bit. A given identifier shall be used for a Classical Frame or for an FD Frame, but not both in a specific implementation. This does not prohibit the switching between the transmission of Classical Frames and the transmission of FD Frames by a given node.

r0 bit

In CEFF, the FDF bit shall be followed by the r0 bit, which is reserved for future expansion of the protocol. The r0 bit shall be transmitted dominant, but receivers shall accept it with recessive and with dominant state.

NOTE Receivers accepting recessive state for r0 bit means that recessive state is not treated as a form error.

res bit

In FD Frames, the FDF bit shall be followed by the res bit, which is reserved for future expansion of the protocol. The res bit shall be transmitted dominant. An FD enabled receiver shall detect a protocol exception event (see 10.9.5) when it detects the res bit to be recessive instead of the expected dominant

value. This is an implementation option. If this option is implemented, the protocol exception event detection may be disabled. When protocol exception event detection is disabled, FD enabled CAN nodes detecting the res bit following the FDF bit to be recessive shall treat this as a form error (see 10.11).

BRS bit

This bit indicates whether or not the bit rate is switched inside the FD Frame. If the bit is detected recessive, the bit rate shall be switched from the nominal bit rate of the arbitration phase to the preconfigured data bit rate of the data phase. It is not required for the BRS bit to be the same in all FD Frames in a single network. It is not required for nodes to be able to send particular LLC frames with particular BRS bit values. BRS does not exist in Classical Frames.

ESI bit

This flag shall be transmitted dominant by error active nodes and shall be transmitted recessive by error passive nodes. ESI does not exist in Classical Frames.

Optionally, error active CAN nodes may, under the control of the LLC user, transmit the ESI bit in recessive state.

10.4.2.5 Data field

The MAC frame data field shall be equivalent to the LLC data field (see 8.4.2.5).

10.4.2.6 CRC field

The CRC field shall contain the CRC sequence followed by a recessive CRC delimiter. For FD Frames, the CRC field shall also contain the stuff count.

Stuff count

In FD Frames, the stuff count shall be at the beginning of the CRC field. It shall consist of the stuff bit count modulo 8 in a 3-bit gray code followed by a parity bit as shown in Table 7.

Table 7 — Coding of the stuff count

Stuff count	Coding							
Stuff bit count modulo 8	0	1	2	3	4	5	6	7
Gray-coded with parity bit	000 0	001 1	011 0	010 1	110 0	111 1	101 0	100 1

Both transmitter and receivers of a frame shall count the number of stuff bits before the first fixed stuff bit in the frame. The transmitter shall transmit its stuff bit count, coded as stuff count, at the beginning of the CRC field, before the CRC sequence. Receivers shall check whether the received stuff count matches with the value calculated from their own stuff bit count.

CRC sequence

The frame check sequence shall be derived from a CRC (BCH-code).

A CAN node shall use different CRC generator-polynomials for different frame formats. The first polynomial, CRC_15, is used for Classical Frames. The second, CRC_17, is used for FD Frames with a data field up to 16 byte long. The third, CRC_21, is used for FD Frames with a data field longer than 16 byte. Each polynomial results in a Hamming Distance of six.

At the start of the frame, all three CRC sequences shall be calculated concurrently; in all nodes including the transmitter. The node that wins the arbitration sends the CRC sequence selected by the values of the frame's FDF bit and DLC. The receivers shall consider only the selected CRC polynomial to check for a CRC-error.

The length of the CRC sequence (n_{CRC}, the order of the generator-polynomial) is set to 15 for CRC_15, to 17 for CRC_17, and to 21 for CRC_21.

CRC_15	$C599_{16}$	$(x^{15}+x^{14}+x^{10}+x^{8}+x^{7}+x^{4}+x^{3}+1)$
	=	$(x+1) \times (x^{7}+x^{3}+1) \times (x^{7}+x^{3}+x^{2}+x^{1}+1)$
CRC_17	$3685B_{16}$	$(x^{17}+x^{16}+x^{14}+x^{13}+x^{11}+x^{6}+x^{4}+x^{3}+x^{1}+1)$
	=	$(x+1) \times (x^{16}+x^{13}+x^{10}+x^{9}+x^{8}+x^{7}+x^{6}+x^{3}+1)$
CRC_21	302899_{16}	$(x^{21}+x^{20}+x^{13}+x^{11}+x^{7}+x^{4}+x^{3}+1)$
	=	$(x+1) \times (x^{10}+x^{3}+1) \times (x^{10}+x^{3}+x^{2}+x^{1}+1)$

There are different conventions how to represent a generator-polynomial in hexadecimal notation. The notation in this part of ISO 11898 includes the high-order bit.

The CRC_INIT_VECTOR shall be (0,....,0) for CRC_15 and shall be (1,0,...,0) for CRC_17 and for CRC_21; where the single “1” is at the most significant bit position.

The relevant bit stream for CRC calculation is the bit stream consisting of SOF, arbitration field, control field, and (if present) data field, supplemented with n_{CRC} bits of “0”. In Classical Frames, stuff bits shall not be included in the relevant bit stream for CRC calculation. In FD Frames, the stuff count and the stuff bits, with the exception of the fixed stuff bits, shall be included in the relevant bit stream for CRC calculation.

In order to carry out the CRC calculation, the polynomial to be divided is defined by the coefficients of the relevant bit stream. This polynomial is divided (the coefficients are calculated modulo-2) by the generator-polynomial.

The remainder of this polynomial division is the CRC sequence transmitted over the bus. In order to implement this function, a n_{CRC} bit shift register CRC_RG(n_{CRC} –1:0) can be used. Each CRC sequence is calculated in a separate shift register block. If NXTBIT denotes the next bit of the bit stream, given by the relevant bit stream from SOF until the end of the data field, the CRC sequences are calculated as follows:

```
CRC_RG(nCRC -1: 0) = CRC_INIT_VECTOR;   //initialize shift register
REPEAT
     CRCNXT = NXTBIT EXOR CRC_RG(nCRC -1);
     CRC_RG(nCRC -1: 1) = CRC_RG(nCRC -2: 0);   //shift left by one position
     CRC_RG(0) = 0;
     IF CRCNXT THEN
          CRC_RG(nCRC -1: 0) = CRC_RG(nCRC -1:0) EXOR (CRC polynomial);
     ENDIF
UNTIL (NXTBIT = [End of bit stream] or there is an error condition).
```

After the transmission / reception of the last bit of the relevant bit stream, each CRC_RG contains one of the three CRC SEQUENCES.

CRC delimiter

The CRC sequence shall be followed by the CRC delimiter. In Classical Frames, the CRC delimiter is one single recessive bit. In FD Frames, the CRC delimiter may consist of one or two recessive bits. A transmitter shall send only one recessive bit as CRC delimiter, but it shall accept two recessive bits before the edge from recessive to dominant that starts the acknowledge slot. A receiver will send its acknowledge bit after the first CRC delimiter bit.

NOTE CAN implementations switch back from the data phase to the arbitration phase of FD Frames when they reach the sample point of the (first bit of the) CRC delimiter.

10.4.2.7 ACK field

The ACK field shall contain the ACK slot and the ACK delimiter. In the ACK field, the transmitter node shall send recessive bits. All receivers check the consistency of the received DF or RF and shall acknowledge a consistent frame and shall flag an inconsistent frame by means of an EF (see 10.12). A DF

or RF that is not acknowledged shall be regarded as corrupted and shall be flagged by the transmitting node with an EF.

ACK slot

All nodes that have received the matching CRC sequence (and, in FD Frames the matching stuff count) shall send an ACK within the ACK slot by overwriting the recessive bit of the transmitter by a dominant bit (they send ACK). In FD Frames, all nodes shall accept an up to two bit long dominant phase of overlapping ACK slot bits as a valid ACK, to compensate for phase shifts between the receivers. In Classical Frames, a dominant bit following the single ACK slot bit shall be regarded as a form error.

ACK delimiter

The ACK delimiter, being the last bit of the ACK field, shall be a recessive bit. As a consequence, the ACK slot is surrounded by recessive bits (CRC delimiter, ACK delimiter).

10.4.2.8 EOF

Each DF and RF shall be delimited by a flag sequence consisting of seven recessive bits forming the EOF.

10.4.3 Specification of MAC RF

10.4.3.1 Description

A node acting as a receiver for certain data may initiate the transmission of the respective data by its source node by sending an RF as given in Figure 8:

Figure 8 — MAC RF

10.4.3.2 Identical fields of MAC DF and of MAC RF

The bit fields SOF, CRC field, ACK field and EOF shall be equivalent to the corresponding bit fields of the MAC DF (see Figure 7). There shall be no data field in the RF.

10.4.3.3 Arbitration field

Arbitration field shall be composed of the identifier field, passed from the LLC sub-layer, and the RTR bit. In CBFF and in CEFF, the value of the RTR bit in a MAC RF shall be recessive. There is no RF in FBFF or FEFF.

10.4.3.4 Control field

In CBFF and in CEFF, the control field of the MAC RF shall be equivalent to the control field of the MAC DF (see 8.4.2.4). The collision resolution (see 10.9.9) requires that the value of an RF's DLC equals the DLC of the requested DF.

10.4.4 Specification of EF

10.4.4.1 Description

The EF shall consist of two different fields. The first field shall be as given by the superposition of error flags contributed from different nodes. The second field shall be the error delimiter.

10.4.4.2 Error flag

Two forms of error flag may be used, the active error flag and the passive error flag, where

— the active error flag shall consist of 6 consecutive dominant bits, and

— the passive error flag shall consist of 6 consecutive recessive bits unless it is overwritten by dominant bits from other nodes.

An error-active node detecting an error condition shall signal this by sending an active error flag. The form of the error flag violates the rule of bit stuffing or destroys the bit field requiring fixed form. As a consequence, all other nodes shall detect an error condition too, and shall start sending an error flag. So the sequence of dominant bits, which may actually be monitored on the bus, results from a superposition of different error flags sent by individual nodes. The total length of this sequence may vary between a minimum of 6 bit and a maximum of 12 bit.

Passive error flags initiated by a transmitter shall cause error(s) (with two exceptions) at the receiver(s) when they start in a frame field encoded by the method of bit stuffing, because then they lead to stuff errors detected by the receivers. The first exception is a passive error flag that starts during arbitration and another node continues transmitting, and the second exception is a passive error flag that starts less than 6 bit before the end of the CRC sequence and the last bits of the CRC sequence happen to be all recessive.

Passive error flags initiated by receivers shall not be able to prevail over any activity on the bus. Therefore, error-passive receivers shall always wait for 6 subsequent equal bits after detecting an error condition. The passive error flag is complete when these 6 equal bits have been detected.

10.4.4.3 Error delimiter

The error delimiter shall consist of 8 recessive bits. After sending an error flag, each node shall send recessive bits and monitor the bus until it detects a recessive bit. Afterwards, it shall start sending 7 more recessive bits.

10.4.5 Specification of OF

10.4.5.1 Types

Following types of OF shall have the same format.

— LLC requested OF

  This OF is requested by the LLC sub-layer to indicate an internal overload situation (see 10.13).

— Reactive OF

  The transmission of the reactive OF shall be initiated by the MAC sub-layer upon certain error conditions (see 10.13).

The OF shall contain two bit fields, overload flag and overload delimiter. The overload flag shall correspond to that of the active error flag. The overload delimiter shall be the same as the error delimiter.

10.4.5.2 Overload flag

The overload flag shall consist of 6 dominant bits. It destroys the fixed form of the intermission field (see 10.4.6). As a consequence, all other nodes also detect an overload condition and shall start sending an overload flag.

10.4.5.3 Overload delimiter

The overload delimiter shall consist of 8 recessive bits. After sending an overload flag, each node shall monitor the bus until it detects a recessive bit. At this point in time, every node shall finish sending its overload flag and all nodes shall start sending 7 more recessive bits simultaneously, to complete the 8-bit-long overload delimiter.

10.4.6 Specification of inter-frame space

10.4.6.1 Description

DFs and RFs shall be separated from preceding frames, whatever frame type they are (DF, RF, EF, OF), by a time period called inter-frame space. In contrast to this, EFs and OFs shall not be preceded by inter-frame space, and multiple OFs shall not be separated by inter-frame space.

Inter-frame space shall contain the bit field intermission and bus idle time. For error-passive nodes, which have been transmitter of the previous frame, inter-frame space shall also contain the node's suspend transmission time (see Figure 9 and Figure 10).

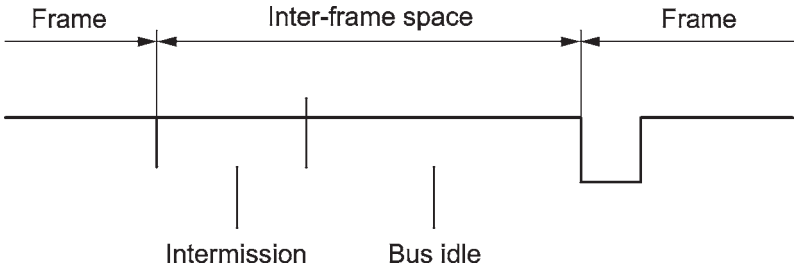


Figure 9 — Inter-frame space for nodes, which are not error-passive or have been receiver of previous frame

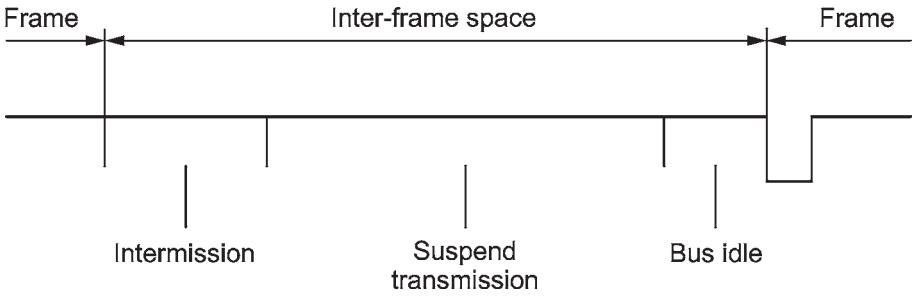


Figure 10 — Inter-frame space for error-passive nodes, which have been transmitter of previous frame

10.4.6.2 Intermission

The intermission field shall consist of 3 recessive bits. During intermission no node shall start transmission of a DF or RF. Only signalling of overload condition is allowed.

The detection of a dominant bit at the third bit of intermission shall be interpreted as SOF (see 10.4.2.2).

10.4.6.3 Bus idle

The period of bus idle may be of arbitrary length. The bus shall be recognized to be idle by receivers and by error active transmitters when the third bit of intermission is seen recessive; by error passive transmitters when the eighth bit of suspend transmission time is seen recessive; or when the bus integrating state is left (see 10.9.4). When the bus is idle, any node may access the bus for transmission.

A frame pending for transmission during the transmission of another frame shall be started in the first bit following intermission.

The detection of a dominant bit on the bus during bus idle time shall be interpreted as SOF.

10.4.6.4 Suspend transmission

An error-passive node, which has been transmitter of the previous frame, shall suspend the start of further frame transmissions for 8 bit times following intermission. If another node starts a transmission during that suspend transmission time, the node shall become a receiver of this DF or RF frame.

10.5 Frame coding

The bit stream in a frame shall be coded according to the NRZ method. This means that the generated bit level is constant during the total bit time.

In order to limit the maximum distance between edges available for synchronization, the frame segments as SOF, arbitration field, control field, data field, and CRC sequence shall be coded by the method of bit stuffing. Whenever a transmitter detects five consecutive bits (including stuff bits) of identical value in the bit stream to be transmitted, it shall automatically insert a complementary bit (called stuff bit) into the actual transmitted bit stream (see Figure 11). The receiver shall recognize a sequence of five consecutive bits of identical value and shall discard the following stuff bit.

The bit stuffing is shown in Figure 11.

Destuffed bit stream	01011111010	10100000101	01011111000010	10100000111101
Stuffed bit stream	01011111o010	10100000i101	01011111o0000i10	10100000i1111o01
"0", "o" = dominant (stuff) bit; "1", "i" = recessive (stuff) bit				

Figure 11 — Bit stuffing

In the CRC field of FD Frames, the stuff bits shall be inserted at fixed positions; they are called fixed stuff bits. There shall be a fixed stuff bit before the first bit of the stuff count, even if the last bits of the preceding field are not a sequence of five consecutive bits of identical value. If the last bits of the preceding field are a sequence of five consecutive bits of identical value, there shall be only the fixed stuff bit, there shall not be two consecutive stuff bits. A further fixed stuff bit shall be inserted after each fourth bit of the CRC field. The value of such a fixed stuff bit shall be the inverse value of the bit preceding the fixed stuff bit. A receiver shall discard the fixed stuff bits from the bit stream for the CRC check. It shall detect a form error if the fixed stuff bit has the same value as its preceding bit. The number of fixed stuff bits in the CRC field of FD Frames is equal to the maximum number of stuff bits that would result from applying the bit stuffing method of the Classical Frames.

The remaining bit fields of the DF or RF (CRC delimiter, ACK field and EOF) shall be of fixed form and not stuffed.

The EF and the OF shall be of fixed form as well and shall not be coded by the method of bit stuffing.

10.6 Frame acknowledgement

All receivers shall check the consistency of all received DFs and RFs and they shall acknowledge all consistent DFs and RFs. Acknowledgement shall not depend on the frame's identifier.

10.7 Frame validation

The point in time at which a frame is taken to be valid shall be the same for all receivers of the frame but different for the transmitter of the frame.

Receiver

The frame shall be valid for receivers, if there is no error until the last but one bit of EOF. The value of the last bit of EOF shall not inhibit frame validation and a dominant value shall not lead to a form error. A receiver that detects a dominant bit at the last bit of EOF shall respond with an OF (see 10.13).

Transmitter

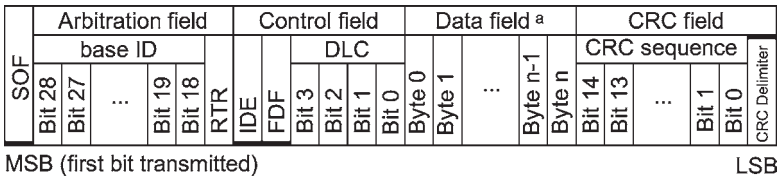
The frame shall be valid for a transmitter, if there is no error until the end of EOF. If a frame is corrupted, recovery shall be processed as described in 10.9.6.

NOTE If the transmitter samples a dominant bit at the last bit of EOF (a global or a local error at the sender), then the frame is valid for the receiver, but not for the transmitter. The transmitter places an EF and restart the transmission of the frame and the frame will be received twice. The receiver treats the dominant bits as an OF and receives the next frame as an independent second frame.

10.8 Order of bit transmission

DFs and RFs shall be transferred bit field by bit field, starting with the dominant SOF bit. Within a field, the MSB shall be transmitted first. Within the data field (if any), the bytes are transferred from byte 0 to n (n + 1 is the number of data bytes as defined in Table 5). Within each byte, the bits are transferred from bit 7 down to bit 0.

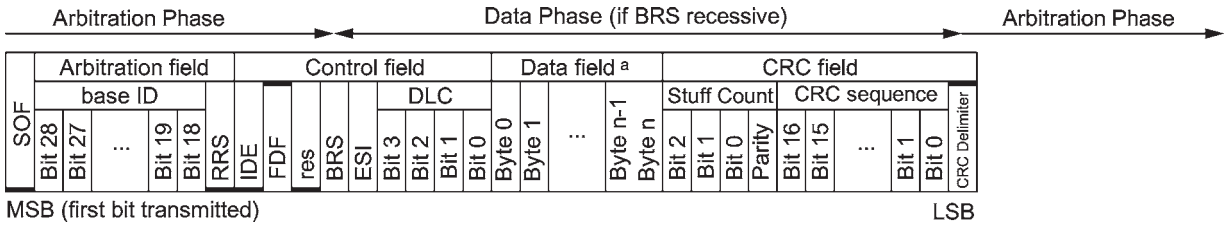
In Figure 12 to Figure 17, a broad line at the bottom of a bit indicates that the bit shall be transmitted dominant, a broad line at the top of a bit indicates that the bit shall be transmitted recessive. Stuff bits are not shown in the figures.



Key

a No data field if DLC = 0 or RTR = recessive.

Figure 12 — Order of bit transmission in Classical Base Frame Format, up to 8 data bytes



Key

a No data field if DLC = 0.

Figure 13 — Order of bit transmission in FD Base Frame Format, up to 16 data bytes

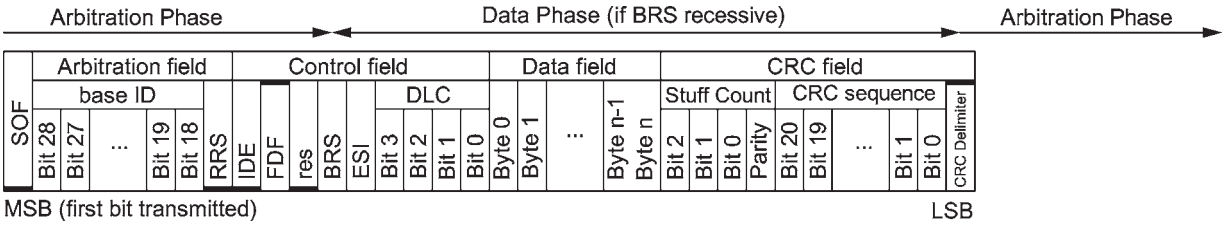
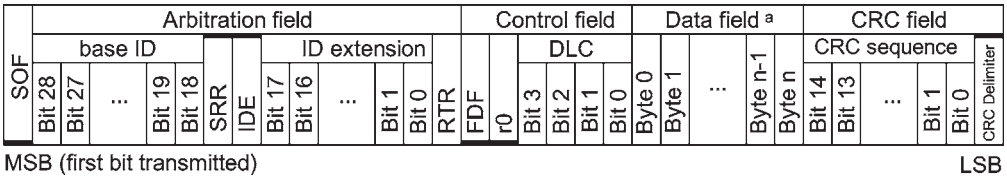


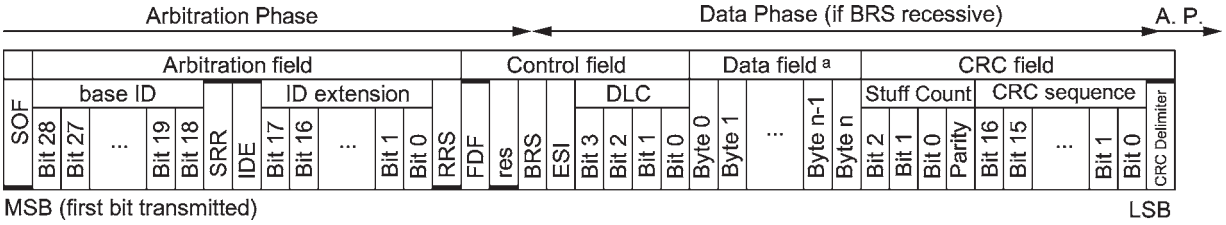
Figure 14 — Order of bit transmission in FD Base Frame Format, 20 to 64 data bytes



Key

a No data field if DLC = 0 or RTR = recessive.

Figure 15 — Order of bit transmission in Classical Extended Frame Format, up to 8 data bytes



Key

a No data field if DLC = 0.

Figure 16 — Order of bit transmission in FD Extended Frame Format, up to 16 data bytes

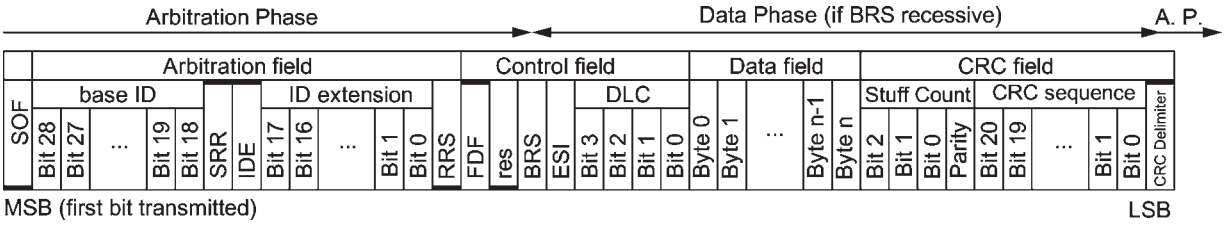


Figure 17 — Order of bit transmission in FD Extended Frame Format, 20 to 64 data bytes

10.9 Medium access method

10.9.1 General

This Clause describes the functions and characteristics related to the medium access method of CAN.

10.9.2 Multi-master

Every node transmitting a DF or an RF shall be the bus master during that transmission.

10.9.3 Bus access

An error-active node may access the bus as soon as the bus is idle (see 10.4.6.3). An error-passive node, which is the receiver of the current or previous frame, may access the bus as soon as the bus is idle. An error-passive node, which is transmitter of the current frame or has been transmitter of the previous frame, may access the bus as soon as its suspend transmission time is finished, provided that no other node has started transmission meanwhile. Whenever several nodes start transmitting in coincidence, that node transmitting the frame with the highest priority at this time shall become the bus master. The mechanism to resolve the resulting bus access conflict shall be content-based arbitration.

10.9.4 Bus integration state

CAN nodes shall enter the bus integration state after starting the protocol operation, during bus-off recovery, or (for nodes that are FD tolerant or FD enabled) after detecting the protocol exception state. CAN nodes shall leave the bus integration state when the idle condition (see 4.28) is detected.

There shall be a bit counter that is reset when the bus integration state is entered or when the CAN bus is detected dominant at the sample point. The bit counter shall be incremented when the CAN bus is detected recessive at the sample point. The idle condition shall be detected when this bit counter reaches the value 11. For the detection of the bus-off recovery condition (see 12.1.4.4), there shall be a second counter that is incremented once each time the idle condition is detected.

Nodes that are in bus-off state shall re-enter the bus integration state immediately after detecting the idle condition if the bus-off recovery condition is not yet met.

For nodes that are FD tolerant or FD enabled, there shall be a third reset condition for the bit counter. It shall be reset when detecting an edge that causes synchronization. When synchronization occurs, the counting of the sequence of 11 consecutive recessive bits shall be restarted. For an optional edge filtering see 11.3.2.3.

NOTE When an FD tolerant or FD enabled node is integrating, it reacts on dominant pulses that are shorter than a nominal bit time to ensure the bits in the data phase of an FD Frame are not mistaken for an idle condition.

10.9.5 Protocol exception event

FD tolerant CAN nodes and FD enabled CAN nodes detect a protocol exception event as it is specified in 10.4.2.4. As reaction to the protocol exception event, the error counters shall not be changed, the hard synchronization shall be enabled, the node shall send recessive bits and shall enter the bus integration state (see 10.9.4).

10.9.6 Transmission of MAC frames

MAC DFs and MAC RFs may be started when the node is allowed to access the bus according to 10.9.3. A MAC EF shall be transmitted as specified in 10.12. A MAC OF shall be transmitted as specified in 10.13.

MAC DFs and MAC RFs that have lost arbitration, that have not been acknowledged, or that have been disturbed by errors during transmission shall be retransmitted automatically. A MAC DF or MAC RF that is retransmitted shall be handled as any other MAC DF or MAC RF, i.e. it participates in the arbitration process to gain bus access. The automatic retransmission of a frame shall be disabled when the transmission of that frame is no longer requested. Optionally, there may be a configuration of the CAN node to limit the retransmission attempts to a specific number. The automatic retransmission may be disabled for all frames.

10.9.7 Content-based arbitration

During arbitration, every transmitter shall compare the level of the bit transmitted with the level monitored on the bus. If these levels are equal, the node may continue to send. When a recessive level is sent and a dominant level is monitored, the node lost arbitration and shall withdraw without

sending any more bits. When a dominant level is sent and a recessive level is monitored, the node shall detect a bit error.

Content-based arbitration shall be performed from the first bit of the identifier up to the IDE bit in the case of a base format message and up to the bit following the identifier, which is the RTR bit or the RRS bit in FD Frames, of an extended format message.

10.9.8 Frame priority

Among two frames with different identifiers, the higher priority shall be assigned to the frame containing the identifier of lower binary value.

If a DF and an RF with the same identifier are initiated at the same time, the DF shall have higher priority than the RF. This shall be achieved by assigning according values to the RTR bit.

10.9.9 Collision resolution

Transmissions may only be initiated when the bus is idle. When two or more frames started at the same time, this is called a collision. The CAN bit-wise arbitration method resolves all collisions between DFs and RFs that have different identifiers or different frame types. Unresolved collisions cause EFs, if the colliding frames are not identical.

10.9.10 Disabling of frame formats

Optionally, an implementation may have a configuration interface that allows the disabling of either the Classical CAN frame format or the Flexible Data Rate frame format. If a frame format is disabled, frames in that format shall be treated as invalid frames, causing error frames. An FD enabled implementation shall not be set into a mode where it behaves as an FD tolerant implementation.

10.10 MAC data consistency

Messages to be transmitted are prepared by the LLC user and are transferred via the node's controller-host interface and LLC sub-layer of the Data Link Layer to the MAC sub-layer that is responsible for message framing. Messages may be stored in a shared memory. Data consistency of transmitted messages from a shared memory shall be ensured by at least one of two methods.

— The MAC sub-layer shall store the whole message to be transmitted in a temporary buffer that is filled before the transmission is started.

— The LLC sub-layer shall check for data errors while the message to be transmitted is transferred to the MAC sub-layer. If a data error is detected, the transmission shall not be started. If it is already started when the data error is detected, the node shall be switched into bus monitoring mode, see 10.14, or into restricted operation mode, see 10.15. Receiving nodes will not see a valid message.

NOTE Data errors are, e.g. parity errors in a RAM word, data not provided in time, or data partially updated by the LLC user while a transmission is in progress. Implementation of the operation modes bus monitoring mode and restricted operation mode is optional if the first method to ensure data consistency is used.

10.11 Error detection

The MAC sub-layer shall provide the following mechanisms for error detection:

— monitoring;

— stuff rule check;

— frame check;

— stuff count check in FD Frames,

— 15-bit, 17-bit, or 21-bit CRC;

— ACK check.

There are five different error types, which are not mutually exclusive:

a) **Bit error**

A node sending a bit on the bus shall also monitor the bus. A bit error is detected at that bit time, when the bit value that is monitored differs from the bit value sent.

Exceptions: a dominant bit shall not lead to a bit error when a recessive information bit is sent during arbitration, or a recessive bit is sent during ACK slot. A node sending a passive error flag and detecting a dominant bit shall not interpret this as a bit error.

b) **Stuff error**

A stuff error shall be detected at the bit time of the sixth consecutive equal bit level in a frame field that is coded by the method of bit stuffing. It shall be regarded as a form error, not as a stuff error, when a fixed stuff bit in the CRC field of an FD Frame is not at its expected value.

c) **CRC error**

The CRC sequence shall consist of the result of the CRC calculation of the transmitter. The receivers shall calculate the CRC in the same way as the transmitter. A CRC error shall be detected when the calculated CRC sequence does not equal the received one. In FD Frames, a mismatch between the counted stuff bits and the received stuff count shall be treated as a CRC error.

d) **Form error**

A form error shall be detected when a fixed-form bit field contains one or more illegal bits.

Exception: A receiver monitoring a dominant bit at the last bit of EOF, or any node monitoring a dominant bit at the last bit of error delimiter or of overload delimiter, shall not interpret this as a form error.

e) **ACK error**

An ACK error shall be detected by a transmitter whenever it does not monitor a dominant bit during ACK slot.

Whenever one of these errors is detected, the LLC sub-layer shall be informed. As a consequence, the MAC sub-layer shall initiate the transmission of an error flag.

10.12 Error signalling

Whenever a bit error, stuff error, form error, or ACK error is detected by any node, an error flag shall be started by the respective node at the next bit. When an error is detected in the data phase of an FD Frame, the node shall switch from the data bit time back into the nominal bit time of the arbitration phase before it starts the error flag. A transmitter that uses TDC (see 11.3.3) shall switch the bit time as shown in Figure 18. A receiver (and a transmitter not using TDC) shall switch the bit time as shown in Figure 19. The shadings in Figure 18 and Figure 19 represent the parts of the bit time as shown in Figure 22.

After detection of the error at the SSP, then, after SP and IPT, the bit timing is set back to nominal bit rate, but the bits during IPT are already counted for the Phase_Seg2, which must pass, before the next bit is started and the error flag sending is started.

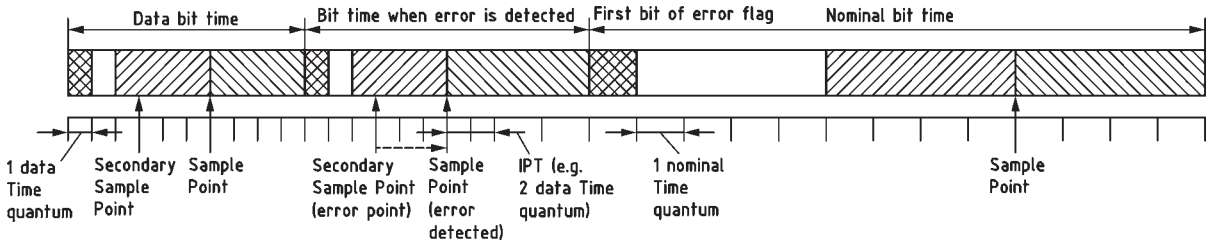


Figure 18 — Transmitter detects bit error at SSP in data phase of an FD Frame

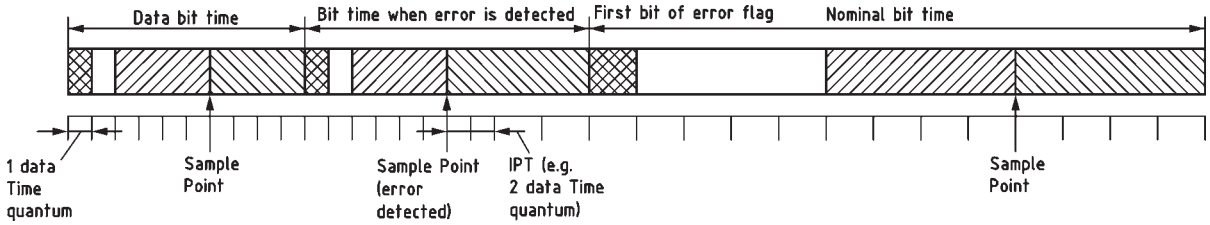


Figure 19 — Receiver detects error in data phase of an FD Frame

A receiver receiving a Classical Frame and detecting a CRC error shall send an EF after the ACK delimiter. A receiver receiving an FD Frame and detecting a CRC error shall send an EF after three bit-times following the CRC Delimiter. This is shown in Figure 20 for Classical Frames and in Figure 21 for FD Frames. Dominant bits seen between the CRC delimiter and the start of the EF shall not be treated as errors.

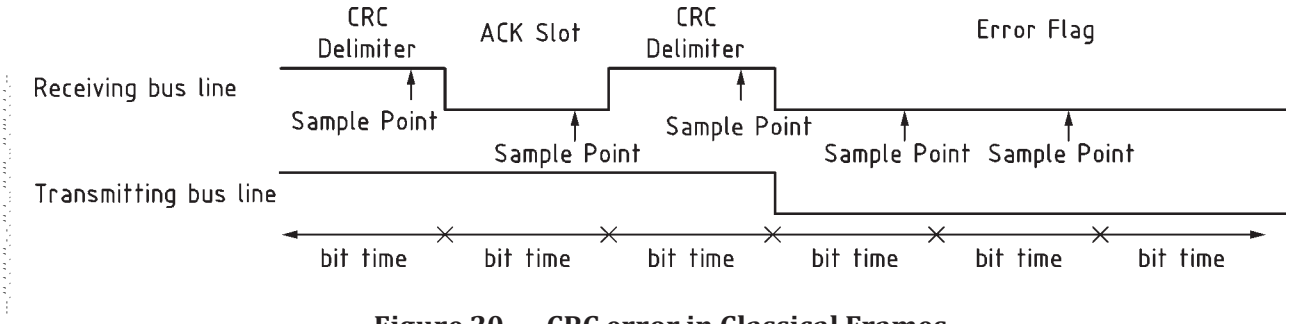


Figure 20 — CRC error in Classical Frames

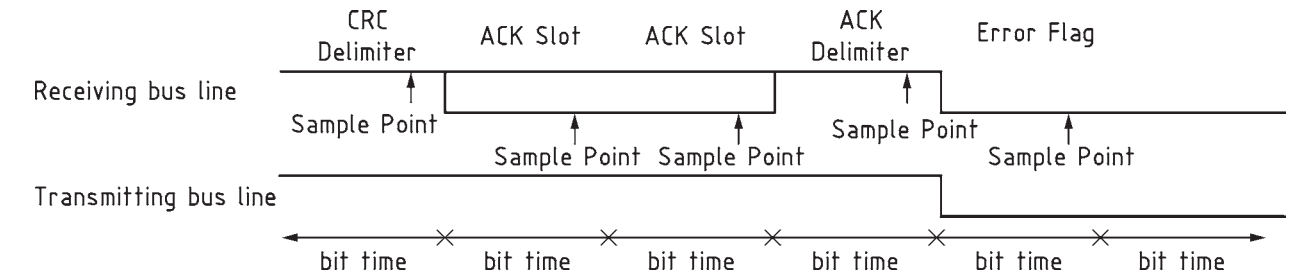


Figure 21 — CRC error in FD Frames

10.13 Overload signalling

The following conditions shall lead to the transmission of an OF.

a) LLC-requested OF (initiated by the LLC sub-layer): Internal conditions of a receiver, which require a delay of the next MAC DF or MAC RF.

b) Reactive OF (initiated by the MAC sub-layer): Detection of a dominant bit during either of the first two bits of intermission, detection of one dominant bit at the last bit of EOF by a receiver, or detection of one dominant bit by any node at the last bit of error delimiter or overload delimiter.

An LLC-requested OF shall only be started at the first bit of an expected intermission, whereas reactive OFs shall start one bit after detecting the dominant bit due to condition b) above. The start of LLC-requested OFs due to condition a) above shall be allowed, but are not required to be implemented.

At most, two LLC OFs shall be generated to delay the next MAC DF or MAC RF.

10.14 Bus monitoring

Optionally, CAN implementations may provide the bus monitoring mode, where they shall be able to receive valid DFs and valid RFs, but it sends only recessive bits on the CAN network and does not start a transmission. If the MAC sub-layer is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit shall be rerouted internally so that the MAC sub-layer monitors this dominant bit, although the CAN network may remain in recessive state.

10.15 Restricted operation

Optionally, CAN implementations may provide the restricted operation mode, where they shall be able to receive DFs and RFs and shall give ACK to valid frames, but it shall not send EFs or OFs. In case of an error condition or overload condition, it shall not send dominant bits; instead it shall treat this as a protocol exception event and shall enter the bus integration state (see 10.9.4). The error counters shall not be incremented or decremented while the CAN node is in restricted operation mode. If the CAN node in restricted operation mode is a potential time master in a network, it shall be able to transmit time reference messages to start up the network; other frames shall not be transmitted.

11 PL specification

11.1 General and functional modelling

The PL implementation connects a CAN node to the bus-lines. The number of nodes is limited by the electric loads on the bus-lines and by the CAN data link layer protocol.

The PL is modelled according to ISO/IEC 8802-3. It has three sub-layers.

a) The PCS shall encompass functions related to bit encoding/decoding, timing, and synchronization, as well as bus failure detection. It is specified in the subsequent subclauses.

b) The PMA sub-layer encompasses functional circuitry for bus-line transmission/reception. It is not in the scope of this part of ISO 11898.

c) The PMD sub-layer encompasses the mechanical and electrical interfaces between the physical media and the PMA sub-layer. It is not in the scope of this part of ISO 11898.

11.2 Services of PL

11.2.1 Description

The services of PL shall allow the local MAC sub-layer entity to exchange bits with peer MAC sub-layer entities.

The PL shall provide the following service primitives to the MAC sub-layer:

— PCS_Data.Request;

— PCS_Data.Indicate.

For FD enabled implementation, there shall be two additional optional service primitives:

— PCS_Status.Transmitter;

— PCS_Status.Receiver.

11.2.2 PCS_Data.Request

The PCS_Data.Request primitive shall be passed from the MAC sub-layer to the PL to request transmission of a dominant or recessive bit. The primitive provides the following parameter:

```
PCS_Data.Request(
     Output_Unit
     )
```

The Output_Unit parameter shall take on one of the two values: dominant or recessive.

11.2.3 PCS_Data.Indicate

The PCS_Data.Indicate primitive shall be passed from the PL to the MAC sub-layer in order to indicate the arrival of a dominant or recessive bit. The primitive shall provide the following parameter.

```
PCS_Data.Indicate(
     Input_Unit
     )
```

The Input_Unit parameter shall take on one of the two values each representing a single bit: dominant or recessive.

11.2.4 PCS_Status.Transmitter

The PCS_Status.Transmitter primitive shall be passed from the MAC sub-layer to the PL to indicate that the MAC sub-layer transmits the data phase of an FD frame. The primitive shall provide the following parameter.

```
PCS_Status.Transmitter(
     FD_Transmit
     )
```

The FD_Transmit parameter shall take on one of the two values: active when the MAC sub-layer transmits the data phase of an FD frame and passive when the MAC sub-layer does not transmit the data phase of an FD frame.

11.2.5 PCS_Status.Receiver

The PCS_Status.Receiver primitive shall be passed from the MAC sub-layer to the PL to indicate that the MAC sub-layer receives the data phase of an FD frame. The primitive shall provide the following parameter.

```
PCS_Status.Receiver(
     FD_Receive
     )
```

The FD_Receive parameter shall take on one of the two values: active when the MAC sub-layer receives the data phase of an FD frame and passive when the MAC sub-layer does not receive the data phase of an FD frame.

11.3 PCS specification

11.3.1 Bit encoding/decoding

11.3.1.1 Bit time

Bus management functions executed within the bit time frame, such as CAN node synchronization behaviour, network transmission delay compensation, and sample point positioning, shall be as given by the programmable bit timing logic of the CAN implementation.

FD enabled implementations shall support two bit rates, the nominal bit rate and the data bit rate. Implementations that are not FD enabled are limited to the nominal bit rate as specified for Classical CAN. The definition for the second bit rate, the data bit rate with the data bit time, requires a separate configuration register set. The data bit time shall have the same length as the nominal bit time or it shall be shorter than the nominal bit time.

The data bit time shall only be used inside the data phase of an FD Frame. The data phase shall start at the sample point of the BRS bit if that bit is detected to be recessive. This data phase shall be finished when the first sample point of the CRC Delimiter is reached or when the CAN implementation sees an error condition that results in the starting of an EF. The phase outside the data phase shall be the arbitration phase. All Classical Frames, EFs, OFs, idle time, and all parts of FD Frames with a dominant BRS bit shall belong to the arbitration phase. The nominal bit time shall be used inside the arbitration phase. The bit rate shall be switched from nominal bit rate to data bit rate at the sample point of the recessive BRS bit. The bit rate shall be switched from data bit rate to nominal bit rate at the first sample point of the CRC Delimiter or when an error condition is detected. When the bit rate is switched because an error condition is detected, the switching time shall be shifted after the sample point, by less than or equal to two time quanta; see Figure 18 and Figure 19.

NOTE Since the bit rate is switched at the sample points of the BRS bit and of the CRC Delimiter bit, the lengths of these two bits are intermediate. The sum of the length of these two bits is the same as the sum of one bit of the nominal bit time and one bit of the data bit time; see Figure 22.

Time quantum

The time quantum shall be a fixed unit of time derived from the node clock period. There shall exist at least one programmable prescaler, with integral values, ranging at least from 1 to 32. The minimum time quantum is one node clock period long. The time quantum shall have a length of

— time quantum(N) = $m(N)$ · minimum time quantum for the nominal bit time, and

— time quantum(D) = $m(D)$ · minimum time quantum for the data bit time,

where $m(N)$ and $m(D)$ are values of the prescaler.

There shall be two implementations options, either

— two separate prescalers with $m(N)$ for the nominal bit time and $m(D)$ for the data bit time, or

— a shared prescaler with $m(N) = m(D)$.

Two separate prescalers enable different lengths for time quantum(N) and for time quantum(D). With a shared prescaler, time quantum(N) has the same length as time quantum(D). The length of a bit time depends on the length of the time quantum and on the number of time quanta in the bit. If different parameter combinations come to the same length of the bit time, the combination with the shorter time quantum shall be used.

Implementations that are not FD enabled are limited to the nominal bit time based on time quantum(N).

Bit rates and bit times

The bit rate gives the number of bits per second transmitted in the absence of resynchronization by an ideal transmitter; the relations between bit rates and bit times are

$$\text{nominal bit time} = \frac{1}{\text{nominal bit rate}} \text{ and data bit time} = \frac{1}{\text{data bit rate}}$$

Both bit times shall consist of separate non-overlapping time segments. The segments shall form the bit times as shown in Figure 22.

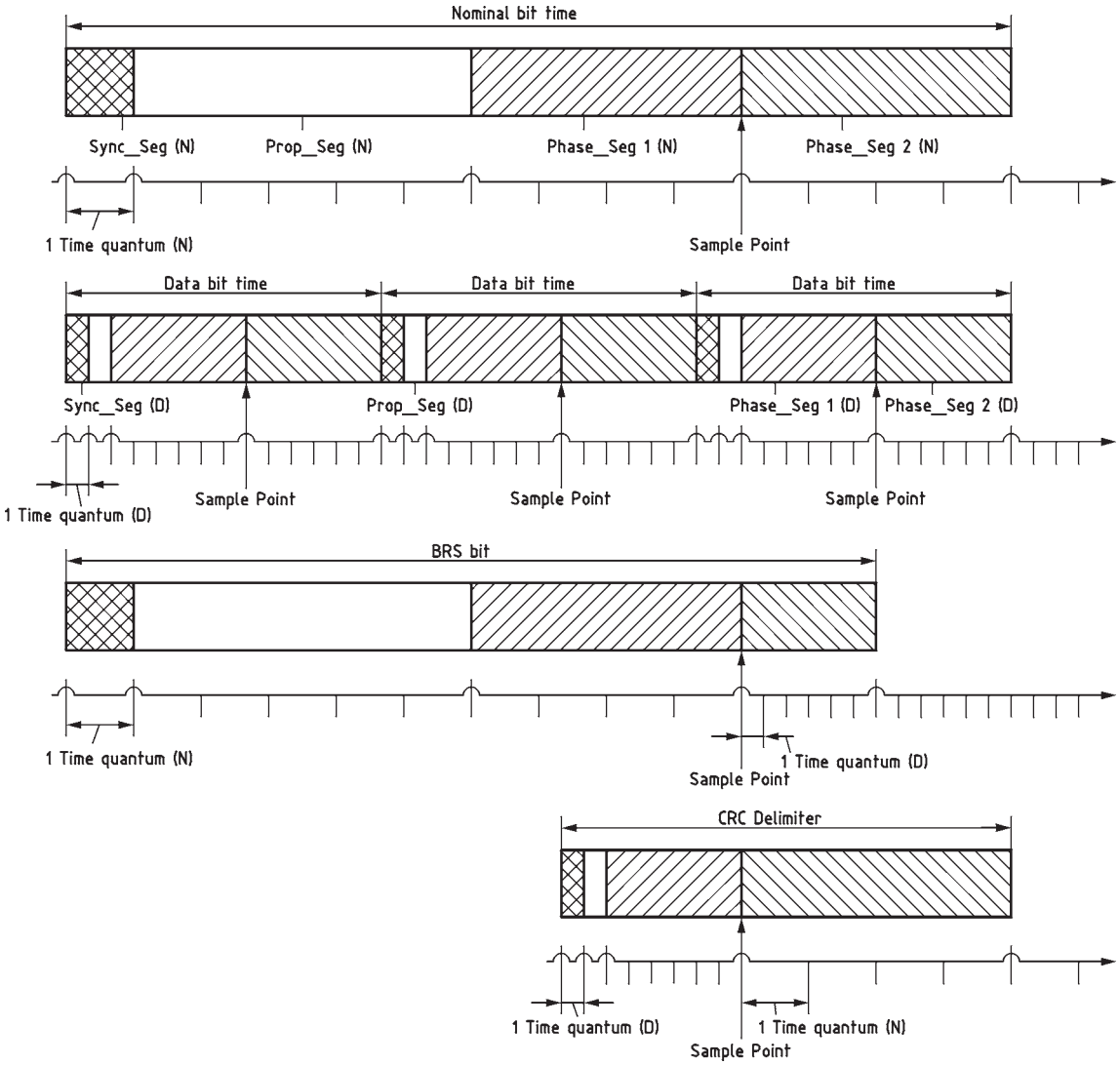


Figure 22 — Segments of nominal bit time and of data bit time

Sync_Seg

This part of the bit time, the synchronization segment, shall be used to synchronize the various CAN nodes on the bus. An edge is expected to be detected within this segment.

Prop_Seg

This part of the bit time, the propagation time segment, shall be used to compensate for physical delay times within the network. These delay times consist of the signal propagation time on the bus and the internal delay time of the CAN nodes; see Figure 23.

Phase_Seg1, Phase_Seg2

These phase buffer segments 1 and 2 are used to compensate for edge phase errors. These segments may be lengthened or shortened by resynchronization.

Sample point

The sample point shall be the point in time at which the bus level is read and interpreted as the value of that respective bit. Its location shall be at the end of Phase_Seg1.

Information processing time

The information processing time shall be the number of time quanta required for the calculation of the subsequent bit level. This calculation begins at the sample point and must be less than or equal to Phase_Seg2 (see 11.3.2.1 for additional restrictions).

Synchronization jump width (SJW)

As a result of resynchronization, Phase_Seg1 may be lengthened or Phase_Seg2 may be shortened. The amount of lengthening and shortening of the phase buffer segments has an upper limit given by the synchronization jump width.

Internal delay time

The internal delay time of a CAN node, t_{node}, shall be the sum of all asynchronous delays that occur along the transmission and reception path, relative to the bit timing logic unit of the CAN implementation. For more details see Figure 23.

In connection with Figure 23,

— the sum of output and input CAN node delays is critical relative to the configuration of the nominal bit time. The important characteristic parameter of a CAN node is given by Formula (1):

$$t_{node} = t_{output} + t_{input} \quad (1)$$

— for proper arbitration, the following conditions shall be met

$$t_{Prop\_Seg} \geq t_{node\_A} + t_{node\_B} + 2 \times t_{busline} \quad (2)$$

— the leading transmitting bit timing logic with respect to synchronization of CAN, node A shall be able to know the correct bus level of bit n at the sampling point. The tolerable values of t_{node} depend on the required bit rate and line length of the bus (maximum distance between any two nodes) and of the possible bit timing as shown by the arbitration condition.

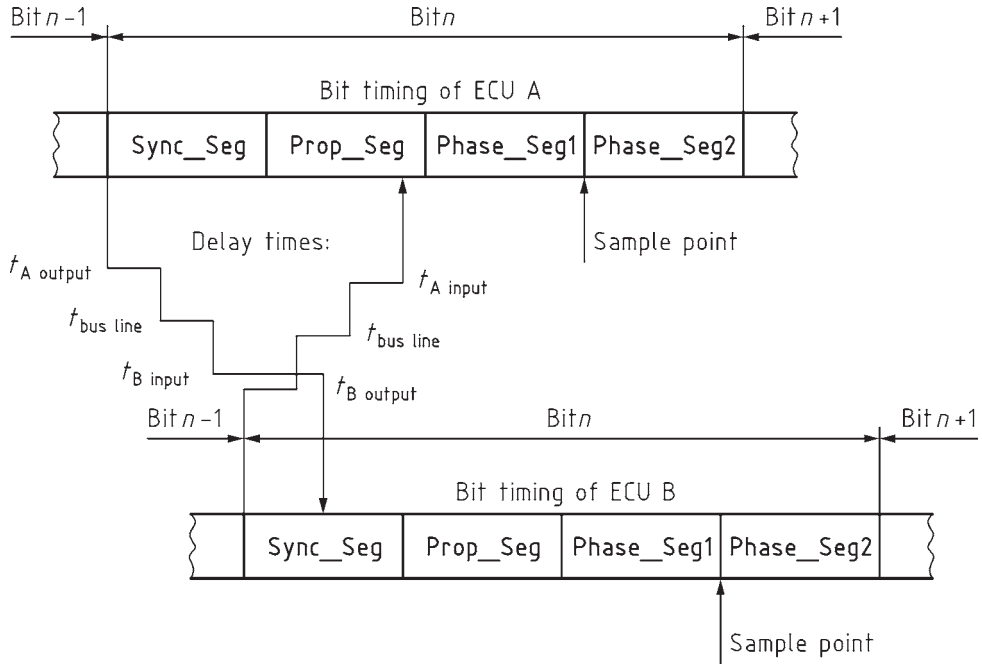


Figure 23 — Time relationship between delay times and bit time phases of CAN nodes A and B during arbitration phase

11.3.1.2 Configuration of the bit time parameters

Configuration of the bit time shall be performed using the following periods of time: Sync_Seg, Prop_Seg, Phase_Seg1, Phase_Seg2, and SJW, all representing integer number of time quanta based on the minimum time quantum and a prescaler m. For FD enabled implementations, there are two sets of configuration values, one for each bit time; see also Figure 22.

The configuration ranges for the lengths of the bit time segments shall be different for implementations that are FD enabled and for implementations that are not FD enabled. With the exception of the synchronization segment, which shall be exactly one time quantum long, implementations may allow time segments that exceed the minimum required configuration ranges specified in Table 8.

Table 8 — Time segments' minimum configuration ranges

Parameter	Not FD enabled	FD enabled		
		Separate prescalers	Shared prescaler	Separate or shared
	Nominal bit time	Nominal bit time	Nominal bit time	Data bit time
Prescaler m	1 to 32	1 to 32	1 to 32	
Sync_Seg	1 time quantum(N)	1 time quantum(N)	1 time quantum(N)	1 time quantum(D)
Prop_Seg	1 to 8 time quanta(N)	1 to 48 time quanta(N)	1 to 96 time quanta(N)	0 to 8 time quanta(D)
Phase_Seg1	1 to 8 time quanta(N)	1 to 16 time quanta(N)	1 to 32 time quanta(N)	1 to 8 time quanta(D)
Phase_Seg2	2 to 8 time quanta(N)	2 to 16 time quanta(N)	2 to 32 time quanta(N)	2 to 8 time quanta(D)
SJW	1 to 4 time quanta(N)	1 to 16 time quanta(N)	1 to 32 time quanta(N)	1 to 8 time quanta(D)

The following restrictions shall be met for the configuration of the bit time segments.

— The information processing time shall be less than or equal to 2 time quanta long.

— In data bit time, Phase_Seg2 shall be greater than or equal to the maximum information processing time.

— In nominal bit time, Phase_Seg2 shall be greater than or equal to the maximum of these two items: SJW and the information processing time.

— In nominal bit time and in data bit time, SJW shall be less than or equal to the minimum of these two items: Phase_Seg1 and Phase_Seg2.

In case of synchronization, Phase_Seg1 may be longer and Phase_Seg2 may be shorter than its programmed value. The position of the sample point may differ in the two bit timing configurations; the length of the Prop_Seg may be zero in the configuration for the data bit rate.

An example for the nominal bit time configuration based on a prescaler of $m(N) = 3$ and the time segments Prop_Seg(N) = 5, Phase_Seg1(N) = 4, Phase_Seg2(N) = 4 combined with the data bit time configuration based on a prescaler of $m(D) = 1$ and the time segments Prop_Seg(D) = 1, Phase_Seg1(D) = 6, Phase_Seg2(D) = 6, as well as the resulting bits of intermediate length BRS and CRC Delimiter, is shown in Figure 22.

In a CAN implementation, Prop_Seg and Phase_Seg1 do not need to be programmable separately; it shall be sufficient to program the sum of Prop_Seg and Phase_Seg1. The total number of time quanta in a nominal bit time shall be programmable at least from 8 to 25 for implementations that are not FD enabled. For implementations that are FD enabled, the total number of time quanta in a data bit time shall be programmable at least from 5 to 25 and in a nominal bit time at least from 8 to 80.

If the same time quantum length is used in the nominal bit time and in the data bit time and the positions of the sample points in the nominal bit time are the same in all CAN nodes of a network, then optimum clock tolerance is accomplished for networks using FD frames.

The frequencies of the node clock oscillators in the different CAN nodes shall be coordinated in order to provide a network-wide specified time quantum for the nominal bit time and, for FD enabled implementations, for the data bit time. The acceptable oscillator tolerances of the protocol implementations (see 11.3.2.5) and the potential for incorrect synchronization are determined by Phase_Seg1, Phase_Seg2, and SJW.

When the bit time configuration of a CAN implementation is not programmable, its fixed bit time configuration shall meet the restrictions for the configuration of the bit time segments as specified in this chapter.

11.3.2 Synchronization

11.3.2.1 Description

The state machine synchronizing the operation of the CAN implementation to the signals on the CAN bus shall operate in time steps of one time quantum. At each time quantum, it shall be analysed whether the bus state is recessive or dominant. The bus state detected at the sample point of a bit shall be regarded as the value of that bit. A difference in bus states between two consecutive time quanta is an edge. A receiving node that is synchronized to an undisturbed frame detects edges only in the Sync_Seg of a bit time. Edges detected outside the Sync_Seg may cause the CAN node to synchronize its operation to that edge.

NOTE The bus comparator converting the physical signals into data signals may operate independently from the node clock. The time needed to synchronize the data signals to the node clock before they can be regarded by the clocked state machines is part of the CAN node's input delay time (see Figure 23).

Hard synchronization and resynchronization shall be two forms of synchronization. They shall obey the following rules.

a) Only one synchronization within one bit time (between two sample points) shall be allowed. After an edge was detected, synchronizations shall be disabled until the next time the bus state detected at the sample point is recessive.

b) An edge shall cause synchronization only if the bus state detected at the previous sample point (previous read bus state) was recessive. If a transmitter uses the transmitter delay compensation (see 11.3.3), also the first detected edge from recessive to dominant after the sample point of the CRC delimiter shall cause synchronization. In this case, CAN implementations may choose alternatively to synchronize on the first detected edge from recessive to dominant seen by an amount of at least the transmitter delay plus one time quantum after the sample point of the CRC delimiter. An edge with a positive phase error (see 11.3.2.2) shall not cause synchronization in a node sending a dominant bit.

c) Hard synchronization shall be performed on edges during inter-frame space (with the exception of the first bit of intermission), when a node is in bus integration state, and inside an FD Frame at the edge between the FDF bit and the following dominant res bit.

d) All other recessive to dominant edges fulfilling rules a) and b) shall be used for resynchronization with one exception: A node transmitting an FD Frame shall not synchronize while it transmits the data phase of that frame.

Clocking information shall be derived from transitions from one bit value to the other. The property that (due to the bit stuffing) only a fixed maximum number of successive bits have the same value shall provide the possibility of resynchronizing a CAN node to the bit stream during a frame.

11.3.2.2 Phase error of an edge

The phase error, *e*, of an edge is given by the position of the edge relative to Sync_Seg, measured in time quanta. The sign of phase error is given by the following:

— $e = 0$ if the edge lies within Sync_Seg;

— $e > 0$ if the edge lies between the Sync_Seg and the sample point of the current bit;

— $e < 0$ if the edge lies between the sample point of the current bit and the Sync_Seg of the following bit.

11.3.2.3 Hard synchronization

After a hard synchronization, the bit time shall be restarted by each bit timing logic unit with Sync_Seg completed. Thus hard synchronization shall force the edge which has caused the hard synchronization to lie within the synchronization segment of the restarted bit time. The function of hard synchronization shall not be limited by the synchronization jump width.

Optionally, CAN implementations may perform edge filtering during the bus integration state; see 10.9.4. Two consecutive nominal time quanta with dominant bus state shall be required to detect an edge that causes synchronization when this option is enabled by configuration. The synchronization shall happen after the second time quantum with dominant bus state is seen.

When edge filtering is performed, dominant bus-states shorter than two nominal time quanta shall be ignored.

11.3.2.4 Bit resynchronization

Resynchronization shall lead to a shortening or lengthening of the bit time to correct the position of the sample point in relation to the position of the detected edge. When the magnitude of the phase error of the edge which causes resynchronization is less than or equal to the programmed value of the synchronization jump width, the effect of a resynchronization shall be the same as a hard synchronization.

When the magnitude of the phase error is larger than the programmed value of the synchronization jump width,

— and if the phase error e is positive, Phase_Seg1 shall be lengthened by an amount equal to the synchronization jump width;

— and if the phase error e is negative, then Phase_Seg2 shall be shortened by an amount equal to the synchronization jump width.

If Phase_Seg2 is shortened to a value less than the information processing time, the calculation of the subsequent bit level may be completed after the end of Phase_Seg2.

NOTE Since synchronization is only enabled when the bus state is read recessive at the sample point and at most only one synchronization is allowed between two sample points, only edges from recessive to dominant are relevant for synchronization. The first edge will be used for synchronization, after that, synchronization is disabled at the first edge detected, even if that edge does not cause synchronization, e.g. because it is detected inside the Sync_Seg.

11.3.2.5 Tolerance range of the oscillator frequencies

The tolerance of a node clock oscillator frequency f_{osc} around the nominal frequency f_{nom} shall be given by the range $[(1 - df) \times f_{nom} \leq f_{osc} \leq (1 + df) \times f_{nom}]$. The tolerance *df* depends on the length of the time quantum, the segments of the bit time, and on the synchronization jump width. The maximum difference between the node clock oscillators of any two nodes shall be $2 \times df \times f_{nom}$.

The maximum tolerance df of f_{osc} shall meet the following conditions:

$$df < \frac{\mathrm{SJW}(N)}{2 \times 10 \times \text{bit time}(N)} \tag{3}$$

$$df < \frac{\min[\mathrm{Phase\_Seg1}(N), \mathrm{Phase\_Seg2}(N)]}{2 \times [13 \times \text{bit time}(N) - \mathrm{Phase\_Seg2}(N)]} \tag{4}$$

$$df < \frac{\mathrm{SJW}(D)}{2 \times 10 \times \text{bit time}(D)} \tag{5}$$

$$df < \frac{\min[\mathrm{Phase\_Seg1}(N), \mathrm{Phase\_Seg2}(N)]}{2 \times \left([6 \times \text{bit time}(D) - \mathrm{Phase\_Seg2}(D)] \times \frac{m(D)}{m(N)} + 7 \times \text{bit time}(N) \right)} \tag{6}$$

$$df < \frac{\mathrm{SJW}(D) - \max\left(0, \frac{m(N)}{m(D)} - 1\right)}{2 \times \left([2 \times \text{bit time}(N) - \mathrm{Phase\_Seg2}(N)] \times \frac{m(N)}{m(D)} + \mathrm{Phase\_Seg2}(D) + 4 \times \text{bit time}(D) \right)} \tag{7}$$

where

(N) indicates the values for the arbitration phase;

(D) indicates the values for the data phase.

The conditions given in Formulas (3) and (4) shall be met for Classical Frames; all conditions of Formulas (3) to (7) shall be met for FD Frames.

It is to be considered that SJW may not be larger than the smaller of the phase buffer segments and that the propagation time segment limits that part of the bit time that may be used for the phase buffer segments.

11.3.3 Transmitter delay compensation

CAN nodes are connected to the CAN network via an electrical circuit, the transceiver, that unavoidably imposes a time delay between the point in time when a bit is transmitted at the protocol controller's output signal and the point in time when the same bit signal appears at the protocol controller's input signal. Without transmitter delay compensation, the bit rate in the data-phase of FD Frames is limited by the fact that the transmitter detects a bit error if it cannot receive its own transmitted bit latest at the sample point of that bit.

FD enabled implementations shall support a transmitter delay compensation mechanism, to be used in applications where the length of the bit time in the data phase is shorter than the propagation delay from the PCS to PMA output message (see 11.4.2.1) to the PMA to PCS input message (see 11.4.3.1) as specified in other parts of ISO 11898. This mechanism shall only be used by transmitters in the data phase of FD Frames, when BRS is recessive. It shall be programmable whether the mechanism is used at all. When this mechanism is used, the value of the prescaler for the data time quantum $m(D)$ shall be one or two. The implementation shall be able to compensate transmitter delays of at least two data bit times.

NOTE The PCS to PMA output message corresponds to a transceiver's bus driver input pin RxD, the PMA to PCS input message corresponds to a transceiver's bus comparator output pin TxD.

The transmitter delay compensation mechanism defines a secondary sample point SSP. When it is used, the transmitter shall ignore bit errors detected at the sample point. The received bit value shall be compared, at the SSP, with the (delayed) transmitted bit value. If a bit error is detected at the SSP, the transmitter shall react to this bit error at the subsequent following sample point. Bit error detection shall be disabled for those bits at the end of the data phase where the SSPs of the bits would be in the following arbitration phase.

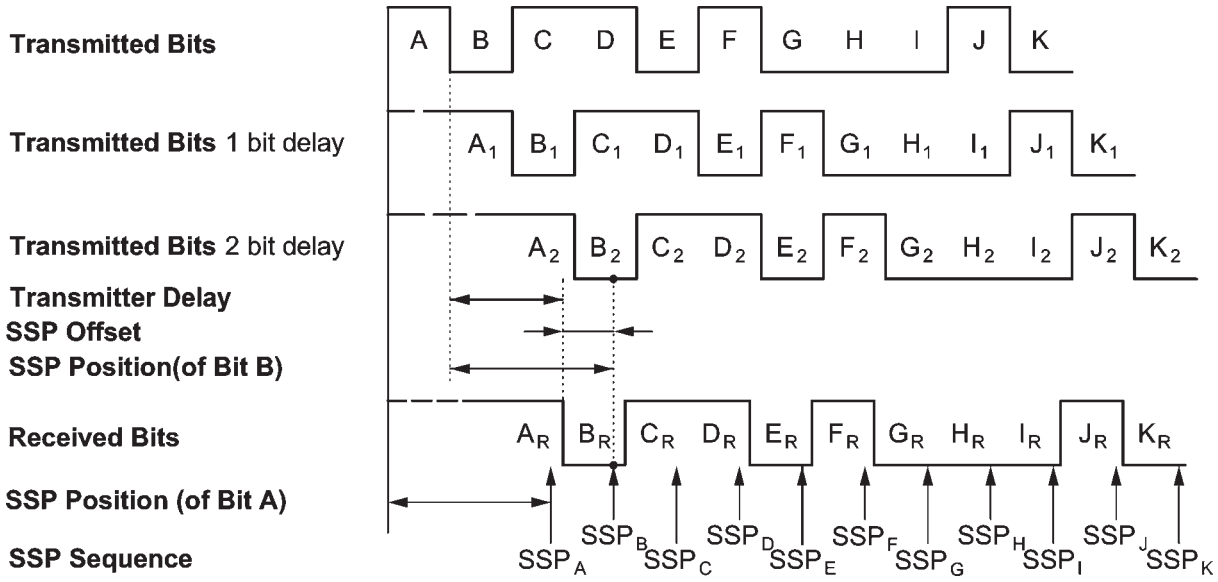


Figure 24 — Position of the secondary sample point

The position of the SSP shall be specified by its distance from the start of the bit time. The configuration range for this SSP position shall be at least 0 to 63 minimum time quanta. The SSP position shall either be set to a fixed value or it shall be set to a value derived from a measurement of the actual transmitter delay. The measurement of the transmitter delay shall be done in each transmitted frame at the recessive to dominant edge from the FDF bit to the res bit. Application of the measured value for the SSP position will be done at the data phase of the same frame. A counter shall be started when the transmitter starts to transmit the dominant res bit at the transmit output. The counter shall be incremented by one each minimum time quantum until the dominant signal is detected at the receive input; then the counter shall be stopped. The counter value is the measured transmitter delay time. When the SSP position is derived from the delay measurement, the position shall be the sum of the measured transmitter delay value and a configurable SSP offset.

The example bit stream [A, B, … K] in Figure 24 shows the relation between the transmitter delay and the SSP position. In that example, the transmitter delay is almost two data bit times long. Therefore the SSP is placed in the range from the transmitter delay up to three data bit times after the start of the bit; at a point where the input signal is expected to have stabilized. In this case, the bits from the received bit stream shall be compared to the bits from the transmitted bit stream that are delayed by two data bit times: At SSP_A, the received bit A_R is compared to the delayed transmitted bit A_2 and so on.

Optionally, if SSP position value is an odd number and the value of the prescaler for the data time quantum is 2, then SSP position may be determined by dividing the SSP position value by 2 and rounding down the result.

NOTE Figure 25 shows the end of the data phase with transmitter delay compensation. The SSP sequence stops at the transmitter's sample point of the (first) CRC Delimiter bit, where the data phase ends. The actual received bit value is ignored at the sample point, the SSP sequence is stopped. In that example, the last CRC sequence bit before the CRC Delimiter is locally disturbed (received bit E_R is seen recessive, but was transmitted dominant). The transmitter tolerates a second recessive CRC Delimiter bit before it sees the dominant ACK. There would be an EF instead of the ACK if the disturbance of bit E were not locally limited.

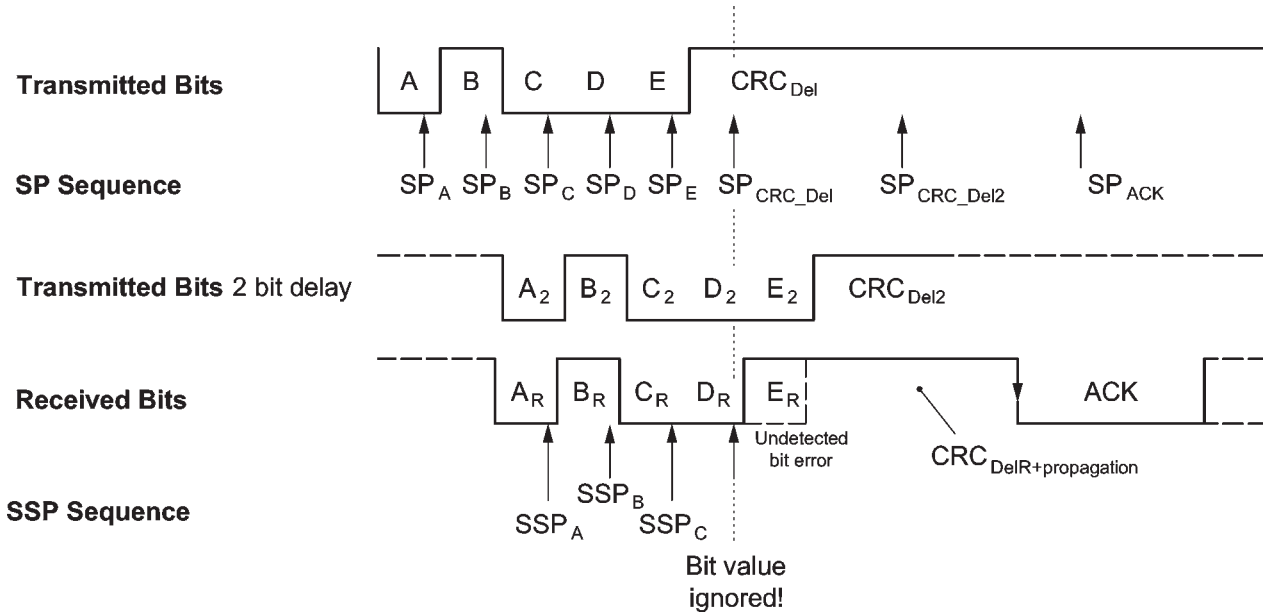


Figure 25 — End of transmitter delay compensation phase

CAN implementations may choose to continue the SSP sequence beyond the limit of the data phase, checking for local errors at the transmitting node as shown in Figure 26.

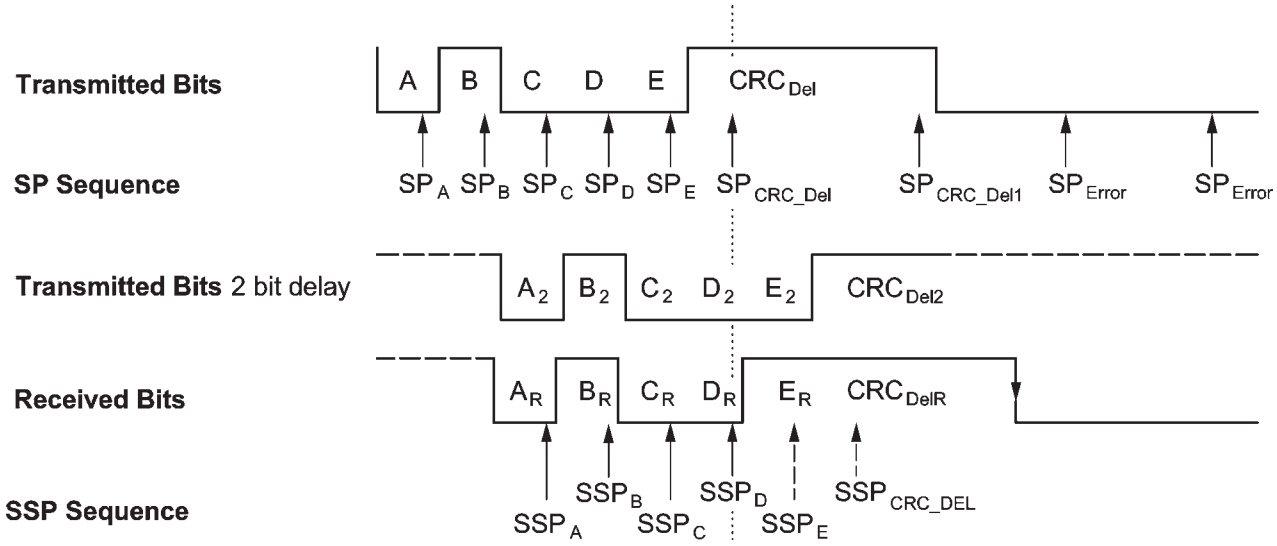


Figure 26 — Optional continuation of SSP sequence

11.4 AUI specification

11.4.1 General

The attachment unit interface is the interface between the PCS that is specified in this part of ISO 11898 and the PMA that is specified in other parts of ISO 11898 or other ISO standards (e.g. ISO 11992-1).

11.4.2 PCS to PMA messages

11.4.2.1 Output message

The PCS shall send an output message to the PMA sub-layer whenever it receives an Output_Unit from the MAC sub-layer. The output message causes the PMA to send a dominant or recessive bit.

11.4.2.2 Bus_off message

The PCS shall send a bus_off message to the PMA sub-layer whenever it receives a bus_off_request from the supervisor (see 12.1).

11.4.2.3 Bus_off_release message

The PCS shall send a bus_off_release message to the PMA sub-layer whenever it receives a bus_off_release_request from the supervisor (see 12.1).

11.4.2.4 FD_Transmit message

The PCS shall send an FD_Transmit message to the PMA sub-layer whenever it receives an FD_Transmit from the MAC sub-layer. This message is optional for FD enabled implementations.

11.4.2.5 FD_Receive message

The PCS shall send an FD_Receive message to the PMA sub-layer whenever it receives an FD_Receive from the MAC sub-layer. This message is optional for FD-enabled implementations.

11.4.3 PMA to PCS message

11.4.3.1 Input message

The PMA sub-layer shall send an input message to the PCS whenever the PMA has received a bit from the medium. The input signal indicates to the PCS the arrival of a dominant or recessive bit.

12 Description of supervisor FCE

12.1 Fault confinement

12.1.1 Objectives

The objective of fault confinement is to preserve a high availability of the data transmission network even in the presence of a defective node. Therefore, the fault confinement strategies shall prove reliable on the following instances:

a) distinction between temporary errors and permanent failures;

b) localization and switching-off of faulty nodes.

12.1.2 Strategies

All nodes shall include a transmit error counter and a receive error counter. The transmit error counter shall register the number of errors during the transmission, and the receive error counter shall register the errors during the reception of frames.

When frames are sent or received correctly, the counters shall be decremented. When frames are sent or received with errors, the counters shall be incremented more than they are decremented in the absence of errors. The ratio in which the counters are incremented/decremented depends on the

acceptable ratio of invalid/valid frames on the bus. At any time the levels of the error counters reflect the relative frequency of previous errors.

Depending on predetermined counter values, the behaviour of nodes in respect to errors shall be modified. I.e. this shall range from a prohibition of sending error flags to cancel frames, up to switching-off of nodes which often send invalid frames.

12.1.3 Fault confinement interface specification

12.1.3.1 Description

Fault confinement interface shall be as given in Figure 27.

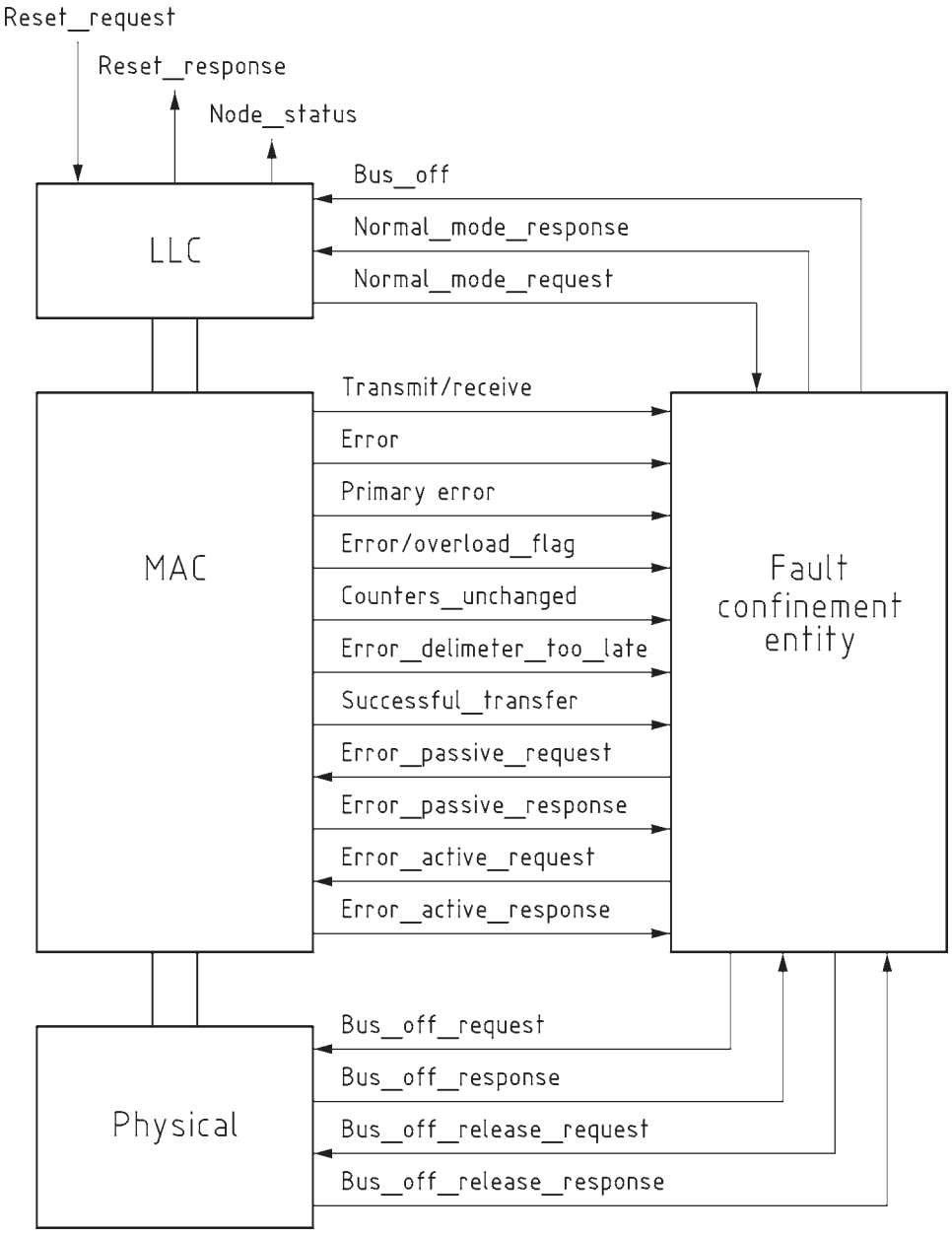


Figure 27 — Fault confinement interface

12.1.3.2 LLC sub-layer/FCE interface

The messages interchanged between the FCE and the LLC sub-layer shall be as given in Table 9 and Table 10.

Table 9 — LLC-to-FCE message

Message	Description
Normal_Mode_Request	Resets FCE to initial state (the values of the transmit error counter and the receive error counter are set to zero when the FCE passes into its initial state)

Table 10 — FCE-to-LCC message

Message	Description
Normal_Mode_Response	Response to the Normal_Mode_Request.
Bus_Off	Indicates that the node is in the bus-off state.

12.1.3.3 MAC sub-layer/FCE interface

The messages exchanges between the FCE and the MAC sub-layer shall be as given in Table 11 and Table 12.

Table 11 — MAC-to-FCE messages

Messages	Description
Transmit/receive	Indicates the node's current transfer mode.
Error	Indicates that the MAC sub-layer has detected an error (bit error, stuff error, CRC error, form error, ACK error).
Primary_error	Signals that the MAC sub-layer has detected a dominant bit after sending an error flag (indicates that the MAC sub-layer has detected a primary error and not an error that is caused by the error flag of another node).
Error/overload flag	Indicates that the MAC sub-layer is sending an error flag or an overload flag.
Counters_unchanged	Indicates that the FCE counters remain unchanged (due to special cases; see rule c) in 12.1.4.2).
Error_delimiter_too_late	Indicates that the MAC sub-layer is waiting too long for error delimiter. This signal is set each time after a sequence of eight consecutive dominant bits have been received after sending an error flag.
Successful_transfer	Indicates that transmission/reception was successfully completed.
Error_passive_response	Indicates that the node was set into the error passive state.
Error_active response	Indicates that the node was set into the error active state again.

Table 12 — FCE-to-MAC messages

Message	Description
Error_passive_request	Request to set the node into the error passive state.
Error_active_request	Request to set the node into the error active state.

12.1.3.4 PL/FCE interface

The messages exchanged between the FCE and the PL shall be as given in Table 13 and Table 14.

Table 13 — FCE-to-PL messages

Message	Description
Bus_off_request	Request to switch off the node from the bus.
Bus_off_release_request	Request to set the node into the normal transmit/receive node.

Table 14 — PL-to-FCE messages

Message	Description
Bus_off_response	Response to bus_off_request.
Bus_off_release_response	Response to bus_off_release_request.

12.1.4 Rules of fault confinement

12.1.4.1 Description

With respect to fault confinement, a node may be in one of the three states, depending on the error counter levels (see 6.13 and 6.15):

— error-active;

— error-passive;

— bus-off.

12.1.4.2 Error counting

The error counters shall be modified according to the following rules (more than one rule may apply during a given frame transfer).

a) When a receiver detects an error, the receive error counter shall be incremented by 1, except when the detected error was a bit error during the sending of an active error flag or an overload flag.

b) When a receiver detects a dominant bit as the first bit after sending an error flag, the receive error counter shall be incremented by 8.

c) When a transmitter sends an error flag, the transmit error counter shall be incremented by 8.

   Exception 1:

   If the transmitter is error-passive and detects an ACK error because of not detecting a dominant ACK and does not detect a dominant bit while sending its passive error flag.

   Exception 2:

   If the transmitter sends an error flag because a stuff error occurred during arbitration, whereby the stuff bit should have been recessive, and has been sent recessive but is monitored to be dominant.

   In exception 1 and in exception 2, the transmit error counter remains unchanged.

d) If a transmitter detects a bit error while sending an active error flag or an overload flag, the transmit error counter shall be incremented by 8.

e) If a receiver detects a bit error while sending an active error flag or an overload flag, the receive error counter shall be incremented by 8.

f) Any node shall tolerate up to 7 consecutive dominant bits after sending an active error flag, passive error flag, or overload flag. After detecting 14 consecutive dominant bits (in case of an active error flag or an overload flag) or after detecting 8 consecutive dominant bits following a passive error flag,

and after each sequence of additional 8 consecutive dominant bits, every transmitter shall increment its transmit error counter by 8 and every receiver shall increment its receive counter by 8.

g) After the successful transmission of a frame (getting ACK and no error has been detected until EOF is finished), the transmit error counter shall be decremented by 1 unless it was already 0.

h) After the successful reception of a frame (reception without error up to the ACK slot and the successful sending of the ACK bit), the receive error counter shall be decremented by 1, if it was between 1 and 127. If the receive error counter was 0, it shall stay at 0, and if it was greater than 127, then it shall be set to a value between 119 and 127.

12.1.4.3 Transition between error-active and error-passive states

If the transmit error counter or the receive error counter of a node exceeds 127 (carry condition in case of a 7-bit receive error counter) then the supervisor shall request the MAC sub-layer to set the corresponding node into the error-passive state.

An error condition causing a node to become error-passive shall cause the node to send an active error flag.

An error-passive node shall become error-active again when both the transmit error counter and the receive error counter are less than or equal to 127 (see Figure 28).

When the receive error counter of a node exceeds the error-passive limit of 127, further increments of this receive error counter shall be limited by the width of the counter. At the next successful reception of a frame (transition to error-active), the receive error counter shall be set to a value below the error-passive limit [see error counting rule h) in 12.1.4.2].

12.1.4.4 Bus-off management

If the transmit error counter of a node is greater than 255 (carry condition in case of a 8-bit transmit error counter) then the supervisor shall request the PL to set the node into the bus-off state.

A node which is in the bus-off state shall have no influence on the bus. It shall not send any frames nor acknowledge DFs or RFs. Whether or not such a node accepts DFs from the bus depends on the implementation.

Upon a restart request, a node which is in the bus-off state shall be integrating to the CAN communication (see 10.9.4) and may become error-active (no longer bus-off) with its error counters both set to zero after having monitored 128 occurrences of the idle condition (see 4.28) on the bus (see Figure 28).

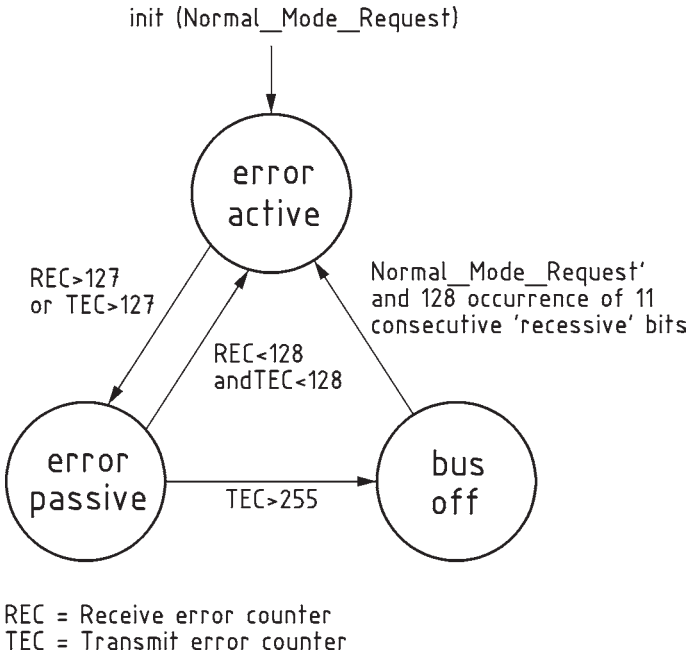


Figure 28 — Node status transition diagram

12.1.5 Network start-up

If during network start-up only one node is online and if this node transmits some frame, it will not get an ACK. In this case, it shall detect an error and repeat the frame. According to rule c) in 12.1.4.2, exception 1, it may become error-passive but shall not go bus-off.

A node that had been switched off and is switched on again shall

— synchronize with already available nodes to integrate into the bus communication (see 4.29) before starting to transmit. Synchronization is achieved when 11 recessive bits that are equivalent to:

  ACK delimiter + EOF + Intermission

  or

  Error/overload delimiter + Intermission have been detected;

— wait for other nodes if there is no other node available at the moment without becoming bus-off.

12.2 Bus failure management

During normal operation, several bus failures may occur that influence the bus operation. These failures and the resulting behaviour of the network shall be in accordance with the used PMA.

Annex A
(informative)

Additional Information

A.1 Differences between Classical Frames and FD Frames

There are three implementation options for the support of frame formats:

a) support for the Classical frame format only, not tolerating the Flexible Data Rate frame format (Classical CAN);

b) support for the Classical frame format and tolerating the Flexible Data Rate frame format (CAN FD tolerant);

c) support for the Classical frame format and for the Flexible Data Rate frame format (CAN FD enabled).

Option c) is recommended to be implemented for new designs.

The difference between implementation options a) and b) is the protocol exception event that causes the FD tolerant implementation to enter the bus integration state when a recessive FDF bit is detected. That is described in 10.9.5.

The differences between implementation options a) and c) concern the frame formats, CRC mechanisms, acknowledgement and the transmitter delay compensation. They are referenced in the following clauses and subclauses:

1 Scope

CAN-FD added to the Scope.

4 Terms and definitions

4.1 arbitration phase

4.11 data bit rate

4.12 data bit time

4.14 data phase

4.17 FD enabled

4.18 FD Base Frame Format

4.19 FD Extended Frame Format

4.20 FD Frame

4.21 FD Intolerant

4.22 FD Tolerant

4.41 Protocol exception event

5 Symbols and abbreviated terms

BRS Bit Rate Switch

ESI	Error State Indicator
FD	Flexible Data Rate
FDF	FD Format Indicator
FBFF	FD Base Frame Format
FEFF	FD Extended Frame Format
res	reserved bit in FD Frames
RRS	Remote Request Substitution
SSP	Secondary Sample Point
TDC	Transmitter Delay Compensation

6.8 Error detection

17-bit CRC for FD Frames with up to 16 data-field bytes, 21-bit CRC for FD Frames with 20 to 64 data-field bytes.

Fixed bit stuffing and stuff count in the CRC field of FD Frames.

8.2.1 Types of connectionless-mode transmission services

Four types of frames have been increased to six due to the addition of:

LLC Data Frame in FD Base Frame format, and

LLC Data Frame in FD Extended Frame format.

8.4.2.3 Format field

The following have been added to support the CAN-FD functionality:

FBFF FD Base Frame Format, and

FEFF FD Extended Frame Format.

8.4.2.4 DLC field

DLC has been extended to support the CAN-FD functionality.

8.4.2.5 Data field

Data field has been extended to support the CAN-FD functionality.

10.4 Structure of MAC frames

The following have been added to support the CAN-FD functionality:

MAC Data Frame in FD Base Frame format,

MAC Data Frame in FD Extended Frame format.

10.4.2.3 Arbitration field

10.4.2.4 Control field

10.4.2.6 CRC field

10.4.2.7 ACK field

10.8 Order of bit transmission

Four figures describing FD Frames: Figure 13, Figure 14, Figure 16, and Figure 17.

10.9.5 Protocol exception event

10.12 Error signalling

Error signalling when leaving data phase of FD Frame

11.3.1.2 Configuration of the bit time parameters

Configuration values for data bit time

11.3.2.5 Tolerance range of the oscillator frequencies

Three formulas for FD Frames

11.3.3 Transmitter delay compensation

Exclusively used for FD Frames

Table A.1 — Compatibility between implementations

Function	Type of implementation			
	Classical CAN or other when disabling of Flexible Data Rate frame format supported and configured	CAN FD tolerant	CAN FD enabled	CAN FD enabled when disabling of Classical frame format supported and configured (see 10.9.10)
Reception and transmission of Classical Frames	Supported	Supported	Supported	No reception, no transmission. Error Frames will be sent when Classical frames present, unless node in Error Passive state or generally unable to transmit
Reception and transmission of FD Frames	No reception, no transmission. Error Frames will be sent when FD frames present, unless node in Error Passive state or generally unable to transmit	No reception, no transmission, no Error Frames will be sent when FD frames present	Supported	Supported

A.2 Non-mandatory implementation features

The non-mandatory features specified in the standard are listed in Table A.2.

Table A.2 — Optional features

No.	Optional feature	Described in
1	FD frame format	Clause 1
2	Disabling of frame formats	10.9.10
3	Limited LLC frames	8.5
4	No transmission of frames including padding bytes	8.5
5	LLC Abort interface	8.2.2, 8.3.4
6	ESI and BRS bit values	8.2.2.x, 10.4.2.4
7	Method to provide MAC data consistency	10.10
8	Time and time triggering	8.3.4, 9.2, 10.15
9	Time stamping	9.4
10	Bus monitoring mode	10.10, 10.14
11	Handle	4.24, 8.2.2.x, 8.4.1
12	Restricted operation	10.10, 10.15
13	Separate prescalers for nominal bits and for data bits	11.3.1.1, 11.3.1.2
14	Disabling of automatic retransmission	4.43, 8.3.4, 9.3.1, 10.9.6
15	Maximum number of retransmissions	10.9.6
16	Disabling of protocol exception event on res bit detected recessive	10.4.2.4
17	PCS_Status	11.2.1, 11.2.4, 11.2.5
18	Edge filtering during the bus integration state	10.9.4, 11.3.2.3
19	Time resolution for SSP placement	11.3.3
20	FD_T/R message	11.4.2.4, 11.4.2.5

A.3 Implementation hints

This part of ISO 11898 specifies the CAN communication on the CAN network; it does not specify the interface between an implementation of the CAN protocol and a host controller. This allows integration of CAN implementations into simple sensor or actor nodes, as well as into µCs or as separate ICs.

If the implementation allows to change the configuration of a node by software, the configuration data (e.g. bit time configuration, operating mode) needs to be locked against changes while the CAN communication is ongoing.

Bibliography

[1] ISO 7637-3, *Road vehicles — Electrical disturbances from conduction and coupling — Part 3: Electrical transient transmission by capacitive and inductive coupling via lines other than supply lines*

[2] ISO 11992-1, *Road vehicles — Interchange of digital information on electrical connections between towing and towed vehicles — Part 1: Physical and data-link layers*

[3] ISO/IEC 8802-2, *Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 2: Logical link control*

[4] ISO/IEC 8802-2:1998/Cor 1:2000

ICS 43.040.15

Price based on 65 pages