



BSI Standards Publication

Road vehicles — Local Interconnect Network (LIN)

Part 3: Protocol specification

National foreword

This British Standard is the UK implementation of ISO 17987-3:2025. It supersedes BS ISO 17987-3:2016, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee AUE/16, Data Communication (Road Vehicles).

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2025
Published by BSI Standards Limited 2025

ISBN 978 0 539 22716 1
ICS 43.040.15

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 July 2025.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------



**International
Standard**

ISO 17987-3

**Road vehicles — Local Interconnect
Network (LIN) —**

**Part 3:
Protocol specification**

*Véhicules routiers — Réseau Internet local (LIN) —
Partie 3: Spécification du protocole*

**Second edition
2025-07**

Reference number
ISO 17987-3:2025(en)

 **Documents**
Document Sharing Hub
© ISO 2025



COPYRIGHT PROTECTED DOCUMENT

© ISO 2025

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword	v
Introduction	vi
1 Scope	1
2 Normative references	1
3 Terms, definitions, symbols and abbreviated terms	1
3.1 Terms and definitions	1
3.2 Symbols	4
3.3 Abbreviated terms	4
4 Node concept	5
4.1 General	5
4.2 Concept of operation	6
4.2.1 Commander and responder	6
4.2.2 Frames	6
4.2.3 Data transport	7
5 Protocol requirements	7
5.1 Signal	7
5.1.1 Management	7
5.1.2 Types	7
5.1.3 Consistency	7
5.1.4 Packing	7
5.1.5 Reception and transmission	8
5.2 Frame	9
5.2.1 Transfer	9
5.2.2 Structure	9
5.2.3 Frame length	13
5.2.4 Frame types	14
5.3 Schedule tables	19
5.3.1 General	19
5.3.2 Time definitions	19
5.3.3 Frame slot	19
5.3.4 Schedule table handling	20
5.4 Task behaviour model	20
5.4.1 General	20
5.4.2 Commander task state machine	20
5.4.3 Responder task state machine	20
5.5 Status management	23
5.5.1 General	23
5.5.2 Concept	23
5.5.3 Event-triggered frames	23
5.5.4 Reporting to the cluster	23
6 Node configuration and identification	24
6.1 General	24
6.2 LIN product identification	24
6.2.1 Supplier ID, function ID and variant ID	24
6.2.2 Serial number	24
6.2.3 Wildcards	25
6.3 Responder node model	25
6.3.1 Memory model	25
6.3.2 Responder node configuration variants	25
6.3.3 Initial node address	26
6.3.4 PDU structure	27
6.3.5 Node configuration handling	29
6.3.6 Node configuration services	30

ISO 17987-3:2025(en)

Annex A (normative) Definition of properties, frame identifiers and various examples	35
Annex B (informative) LIN history and version compatibility	39
Annex C (normative) LIN auto addressing procedures	43
Annex D (normative) LIN AA Procedure C	45
Annex E (normative) LIN AA Procedure D	53
Annex F (normative) LIN AA Procedure E	59
Bibliography	74

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).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

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

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 31, *Data communication*.

This second edition cancels and replaces the first edition (ISO 17987-3:2016), which has been technically revised.

The main changes are as follows:

- master and slave terms used for the LIN node types in ISO 17987:2016 (all parts) are replaced within this document with inclusive language terms commander and responder. This also applies for abbreviations and file formats NCF and LDF;
- auto addressing added;
- editorial updates.

A list of all parts in the ISO 17987 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

The LIN protocol as proposed is an automotive focused low speed UART-based network (Universal Asynchronous Receiver Transmitter). Some of the key characteristics of the LIN protocol are signal-based communication, schedule table-based frame transfer, commander/responder communication with error detection, node configuration and diagnostic service communication.

The LIN protocol is for low-cost automotive control applications as, for example, door module and air conditioning systems (see also history described in [Annex B](#)). It serves as a communication infrastructure for low-speed control applications in vehicles by providing the following:

- signal-based communication to exchange information between applications in different nodes;
- bit rate support from 1 kbit/s to 20 kbit/s;
- deterministic schedule table-based frame communication;
- network management that wakes up and puts the LIN cluster into sleep mode in a controlled manner;
- status management that provides error handling and error signalling;
- transport layer that allows large amount of data to be transmitted (such as diagnostic services);
- specification of how to handle diagnostic services;
- electrical physical layer specifications;
- node description language describing properties of responder nodes;
- network description file describing behaviour of communication;
- application programming interface.

The ISO 17987 series is based on the open systems interconnection (OSI) basic reference model as specified in ISO/IEC 7498-1 which structures communication systems into seven layers.

The OSI model structures data communication into seven layers called (top down) application layer (layer 7), presentation layer, session layer, transport layer, network layer, data link layer and physical layer (layer 1). A subset of these layers is used in the ISO 17987 series.

The ISO 17987 series distinguish between the services provided by a layer to the layer above it and the protocol used by the layer to send a message between the peer entities of that layer. The reason for this distinction is to make the services, especially the application layer services and the transport layer services, reusable also for other types of networks than LIN. In this way, the protocol is hidden from the service user and it is possible to change the protocol if special system requirements demand it.

The ISO 17987 series provides all documents and references required to support the implementation of the requirements related to.

- ISO 17987-1: provides an overview of the ISO 17987 series and structure along with the use case definitions and a common set of resources (definitions, references) for use by all subsequent parts.
- ISO 17987-2: specifies the requirements related to the transport protocol and the network layer requirements to transport the PDU of a message between LIN nodes.
- ISO 17987-3 (this document): specifies the requirements for implementations of the LIN protocol on the logical level of abstraction. Hardware related properties are hidden in the defined constraints.
- ISO 17987-4: specifies the requirements for implementations of active hardware components which are necessary to interconnect the protocol implementation.

ISO 17987-3:2025(en)

- ISO/TR 17987-5: specifies the LIN API (Application Programming Interface) and the node configuration and identification services. The node configuration and identification services are specified in the API and define how a responder node is configured and how a responder node uses the identification service.
- ISO 17987-6: specifies tests to check the conformance of the LIN protocol implementation according to ISO 17987-2 and ISO 17987-3. This comprises tests for the data link layer, the network layer and the transport layer.
- ISO 17987-7: specifies tests to check the conformance of the LIN electrical physical layer implementation (logical level of abstraction) according to ISO 17987-4.

Road vehicles — Local Interconnect Network (LIN) —

Part 3: Protocol specification

1 Scope

This document specifies the LIN protocol including the signal management, frame transfer, schedule table handling, task behaviour, status management, and commander and responder node. It contains also OSI layer 5 properties according to ISO 14229-7 UDSONLIN-based node configuration and identification services (SID: B0₁₆ to B8₁₆) belonging to the core protocol specification.

A node (normally a commander node) that is connected to more than one LIN network is handled by higher layers (i.e. the application) not within the scope of this document.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 17987-1, *Road vehicles — Local Interconnect Network (LIN) — Part 1: General information and use case definition*

ISO 17987-2:2025, *Road vehicles — Local Interconnect Network (LIN) — Part 2: Transport protocol and network layer services*

ISO 17987-4:2025, *Road vehicles — Local Interconnect Network (LIN) — Part 4: Electrical physical layer (EPL) specification 12 V/24 V*

3 Terms, definitions, symbols and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 17987-1 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1.1

big-endian

method of storage of multi-byte numbers with the most significant bytes at the lowest memory addresses

3.1.2

break field

pattern indicating the start of a LIN frame

3.1.3

broadcast NAD

addressing every responder node on LIN

3.1.4

bus interface

hardware (transceiver, UART, etc.) of a node that is connected to the physical bus wire in a *cluster* ([3.1.8](#))

3.1.5

bus sleep state

state in which a node only expects an internal or external wake up

Note 1 to entry: The nodes switch output level to the recessive state.

Note 2 to entry: Adapted from ISO 17987-2:2025, 5.2.

3.1.6

classic checksum

checksum type considering *data bytes* ([3.1.10](#)) only

Note 1 to entry: Used for *diagnostic frames* ([3.1.11](#)) and *frames* ([3.1.15](#)) processed by nodes according to protocol version 1.x.

3.1.7

commander request frame

CRF

diagnostic frames ([3.1.11](#)) issued by the commander node *frame identifier* ([3.1.16](#))

3.1.8

cluster

communication system comprising the LIN wire and all connected nodes

3.1.9

data

response ([3.1.29](#)) part of a *frame* ([3.1.15](#)) carrying one to eight *data bytes* ([3.1.10](#))

3.1.10

data byte

single byte in the *response* ([3.1.29](#))

3.1.11

diagnostic frame

commander request frame ([3.1.7](#)) and the *responder response frame* ([3.1.28](#))

3.1.12

diagnostic trouble code

DTC

value referring to a specific fault in a system implemented in the server

3.1.13

enhanced checksum

checksum model used for all non-*diagnostic frames* ([3.1.11](#)), except for legacy LIN slave nodes of protocol version 1.x

3.1.14

event-triggered frame

frame type allowing multiple responder nodes to provide their *response* ([3.1.29](#)) to the same *header* ([3.1.19](#))

3.1.15

frame

communication entity consisting of a *header* ([3.1.19](#)) and *response* ([3.1.29](#))

3.1.16

frame identifier

unique value identifier of a *frame* ([3.1.15](#))

3.1.17

frame slot

time period reserved for the transfer of a specific *frame* ([3.1.15](#)) on LIN

Note 1 to entry: This corresponds to one entry in the schedule table.

3.1.18

go-to-sleep command

special *commander request frame* ([3.1.7](#)) issued to force responder nodes to *bus sleep state* ([3.1.5](#))

Note 1 to entry: Adapted from ISO 17987-2:2025, 5.4.

3.1.19

header

first part of a *frame* ([3.1.15](#)) consisting of the *break field* ([3.1.2](#)), *sync byte field* ([3.1.31](#)) and *protected identifier* ([3.1.25](#)); always sent by the commander

3.1.20

LIN description file

file format describing a LIN network with one commander and at least one responder node

3.1.21

LIN product identification

number containing the supplier, function and variant identification in a responder node

3.1.22

little-endian

method of storage of multi-byte numbers with the least significant bytes at the lowest memory addresses

3.1.23

node capability file

NCF

file format that describes a responder node as seen from the LIN network

3.1.24

operational state

state in which the responder node may transmit/receive *frames* ([3.1.15](#))

Note 1 to entry: Adapted from ISO 17987-2:2025, 5.2.

3.1.25

protected identifier

PID

8-bit value consisting of a unique 6-bit *frame identifier* ([3.1.16](#)) and 2-bit parity

3.1.26

publisher

node providing a *frame response* ([3.1.29](#)) containing *signals* ([3.1.32](#))

3.1.27

request

diagnostic frame ([3.1.11](#)) transmitted by the commander node requesting *data* ([3.1.9](#)) from a responder node

3.1.28

responder response frame

RRF

frame ([3.1.15](#)) used for diagnostic communication sent by one of the responder nodes

3.1.29

response

answer sent by a responder node on a diagnostic *request* ([3.1.27](#)), or part of the LIN frame following the *header* ([3.1.19](#))

3.1.30

service

combination of a diagnostic *request* ([3.1.27](#)) and *response* ([3.1.29](#))

3.1.31

sync byte field

byte field with fixed value of 55_{16}

3.1.32

signal

value or byte array transmitted in the *cluster* ([3.1.8](#)) using a signal-carrying *frame* ([3.1.15](#))

3.1.33

sporadic frame

group of *unconditional frames* ([3.1.35](#)) assigned to a schedule slot published by the commander node

3.1.34

subscriber

commander or responder node that receives the *data* ([3.1.9](#)) within a LIN frame

3.1.35

unconditional frame

frame ([3.1.15](#)) carrying *signals* ([3.1.32](#)) that is always sent in its allocated *frame slot* ([3.1.17](#))

3.2 Symbols

$\oplus$ exclusive disjunction (exclusive or)

EXAMPLE $a \oplus b$; exclusive or. True if exactly one of 'a' or 'b' is true.

$\neg$ negation

$\in$ element of

EXAMPLE $f \in S$; belongs to. True if 'f' is in the set 'S'.

3.3 Abbreviated terms

AA auto addressing

API application programming interface

ASIC application specific integrated circuit

CRF commander request frame

DTC diagnostic trouble code

ID_n protected identifier bit n, n = {0, 1, ..., 5} (see [5.2.2.5.4](#))

LDF LIN description file

LIN Local Interconnect Network

LSB least significant bit

MSB	most significant bit
NAD	node address
NCF	node capability file
NRC	negative response code
NVRAM	non-volatile random access memory
OSI	Open Systems Interconnection
PCI	protocol control information
PDU	protocol data unit
PID	protected identifier
P _n	parity bit n, n = {0, 1} (see 5.2.2.5.4)
ROM	read only memory
RRF	responder response frame
RSID	response service identifier
SID	service identifier
TL	transport layer
VRAM	volatile RAM

4 Node concept

4.1 General

The LIN specifications assume a software implementation of most functions, alternative realizations are promoted.

A node in the cluster connects to the physical bus wire via the transceiver. The frames are not accessed directly by the application; a signal-based interaction layer is added in between. As a complement, a transport layer (TL) interface exists between the application and the frame handler, see [Figure 1](#).

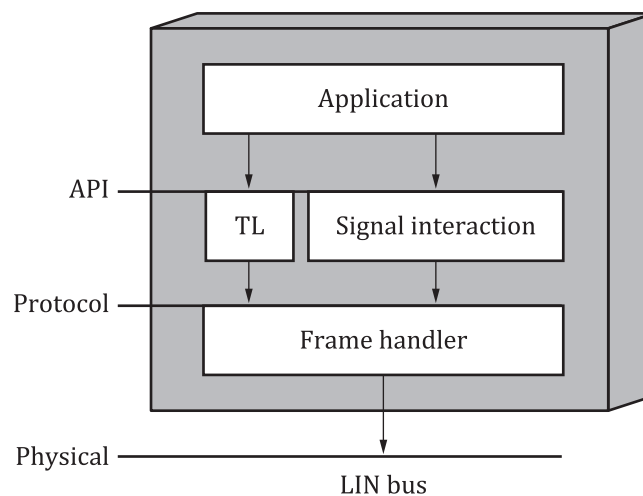


Figure 1 — Node concept

4.2 Concept of operation

4.2.1 Commander and responder

A cluster consists of one commander node and several responder nodes. A commander node contains a commander task as well as a responder task. All responder nodes contain a responder task only.

NOTE The responder task in the commander node and in the responder node is not identical due to differences in PID handling.

A commander node may participate in more than one cluster with one dedicated bus interfaces for each cluster. A sample cluster with one commander node and two responder nodes is shown in [Figure 2](#).

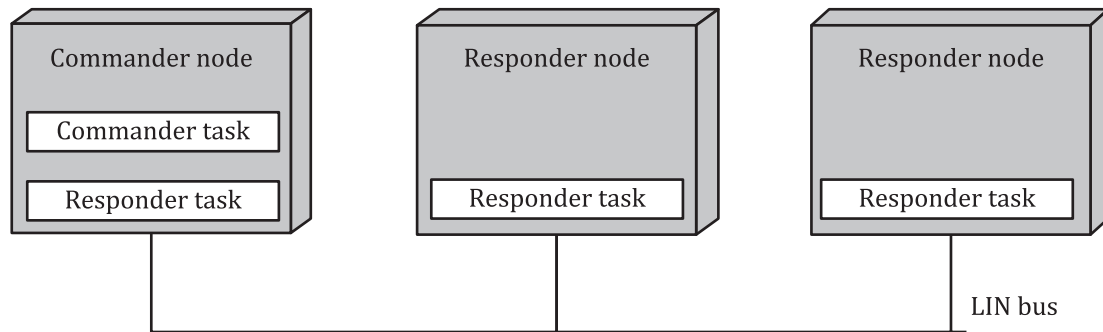


Figure 2 — Commander and responder tasks

The commander task decides when and which frame shall be transferred on the bus. The responder tasks provide the data transmitted by each frame. Both, the commander task and the responder task are parts of the frame handler.

4.2.2 Frames

A frame consists of a header (provided by the commander task) and a response (provided by a responder task).

The header consists of a break field and sync byte field followed by a protected identifier. The protected identifier uniquely defines the purpose of the frame and node providing the response. The responder task of the node assigned as a response transmitter provides the response. In case of diagnostic frames, not only the frame identifier but also the NAD assign the transmitting node.

The response consists of a data field and a checksum field. The responder tasks interested in the data associated with the frame identifier receives the response, verifies the checksum and uses the data transmitted.

[Figure 3](#) shows the LIN frame header and response fields.

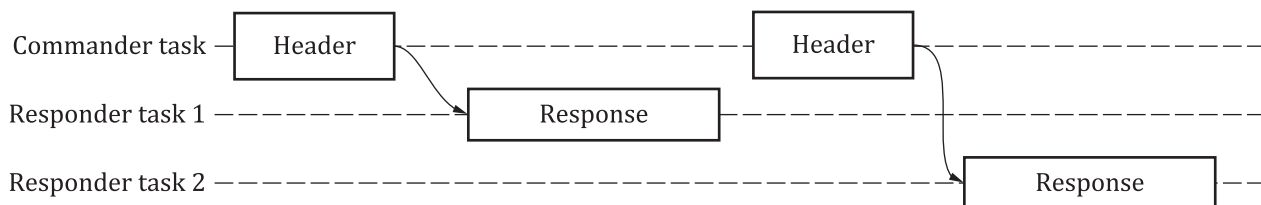


Figure 3 — LIN frame header and response fields

This results in the following desired features.

- System flexibility: nodes can be added to the LIN cluster without requiring hardware or software changes in other responder nodes.

- Message routing: the content of a message is defined by the frame identifier (similar to CAN).
- Multicast: any number of nodes can simultaneously receive and act upon a single frame.

4.2.3 Data transport

The following two types of data may be transmitted in a frame.

- Signals:

Signals are scalar values or byte arrays that are packed into the data field of a frame. A signal is always present at the same position of the data field for all frames with the same frame identifier.

- Diagnostic messages:

Diagnostic messages are transmitted in frames with two reserved frame identifiers. The interpretation of the data field depends on the data field itself as well as the state of the communicating nodes.

5 Protocol requirements

5.1 Signal

5.1.1 Management

A signal is transmitted in the data field of a frame.

5.1.2 Types

A signal is either a scalar value or a byte array.

A scalar signal is between 1 bit and 16 bit long (see [Table A.1](#)). Scalar signals are treated as unsigned integers. A 1-bit scalar signal is called a Boolean signal.

A byte array is an array of one up to eight bytes (see [Table A.1](#)).

Each signal shall have one publisher, i.e. it shall be always sent by the same node in the cluster. Zero, one or multiple nodes may subscribe to the signal.

All signals shall have initial values. The initial value for a published signal should be valid until the node writes a new value to this signal. The initial value for a subscribed signal should be valid until a new updated value is received from another node.

5.1.3 Consistency

Scalar signal writing or reading shall be atomic operations, i.e. it shall not be possible for an application to receive a signal value that is partly updated. This shall also apply to byte arrays. However, no consistency is guaranteed between any signals.

5.1.4 Packing

There is no restriction on packing scalar signals over byte boundaries. Each byte in a byte array shall map to a single frame byte starting with the lowest numbered data byte with LIN default (little-endianness) signal mapping and the highest number in case of big-endianness, see [5.2.2.6](#).

Several signals may be packed into one frame as long as they do not overlap each other.

NOTE Signal packing/unpacking is implemented more efficiently in software-based nodes if signals are byte aligned and/or if they do not cross byte boundaries.

All bits not used/defined in a frame shall be recessive (ones).

The same signal may be packed into multiple frames as long as the publisher of the signal is the same. If a node is receiving one signal packed into multiple frames, the latest received signal value is valid. Handling the same signal packed into frames on different LIN clusters is out of the scope.

5.1.5 Reception and transmission

The point in time when a signal is transmitted/received needs to be defined to help design tools and testing tools to analyse timing of signals. This means that all implementations behave in a predictable way.

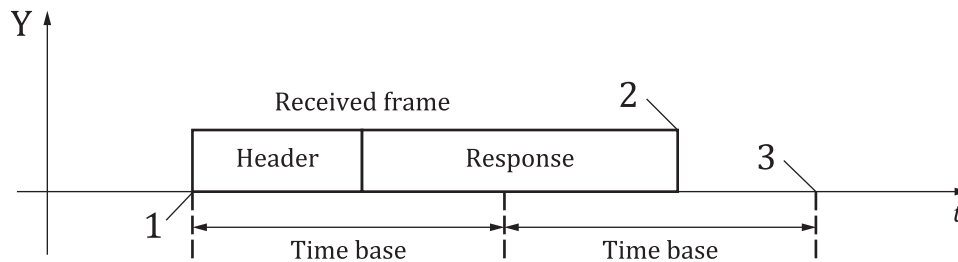
The definitions below do not contain factors such as bit rate tolerance, jitter, buffer copy execution time, etc. These factors need be taken into account to get a more detailed analysis. The intention for the definitions below is to have a basis for such analysis.

The timing is different for a commander node and a responder node. The reason is that the commander node controls the schedule and is aware of the due frame. A responder node gets this information first when the header is received.

The time base and time base tick is defined in [5.3](#).

A signal is considered received and available to the application as shown in [Figure 4](#).

- Commander node: after the maximum frame length expires and reception is successful, the signals received are updated at task level in the next time base tick.
- Responder node: if the checksum of the frame received is successfully validated at interrupt level, or in polling mode in the subsequent cycle, the signals are updated directly.



Key

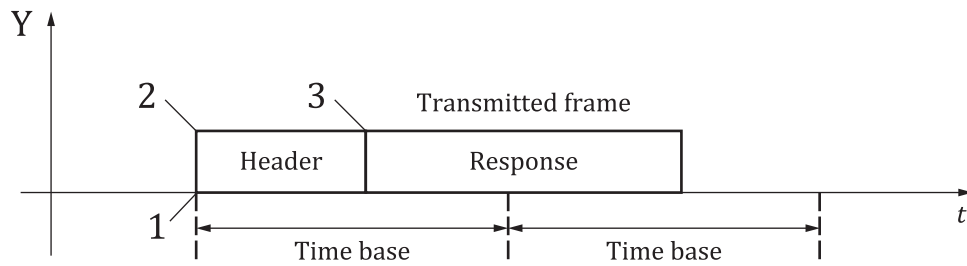
Y	frame on bus
t	time
1	time base tick
2	responder node – signal is available to the application
3	commander node – time the signal is available to the application

Figure 4 — Timing of signal reception

A signal is considered transmitted (latest point in time when the application may write to the signal).

- Commander node – before the frame transmission is initiated.
- Responder node – when the ID for the frame is received.

[Figure 5](#) shows the timing of signal transmission.



Key

Y	frame on bus
t	time
1	time base tick
2	commander – latest point the application can update the signal
3	responder – latest point the application can update the signal

Figure 5 — Timing of signal transmission

5.2 Frame

5.2.1 Transfer

The entities that are transferred on the LIN network are frames.

5.2.2 Structure

5.2.2.1 Definition of fields

The frame consists of a number of fields, one break field followed by 4 – 11 byte fields, labelled as in [Figure 6](#). The time it takes to send a frame is the sum of the time to send each byte plus the response space and the inter-byte spaces.

The header starts at the falling edge of the break field and ends after the end of the stop bit of the protected identifier (PID) field. The response starts at the end of the stop bit of the PID field and ends at the after the stop bit of the checksum field.

The inter-byte space is defined to be the time between the end of the stop bit of the preceding field and the start bit of the following byte. The response space is defined to be the inter-byte space between the PID field and the first data field in the data. Both shall be non-negative.

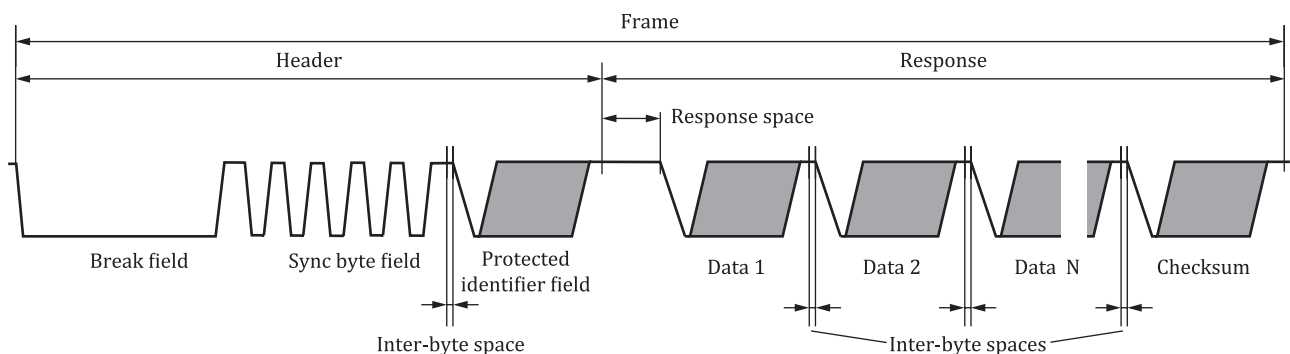


Figure 6 — Structure of a frame

5.2.2.2 Byte field

Each byte field, except the break field, is transmitted as the byte field as specified in [Figure 7](#). The LSB of the data is sent first and the MSB last. The start bit shall be encoded as a bit with value zero (dominant) and the stop bit shall be encoded as a bit with value one (recessive).

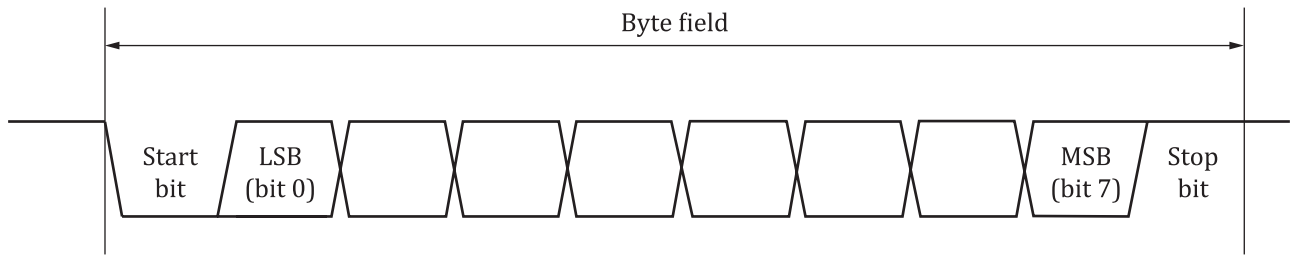


Figure 7 — Structure of a byte field

5.2.2.3 Break field

The break field is used to signal the beginning of a new frame. It is the only field that does not comply with [Figure 7](#). A break field is always generated by the commander task (in the commander node) and it shall be at least 13 nominal bit times of dominant value, followed by a break delimiter, as shown in [Figure 8](#). The break delimiter shall be at least one nominal bit time long. It should be sent with 2 bit length in order to avoid misinterpretation at receiving responder nodes (for details see [Table A.1](#)).

NOTE A UART can only handle complete bits, so it can occur on the physical layer that the break delimiter is shorter than one bit time.

A responder node shall use a break detection threshold of 11 dominant local responder bit times (see also [Table A.1](#)). However, responder nodes without specific break field detection capabilities use a detection threshold of $9,5 t_{BIT}$ on the Rx pin. A responder node shall not check that the break delimiter is at least one nominal bit time long. A responder node shall be capable of detecting a break delimiter of at least 9/16 of a bit in length on the Rx line.

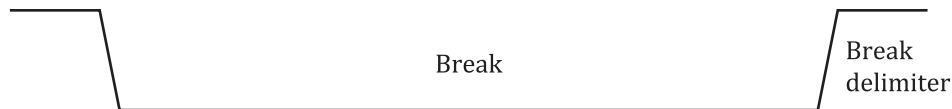


Figure 8 — Break field

5.2.2.4 Sync byte field

A responder task shall always be able to detect the break/sync byte field sequence, even if it expects a byte field (assuming the byte fields are separated from each other). When a break/sync byte field sequence occurs, the transfer in progress shall be aborted and processing of the new frame shall commence.

Sync byte field is a byte field with the data value 55_{16} , as shown in [Figure 9](#).



Figure 9 — Sync byte field

5.2.2.5 PID field

5.2.2.5.1 General

A protected identifier field consists of two subfields:

- the frame identifier;
- the parity.

Bit 0 to bit 5 are the frame identifier and bit 6 and bit 7 are the parity.

5.2.2.5.2 Frame identifier

Six bits (ID0 to ID5) are reserved for the frame identifier and values in the range of 0_{10} to 63_{10} shall be used. The frame identifiers are split into three categories:

- values 0_{10} to 59_{10} are used for signal carrying frames;
- values 60_{10} ($3C_{16}$) and 61_{10} ($3D_{16}$) are used to carry diagnostic and node configuration data;
- values 62_{10} ($3E_{16}$) and 63_{10} ($3F_{16}$) are reserved for future protocol enhancements.

5.2.2.5.3 Parity

The parity (P0 and P1) is calculated on the frame identifier bits as shown in [Formula \(1\)](#) and [Formula \(2\)](#):

$$P0 = ID0 \oplus ID1 \oplus ID2 \oplus ID4 \quad (1)$$

$$P1 = \neg(ID1 \oplus ID3 \oplus ID4 \oplus ID5) \quad (2)$$

Valid frame identifiers are listed in [Table A.2](#).

5.2.2.5.4 Mapping

The mapping of the bits (ID0 to ID5 and P0 and P1) is shown in [Figure 10](#).



Figure 10 — Mapping of frame identifier and parity to the protected identifier field

5.2.2.6 Data

A frame carries between one and eight bytes of data. The number of data contained in a frame with a specific frame identifier shall be agreed by the publisher and all subscribers. A data byte is transmitted as part of a byte field, see [Figure 7](#).

The data fields are labelled Data 1, Data 2, ... up to maximum Data 8, see [Figure 11](#).

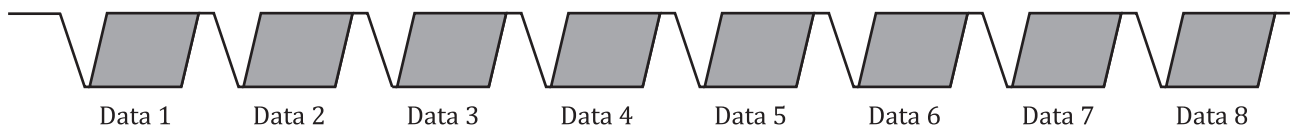


Figure 11 — Numbering of the data bytes in a frame with eight data bytes

How signals are mapped into frames is a decision that shall affect at least the whole LIN cluster.

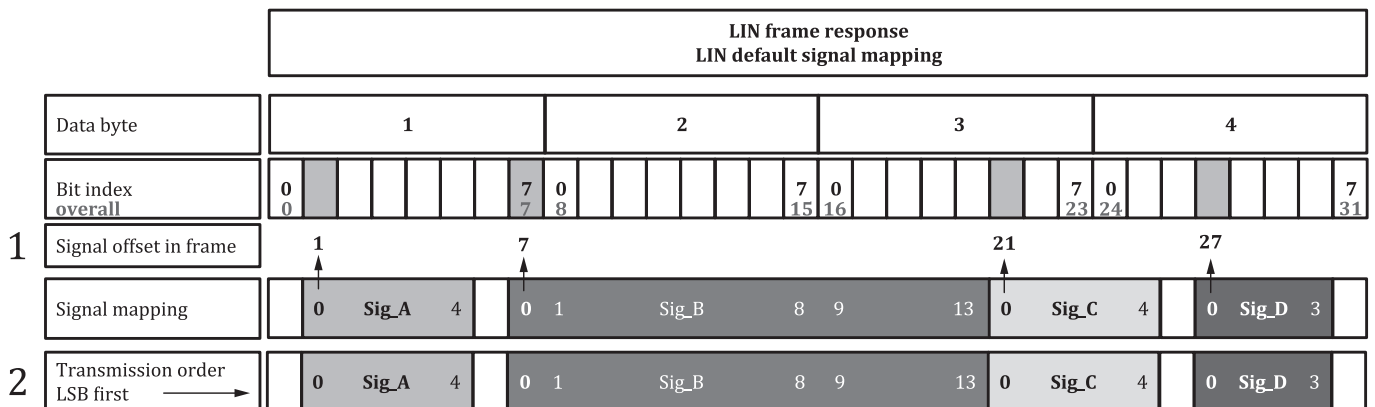
Beside the default signal mapping used in LIN clusters since its founding, an optional signal mapping variant is specified that allows to keep the big-endian signal encoding when routing frames from a backbone bus to a LIN cluster.

The two possible encodings are described in the following subclauses.

5.2.2.6.1 LIN default signal mapping to data bytes

For signals longer than one byte, the LSB signal entity is contained in the byte sent first and the MSB signal entity in the byte sent last (little-endian).

[Figure 12](#) provides an example of the LIN default signal mapping into frame response data bytes with four signals.



Key

- 1 see parameter signal_offset in ISO 17987-2:2025, 12.3.4.2
2 transmission order is starting from least significant bit in least significant data byte

Figure 12 — LIN default signal mapping into frame response data bytes

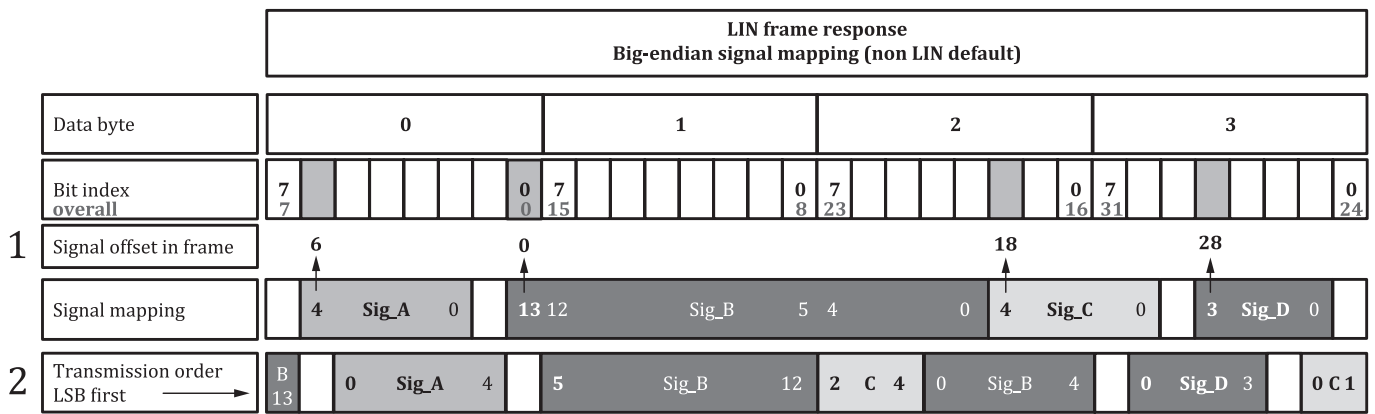
The LSB of each signal is referenced in the LDF and NCF frame definition as offset of the signal position. Key 1 values (Sig. A: 1, Sig. B: 7, Sig. C: 21 and Sig. D: 27) mark the offset for the four signals in the example.

5.2.2.6.2 Optional big-endian LIN signal mapping to data bytes variant

For signals longer than one byte, the MSB signal entity is contained in the byte sent first and the LSB signal entity in the byte sent last (big-endian).

Data byte transmission order, as well as the bit transmission order on LIN, is not changed due to big-endian signal encoding.

[Figure 13](#) provides an example of the LIN big-endian signal mapping into frame response data bytes with four signals.



Key

- 1 see parameter signal_offset in ISO 17987-2:2025, 12.3.4.2
- 2 transmission order is starting from least significant bit in least significant data byte

Figure 13 — LIN big-endian signal mapping into frame response data bytes

The most significant bit of each signal is referenced in the LDF and NCF frame definition as offset of the signal position. Key 1 values (Sig_A: 6, Sig_B: 0, Sig_C: 18 and Sig_D: 28) mark the offset for the four signals in the example.

LIN node configuration commands and belonging responder responses use the signals `supplier_id`, `function_id` and `Serial number` (response for RDBI ID = 1₁₀). As those commands have a LIN specific fixed format, they shall not be affected by this big-endian signal encoding variant.

5.2.2.7 Checksum

The last field of a frame is the checksum. The checksum contains the inverted eight bit sum with carry over all data bytes (classic checksum) or all data bytes and the protected identifier (enhanced checksum).

Eight bit sum with carry is equivalent to the sum of all values and subtract 255₁₀ every time the sum is greater or equal to 256₁₀. See [A.3](#) for examples how to calculate the checksum.

Checksum calculation over the data bytes and the protected identifier byte is called enhanced checksum and it is used for non-diagnostic communication.

The checksum is transmitted in a byte field, see [Figure 7](#).

Use of classic or enhanced checksum is managed by the commander node and it is determined per frame identifier.

Frame identifiers 60₁₀ (3C₁₆) to 61₁₀ (3D₁₆) shall always use classic checksum. Examples of checksum calculations can be found in [A.3](#).

5.2.3 Frame length

The nominal value for transmission of a frame exactly matches the number of bits sent (no response space and no inter-byte spaces). The nominal break field is 14 nominal bits long (break is 13 nominal bits and break delimiter is 1 nominal bit).

[Formula \(3\)](#) defines $t_{\text{HEADER_MIN}}$.

$$t_{\text{HEADER_MIN}} = 34 t_{\text{BIT}} \quad (3)$$

where t_{BIT} is the nominal time required to transmit a bit.

[Formula \(4\)](#) defines the $t_{\text{RESPONSE_MIN}}$.

$$t_{\text{RESPONSE_MIN}} = 10 * (N_{\text{Data}} + 1) * t_{\text{BIT}} \quad (4)$$

where N_{Data} is the number of response data bytes.

[Formula \(5\)](#) defines $t_{\text{FRAME_MIN}}$.

$$t_{\text{FRAME_MIN}} = t_{\text{HEADER_MIN}} + t_{\text{RESPONSE_MIN}} \quad (5)$$

The break field is 14 nominal bits or longer, see [5.2.2.3](#). This means that $t_{\text{HEADER_MAX}}$ puts a requirement on the maximum length of the break field.

The maximum space between the bytes is an additional 40 % duration compared to the nominal transmission time. The additional duration is split between the header (the commander task) and the frame response (a responder task).

[Formula \(6\)](#) defines $t_{\text{HEADER_MAX}}$:

$$t_{\text{HEADER_MAX}} = 1,4 * t_{\text{HEADER_MIN}} = 47,6 t_{\text{BIT}} \quad (6)$$

[Formula \(7\)](#) defines $t_{\text{RESPONSE_MAX}}$:

$$t_{\text{RESPONSE_MAX}} = 1,4 * t_{\text{RESPONSE_MIN}} \quad (7)$$

[Formula \(8\)](#) defines $t_{\text{FRAME_MAX}}$:

$$t_{\text{FRAME_MAX}} = t_{\text{HEADER_MAX}} + t_{\text{RESPONSE_MAX}} \quad (8)$$

The maximum length of the header, response and frame is based on the nominal time for a frame (based on the F_{Nom} as defined in ISO 17987-4:2025, 5.1). Therefore, the bit tolerances are included in the maximum length.

EXAMPLE A commander node that is 0,5 % slower than F_{Nom} is within $1,4 * t_{\text{HEADER_MIN}}$.

All subscribing nodes shall be able to receive a frame that has a zero overhead, i.e. which is $t_{\text{FRAME_MIN}}$ long.

Reception of a frame response after $t_{\text{FRAME_MAX}}$ is not guaranteed. A receiving node can reject the frame.

NOTE Tools and tests are used to check $t_{\text{FRAME_MAX}}$.

5.2.4 Frame types

5.2.4.1 General

The frame type refers to the preconditions that shall be valid to transmit the frame. Some of the frame types are only used for specific purposes, which are also defined in the following subclauses. Not all frame types specified need to be used in a network (see also [A.4](#)).

5.2.4.2 Unconditional frame

Unconditional frames carry signals and their frame identifiers are in the range 0_{10} to 59_{10} ($3B_{16}$).

The header of an unconditional frame is always transmitted when a frame slot allocated to the unconditional frame is processed (by the commander task). The publisher of the unconditional frame (the responder task) shall always provide the response to the header. All subscribers of the unconditional frame shall receive the frame and make it available to the application (assuming no errors were detected).

[Figure 14](#) shows a sequence of three unconditional frames. A transfer is always initiated by the commander. It has a single publisher and one or multiple subscribers.

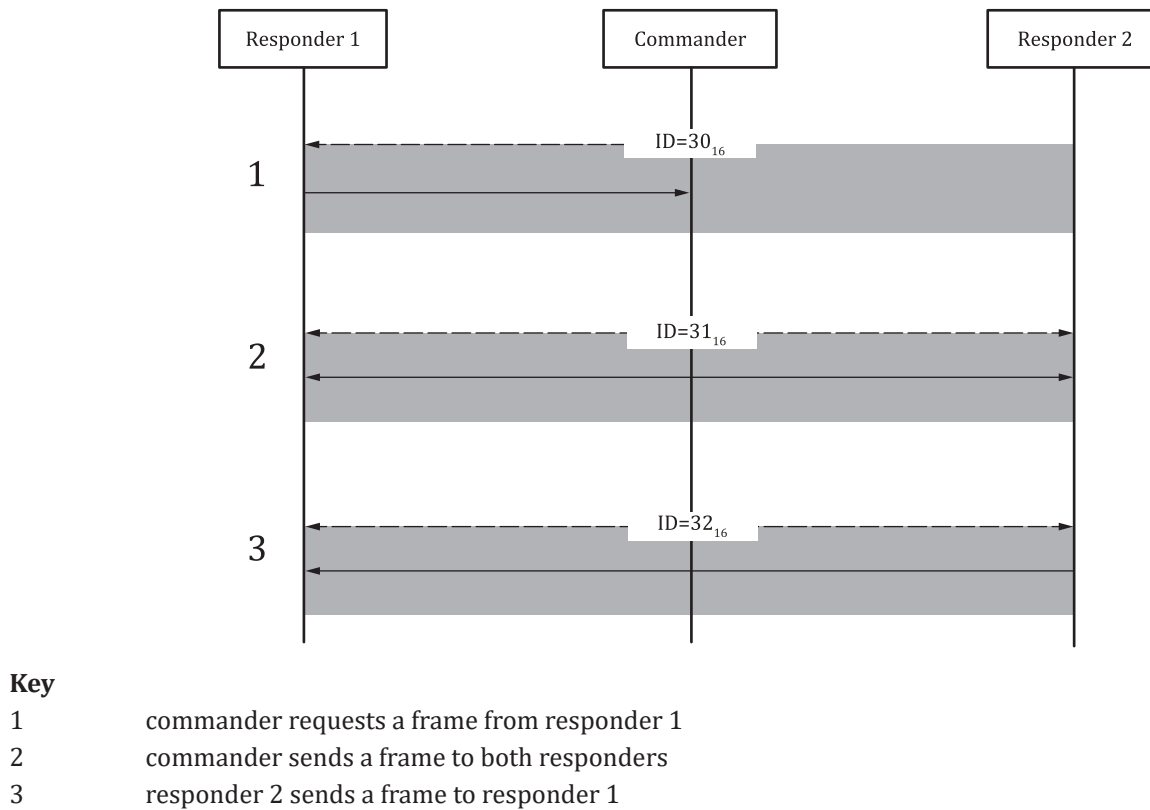


Figure 14 — Three unconditional frame transfers

5.2.4.3 Event-triggered frame

5.2.4.3.1 General

The purpose of an event-triggered frame is to lower the reaction time of the LIN cluster without assigning too much of the bus bandwidth to the polling of multiple responder nodes with seldom occurring events.

All subscribers of the event-triggered frame shall receive the frame and use its data (if the checksum is validated) as if the associated unconditional frame was received.

If the unconditional frame associated with an event-triggered frame is scheduled as an unconditional frame, the response shall always be transmitted (i.e. behave as a scheduled unconditional frame).

5.2.4.3.2 Unconditional frames associated with the event-triggered frame

Event-triggered frames carry the response of one or more unconditional frames. The unconditional frames associated with an event-triggered frame shall:

- have equal length,
- use the same checksum type classic checksum or enhanced checksum,
- not contain a signal mapped in the first data field due to the protected identifier mapped to it (even if the associated unconditional frame is scheduled as an unconditional frame in the same or another schedule table)
- be published by different responder nodes, and
- not be included directly in the same schedule table as the event-triggered frame is scheduled.

5.2.4.3.3 Transmission of the event-triggered frame

The header of an event-triggered frame is transmitted when a frame slot allocated to the event-triggered frame is processed. The publisher of an associated unconditional frame shall only transmit the response if at least one of the signals carried in its unconditional frame is updated. If the response is successfully transmitted, the signal is no longer considered to be updated.

If none of the responder nodes respond to the header, the rest of the frame slot is silent and the header is ignored.

If more than one responder node responds to the header in the same frame slot, a collision occurs.

5.2.4.3.4 Collision resolving

The commander node shall resolve the collision in a collision resolving schedule table. Each event-triggered frame has an associated collision resolving schedule table. The switch to the collision resolving schedule is made automatically by the driver in the commander node (i.e. not by the application). The collision resolving schedule shall be activated at the start of the subsequent frame slot after the collision.

At least all the associated unconditional frames shall be listed in this collision resolving schedule table. The collision resolving schedule may contain other unconditional frames than the associated frames. These other unconditional frames may be of different length.

After the collision schedule table has been processed once, the driver in the commander node shall switch back to the previous schedule table. It shall continue with the schedule entry subsequent to the schedule entry where the collision occurred (or first schedule entry in case the collision occurred in the last entry).

If one of the colliding responder nodes withdraws without corrupting the transfer, the commander node does not detect this. A responder node that has withdrawn its response shall therefore retry transmitting its response in the next occurrence of the appropriate PID until successful; otherwise, the response is lost.

In case the commander node application switches the schedule table before the collision is resolved, the collision resolving is lost. The new schedule table is activated as described in [5.3.4](#).

NOTE The colliding responder nodes still have their responses pending for transmission.

EXAMPLE 1 A schedule table contains only one event-triggered frame (ID = 10₁₆). The event-triggered frame is associated with two unconditional frames from responder 1 (ID = 11₁₆) and responder 2 (ID = 12₁₆). The collision resolving schedule table contains the two unconditional frames. See [Figure 15](#) for the behaviour on the bus.