INTERNATIONAL STANDARD

ISO 14229-7

Second edition
2022-04

Road vehicles — Unified diagnostic services (UDS) —

Part 7: UDS on local interconnect network (UDSonLIN)

Véhicules routiers — Services de diagnostic unifiés (SDU) —

Partie 7: SDU sur l'implémentation LIN (SDUsurLIN)

Reference number
ISO 14229-7:2022(E)

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 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 14229-7:2015), which has been technically revised.

The main changes are as follows:

— restructuration of the document;

— introduction of requirement numbers, names and definitions;

— technical content improvements based on implementation feedback from the automotive industry.

A list of all parts in the ISO 14229 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 ISO 14229 series has been established in order to define common requirements for diagnostic systems, whatever the serial data link is.

To achieve this, the ISO 14229 series is based on the Open Systems Interconnection (OSI) Basic Reference Model in accordance with ISO/IEC 7498-1[1] and ISO/IEC 10731[2], which structures communication systems into seven layers. When mapped on this model, the services used by a diagnostic tester (client) and an electronic control unit (ECU, server) are structured into the following layers:

— application layer (layer 7) specified in ISO 14229-1 and ISO 14229-3 to ISO 14229-8;

— presentation layer (layer 6) specified in ISO 14229-1 and ISO 14229-3 to ISO 14229-8;

— session layer services (layer 5) specified in ISO 14229-2 and ISO 14229-3 to ISO 14229-8.

Figure 1 illustrates the UDSonLIN document and related documents according to the OSI model.

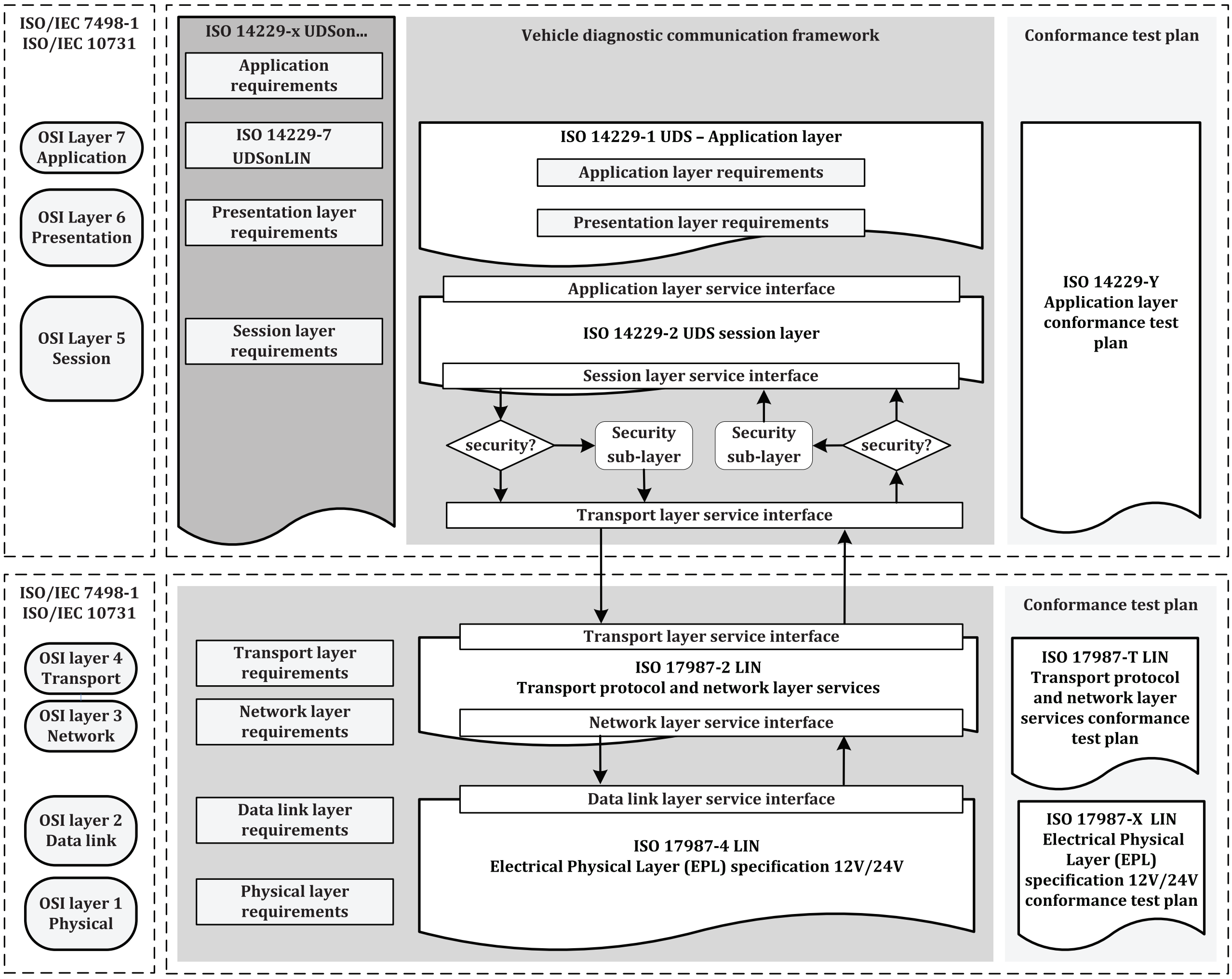


Figure 1 — UDSonLIN document reference according to OSI model

Road vehicles — Unified diagnostic services (UDS) —

Part 7:
UDS on local interconnect network (UDSonLIN)

1 Scope

This document specifies an application profile for the implementation of unified diagnostic services (UDS) local interconnect network (LIN) in road vehicles (UDSonLIN).

UDSonLIN references ISO 14229-1 and ISO 14229-2 and specifies implementation requirements of the diagnostic services to be used for diagnostic communication on Local Interconnect Network.

This document includes:

— additional requirements specific to the implementation of UDS on local interconnect network; and

— specific restrictions in the implementation of UDS on local interconnect network.

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 14229-1, *Road vehicles — Unified diagnostic services (UDS) — Part 1: Application layer*

ISO 14229-2, *Road vehicles — Unified diagnostic services (UDS) — Part 2: Session layer services*

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

ISO 17987-3, *Road vehicles — Local Interconnect Network (LIN) — Part 3: Protocol specification*

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

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 14229-1 and ISO 14229-2 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/

4 Symbols and abbreviated terms

4.1 Symbols

—	empty table cell or feature undefined
t	time
$t_{P2\_CAN\_Client}$	time axis of CAN client
$t_{P2\_CAN\_Server}$	time axis of CAN server
$t_{P2\_LIN\_Commander}$	time axis of LIN commander
$t_{P2\_LIN\_Responder}$	time axis of LIN responder
$t_{P2\_CAN\_Client\_Max}$	maximum time value of the CAN client
$t_{P2\_CAN\_Server\_Max}$	maximum time value of the CAN server
$t_{P2\_LIN\_Commander\_Max}$	maximum time value of the LIN commander
$t_{P2\_LIN\_Responder\_Max}$	maximum time value of the LIN responder

4.2 Abbreviated terms

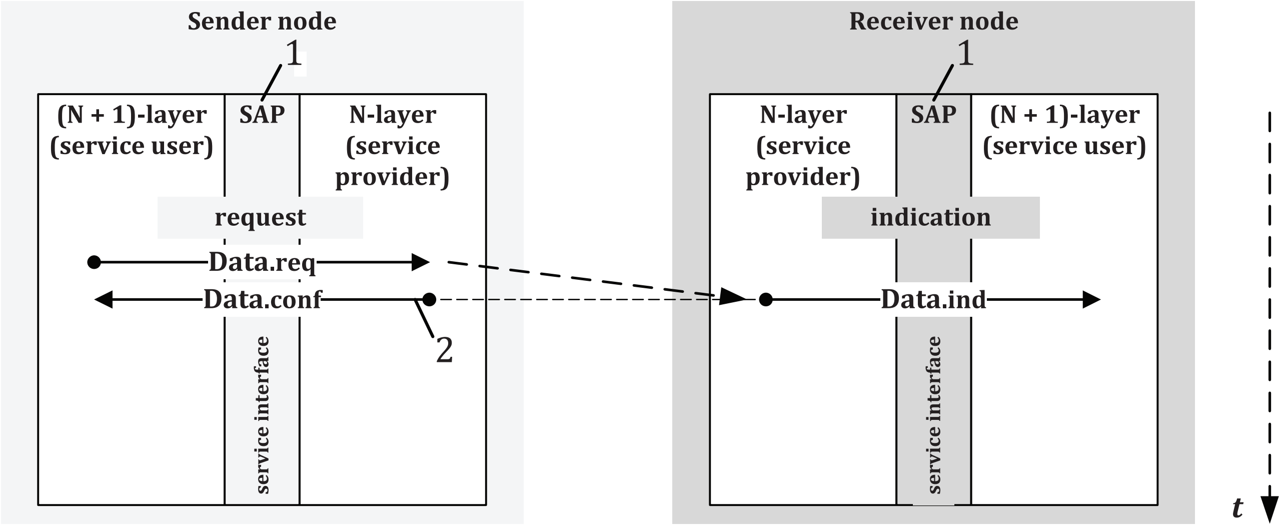
AE	address extension
Mtype	message type
NAD	node address
OSI	Open System Interconnection
UDS	unified diagnostic services
SA	source address
TA	target address

5 Conventions

This document is based on OSI service conventions as specified in ISO/IEC 10731[2].

6 Service primitive interface definition

Figure 2 shows the `Data.req` (request), `Data.ind` (indication), and `Data.conf` (confirmation) service interface.



Key

1 service access point between application and application layer

2 read back from N-layer service provider

t time

Figure 2 — Data.req, Data.ind, and Data.conf service interface

7 Technical requirements

7.1 Overview

Table 1 provides an overview on the technical requirements and their associated requirement number.

Table 1 — Technical requirements overview

OSI#.REQ#	Technical requirement title
7	**Application layer**
7.1	ISO 14229-1 service primitive parameters
7.2	A_Data.req, A_Data.ind, and A_Data.conf service interface
7.3	UDSonLIN – UDSonLIN-specific requirements
7.4	UDSonLIN – No UDSonLIN-specific requirements
7.5	UDSonLIN – A_Length – Definition
7.6	UDSonLIN – A_Length – Message buffer
7.7	UDSonLIN – A_Length – Commander node determines maximum size of receive buffer
7.8	UDSonLIN – CommunicationControl – Activation and de-activation of message type
7.9	UDSonLIN – ResponseOnEvent – ResponseHeader
7.10	UDSonLIN – Request and response message timing parameter values
6	**Presentation layer**
—	No requirement statement in this document
5	**Session layer**
5.1	UDSonLIN – Service primitive parameter definition
5.2	UDSonLIN – S_Data.req, S_Data.ind, and S_Data.conf service interface
4	**Transport layer**

Table 1 *(continued)*

OSI#.REQ#	Technical requirement title
4.1	UDSonLIN – Service primitive parameters
4.2	UDSonLIN – T_Data.req, T_Data.ind, and T_Data.conf service interface
4.3	UDSonLIN – Mapping of data link independent service primitives onto LIN data link-dependent service primitives
4.4	UDSonLIN – Mapping of T_PDU onto N_PDU
3	**Network layer**
3.1	UDSonLIN – Service primitive parameter definition
3.2	UDSonLIN – N_Data.req, N_Data.ind, and N_Data.conf service interface
3.3	UDSonLIN – N_TAtype service primitive parameter
3.4	UDSonLIN – Same N_TAtype request and associated response message format
3.5	UDSonLIN – Responder node diagnostic class I – No additional network layer requirements
3.6	UDSonLIN – Responder node diagnostic class II – Conform to ISO 17987-2
3.7	UDSonLIN – Responder node diagnostic class II – Fixed node address
3.8	UDSonLIN – Responder node diagnostic class II – Ignore NAD 7E16 as broadcast address
3.9	UDSonLIN – Responder node diagnostic class III – Conform to ISO 17987-2
3.10	UDSonLIN – Responder node diagnostic class III – Fixed node address
3.11	UDSonLIN – Responder node diagnostic class III – Ignore NAD 7E16 as broadcast address
3.12	UDSonLIN – Commander node diagnostic class III – Conform to ISO 17987-2
3.13	UDSonLIN – Commander node diagnostic class III – Usage of NAD assignment in LIN cluster
3.14	UDSonLIN – Commander node diagnostic class III – Assignment of subnet number
2	**Data link layer**
2.1	UDSonLIN – Service primitive parameter definition
2.2	UDSonLIN – L_Data.req, L_Data.ind, and L_Data.conf service interface
1	**Physical layer**
—	No requirement statement in this document

7.2 Implementation guidelines

7.2.1 General

This clause defines how the diagnostic services, as defined in ISO 14229-1, apply to LIN.

To allow a common implementation of application layer and session layer, this document uses the session layer protocol as defined in ISO 14229-2 and focuses on necessary modifications and interfaces to adopt it to the ISO 17987 series.

The subfunction parameter definitions take into account that the most significant bit is used for the suppressPosRspMsgIndicationBit parameter as defined in ISO 14229-1.

It is the vehicle manufacturer's responsibility to setup the LIN commander and responder nodes to exchange UDSonLIN information according to the ISO 17987 series.

7.2.2 Definition of diagnostic classes

7.2.2.1 Overview

Architectural, diagnostic communication performance, and transport protocol needs of responder nodes are accommodated by dividing diagnostic services functionality into three diagnostic classes.

Therefore, a diagnostic class is assigned to each responder node according to its level of diagnostic functionality and complexity.

7.2.2.2 Diagnostic class I

Smart and simple devices like intelligent sensors and actuators requiring none or very low amount of diagnostic functionality. Actuator control, sensor reading, and fault memory handling is done by the commander node, using signal carrying frames. Therefore, specific diagnostic support for these tasks is not required. Fault indication is always signal-based.

7.2.2.3 Diagnostic class II

A diagnostic class II responder node is similar to a diagnostic class I responder node, but it provides node identification support. The extended node identification is normally required by vehicle manufacturers. Testers or commander nodes use ISO 14229-1 diagnostic services to request the extended node identification information. Actuator control, sensor reading, and fault memory handling is done by the commander node, using signal carrying frames. Therefore, specific diagnostic support for these tasks is not required. Fault indication is always signal-based.

7.2.2.4 Diagnostic class III

Diagnostic class III responder nodes are devices with enhanced application functions typically performing their own local information processing (e.g. function controllers, local sensor/actuator loops). The responder nodes execute tasks beyond the basic sensor/actuator functionality and, therefore, require extended diagnostic support. Direct actuator control and raw sensor data are often not exchanged with the commander node and, therefore, not included in signal carrying frames. ISO 14229-1 diagnostic services for I/O control, sensor value reading, and parameter configuration (beyond node configuration) are required.

Diagnostic class III responder nodes have internal fault memory, along with associated reading and clearing services. Optionally, reprogramming (flash/NVRAM reprogramming) of the responder node is possible. This requires an implementation of a boot loader and necessary diagnostic services to unlock the device initiate downloads and transfer data, etc.

The primary difference between diagnostic class II and diagnostic class III is the distribution of diagnostic capabilities between the LIN commander node and the LIN responder node for diagnostic class II while for a diagnostic class III LIN responder node, no diagnostic application features of the LIN responder node are implemented in the LIN commander node.

7.2.3 LIN node requirements

7.2.3.1 Commander node requirements

7.2.3.1.1 Commander message routing

The commander node usually is a high-performance ECU and, in most implementations, supports the ISO 14229-1 diagnostic services. The commander node and the external test equipment are connected through a backbone network (e.g. ISO 11898 series). The commander node shall receive all diagnostic requests addressed to the responder nodes from the backbone network and route them to the appropriate LIN cluster(s). Responses from the responder nodes shall be routed back to the backbone network through the commander node.

All diagnostic request and response messages addressed to the responder nodes can be routed in the network layer (i.e. no application layer routing). The commander node shall implement the LIN transport protocol (see ISO 17987-2) as well as the transport protocols used on the backbone network (e.g. ISO 15765-2).

7.2.3.1.2 Commander node fault management, sensor reading, I/O control

Diagnostic class I and diagnostic class II responder nodes (see ISO 17987-3) provide signal-based fault information and sensor, I/O access through signal carrying frames. The LIN commander node is responsible to handle the responder nodes signal based faults and handle the associated DTCs. The LIN commander node serves UDS requests directly to the client/tester and acts as a diagnostic application layer gateway. UDS services provide access to the sensor/actuator signals on the LIN network.

Diagnostic class III responder nodes (see ISO 17987-3) are independent diagnostic entities. The LIN commander node does not implement diagnostic services for the diagnostic capabilities of its diagnostic class III responder nodes.

7.2.3.2 Responder node requirements

Responder nodes are typically electronic devices that are not involved in a complex data communication. Also, their need of distributing diagnostic data is low. Responder nodes transmit simple diagnostic information such as error indications in signal carrying frames.

Node configuration can be performed by the commander node independently while diagnostic services are always routed on request from external or internal test equipment. Both cases use the same node address (NAD) and transport protocol with the exception that configuration is always performed through SingleFrames (SF). Only responder nodes have an NAD. The NAD is also used as the source address in a diagnostic responder response frame.

NOTE There is a one-to-many mapping between a physical node and a logical node and it is addressed using the NAD.

7.2.4 Signal-based diagnostics

7.2.4.1 Responder implementation

Signal-based diagnostics are implemented by responder nodes (diagnostic class I and II), which do not implement a fault memory and the diagnostic protocol to directly access this fault memory from an external test tool.

There are two types of failure transmission through signal carrying frames.

a) Type 1 failure information is periodically transmitted and encoded into an existing signal (e.g. upper values of signal range used to indicate specific failure conditions) by the responder node. A type 1 failure use case-specific and defined by vehicle manufacturers and is not part of this document.

b) Type 2 failure information is not periodically transmitted for components which do not generate a signal that is periodically transmitted (e.g. responder node internal failure). Additional signal-based failure transmission shall be implemented for type 2 failures (i.e. if a responder node is capable of locally detecting faults which are not transmitted through the associated signal in signal carrying frames already).

Each responder node shall transmit the failure status information that is monitored by the responder node to the commander node through signal carrying frames. The status information shall contain the current failure status of the responder nodes' components. A signal shall support the following states:

— no test result available, default, initialization value;

— test result: failed;

— test result: passed.

If a responder node implements more than one independent function, a status signal can be assigned to each function. In this case, only the failing function could be disabled by the application.

The fault state signals are set in the status management of the LIN description file (LDF).

7.2.4.2 Commander implementation

A failure status signal shall be assigned for each failure that would result in a separate DTC in the commander node.

This information is used to indicate a failure of one of the components to the commander node's application, which can then store the associated DTC. There should be one signal per replaceable component to simplify repair and maintenance of the vehicle.

7.2.5 Tool suite support

The implementation of diagnostic services in LIN nodes, using the transport protocol and network layer services as defined in ISO 17987-2, may be supported by a tool suite, which is compatible with the requirements stated in ISO 17987-2, ISO 17987-3, and ISO/TR 17987-5[3].

A standardized API with support of the 'C' programming language is specified in ISO/TR 17987-5[3].

8 Application layer

8.1 ISO 14229-1 service primitive parameters

This document is part of the ISO 14229 series and therefore the service primitive parameter implementation follows the ISO 14229-1 specification.

REQ	7.1 UDSonLIN – ISO 14229-1 service primitive parameters
The service primitive parameters shall be implemented as specified in ISO 14229-1.	

8.2 A_Data.req, A_Data.ind, and A_Data.conf service interface

This document is part of the ISO 14229 series and therefore the service interface implementation follows the ISO 14229-1 specification.

REQ	7.2 UDSonLIN – A_Data.req, A_Data.ind, and A_Data.conf service interface
The `A_Data.req`, `A_Data.ind`, and `A_Data.conf` service interface shall be implemented as specified in ISO 14229-1.	

8.3 UDSonLIN services overview

The purpose of Table 2 is to reference all ISO 14229-1 and ISO 14229-2 services as they are applicable for an implementation in this document. Table 2 contains the UDSonLIN diagnostic services. Certain UDSonLIN applications can restrict the number of useable services and can categorize them in application areas/diagnostic sessions (default session, programming session, etc.).

REQ	7.3 UDSonLIN – UDSonLIN-specific requirements
Services that are marked “UDSonLIN-specific requirements” shall be implemented as specified in the referenced subclause number in accordance with Table 2 "Reference" column.	

REQ	7.4 UDSonLIN – No UDSonLIN-specific requirements
Services specified in Table 2 that are marked “No UDSonLIN-specific requirements” shall be implemented as specified in ISO 14229-1 and ISO 14229-2 with no additional restrictions.	

NOTE The ISO 17987 series supports different diagnostic classes for LIN responder nodes. It is the vehicle manufacturer's responsibility to specify which diagnostic services of ISO 14229-1 are implemented in a LIN responder node.

Table 2 — Overview of applicable ISO 14229-1-defined services

Functional unit name	Diagnostic service name	Comment	Reference
Diagnostic and communication management	DiagnosticSessionControl	No UDSonLIN-specific requirements	—
	ECUReset	No UDSonLIN-specific requirements	—
	SecurityAccess	No UDSonLIN-specific requirements	—
	CommunicationControl	UDSonLIN-specific requirements	see 8.6
	TesterPresent	No UDSonLIN-specific requirements	—
	AccessTimingParameters	Not supported	—
	Authentication	Not supported	—
	SecuredDataTransmission	Not supported	—
	ControlDTCSetting	No UDSonLIN-specific requirements	—
	ResponseOnEvent	UDSonLIN-specific requirements	see 8.7
	LinkControl	Not supported	—
Data transmission	ReadDataByIdentifier	No UDSonLIN-specific requirements	—
	ReadMemoryByAddress	Not supported	—
	ReadScalingDataByIdentifier	Not supported	—
	ReadDataByPeriodicIdentifier	Not supported	—
	DynamicallyDefineDataIdentifier	Not supported	—
	WriteDataByIdentifier	No UDSonLIN-specific requirements	—
	WriteMemoryByAddress	Not supported	—
Data transmission	ReadDataByIdentifier	No UDSonLIN-specific requirements	—
	WriteDataByIdentifier	No UDSonLIN-specific requirements	—
Stored data transmission	ReadDTCInformation	No UDSonLIN-specific requirements	—
	ClearDiagnosticInformation	No UDSonLIN-specific requirements	—
Input/output control	InputOutputControlByIdentifier	No UDSonLIN-specific requirements	—
Remote activation of routine	RoutineControl	No UDSonLIN-specific requirements	—
Upload/ download	RequestDownload	No UDSonLIN-specific requirements	—
	RequestUpload	No UDSonLIN-specific requirements	—
	TransferData	No UDSonLIN-specific requirements	—
	RequestTransferExit	No UDSonLIN-specific requirements	—
	RequestFileTransfer	No UDSonLIN-specific requirements	—

NOTE Services which are not listed in Table 2 are not supported.

8.4 A_PDU definition

Figure 3 shows the A_PDU.

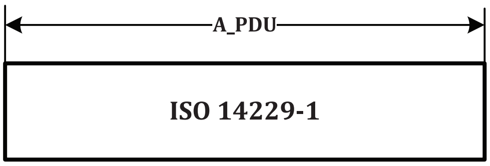


Figure 3 — A_PDU definition

8.5 A_Length definition

REQ	7.5 UDSonLIN – A_Length – Definition
The message length of an UDSonLIN message shall be implemented as specified in ISO 17987-2.	

REQ	7.6 UDSonLIN – A_Length – Message buffer
The message buffer shall be determined by the definition specified in REQ 7.5.	

REQ	7.7 UDSonLIN – A_Length – Commander node determines maximum size of receive buffer
The commander node shall synchronize with all its responder nodes to determine the maximum size of receive buffer based on the data length specified in order to store transmitted data.	

Certain diagnostic services, e.g. ReadDTCInformation, might exceed the message length restriction (depends on the number of DTCs to be reported). In such case, the negative response handling applies as specified in ISO 14229-1 for each concerned service.

8.6 CommunicationControl service UDSonLIN implementation requirements

This document specifies the specific UDSonLIN CommunicationControl service implementation requirements or restrictions.

REQ	7.8 UDSonLIN – CommunicationControl – Activation and de-activation of message type
The LIN commander node shall implement the service CommunicationControl as specified in ISO 14229-1 to allow for LIN cluster specific activation and de-activation of message type “normal communication” (see Table 3).	

NOTE The command to enable and disable the diagnostics-only mode is the diagnostic service CommunicationControl (see ISO 14229-1) with the parameter CommunicationType set to “Normal Communication Messages”.

Table 3 — CommunicationControl service and associated scheduling

Normal communication	Active scheduling
enabled	Interleaved diagnostics Scheduling active for the LIN cluster(s) requested through the parameter “nodeIdentificationNumber” of the diagnostic service CommunicationControl.

Table 3 *(continued)*

Normal communication	Active scheduling
disabled	Diagnostics-only Scheduling active for the LIN cluster(s) requested through the parameter “nodeIdentificationNumber” of the diagnostic service CommunicationControl.

8.7 ResponseOnEvent service UDSonLIN implementation requirements

This document specifies the specific UDSonLIN ResponseOnEvent service implementation requirements or restrictions.

REQ	7.9 UDSonLIN – ResponseOnEvent – ResponseHeader
The commander node shall provide a ResponseHeader in order to enable the responder to provide a ServiceToRespondTo in case the specified event occurs (see ISO 14229-1).	

The ResponseHeader is relevant for the LIN-schedule definition in case the responder node’s diagnostic implementation requires a ResponseOnEvent service.

8.8 Timing parameter definition

The request and response message timing parameters belong to the application layer.

REQ	7.10 UDSonLIN – Request and response message timing parameter values
The request and response message timing parameter values shall be implemented as specified in ISO 14229-2.	

Due to timing performance requirements for emissions-related systems, it is not recommended to connect an emissions-related primary ECU to a LIN cluster, if the LIN responders have any impact on the emissions-related communication, i.e. providing emissions relevant data for the emissions-related communication.

It is the vehicle manufacturer's responsibility to ensure that in case the client does not require a response message [suppressPosRspMsgIndicationBit = TRUE ('1')] and the server might need more than $t_{\mathtt{P2\_LIN\_Commander\_Max}}$ to process the request message that the client inserts sufficient time between subsequent requests.

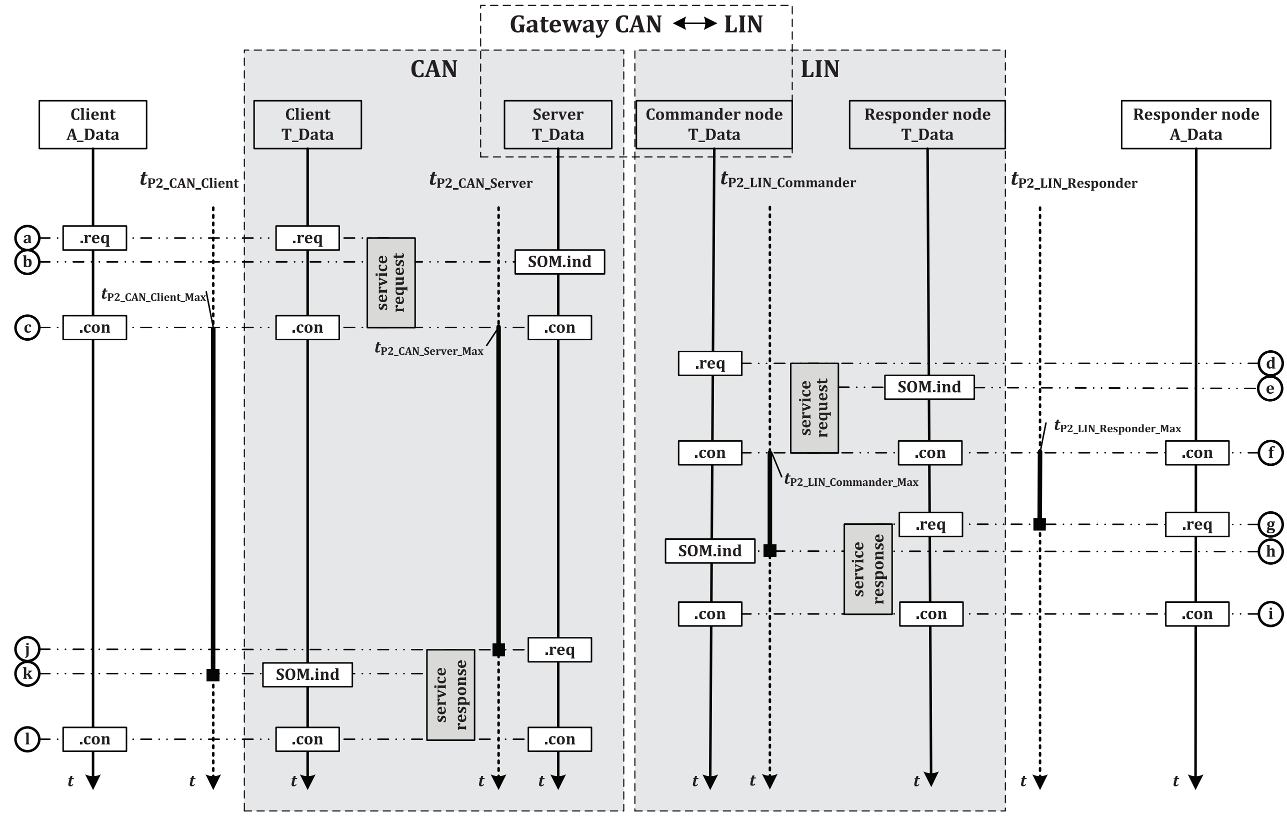
Timing on LIN according to ISO 14229-2 is a default behaviour. It is the vehicle manufacturer's responsibility to specify the $t_{\mathtt{P2\_LIN\_Commander\_Max}}$ timing value if the default values do not meet the timing requirements of the selected in-vehicle network architecture.

Depending on the choice of implementation, there are two possibilities:

— the message reception in the server (gateway) received on the backbone network is mirrored on the LIN. The commander node transmission on the LIN network starts after the transport layer indicates the transmission of the message via a `T_Data.req`; or

— the server waits until it receives a `T_Data.con` and then the commander node starts the transmission of the message to the responder node by issuing a `T_Data.req` to its transport layer.

Figure 4 shows the timing sequence chart with store and forward method.



Key

a Client T_Data.req: the diagnostic application of the client starts the transmission of the request message by issuing a `T_Data.req` to its transport layer.

b Server T_DataSOM.ind: the start of the request message is indicated in the server via the `T_DataSOM.ind`.

c Server T_Data.con: the completion of the request message is indicated in the server via the `T_Data.con`. Now the response timing as specified in ISO 14229-2 applies. The server loads the $t_{P2\_CAN\_Server\_Max}$ timer value into the $t_{P2\_CAN\_Server}$ timer.

Client T_Data.con: the completion of the request message is indicated in the client via `T_Data.con`. Now the response timing as specified in ISO 14229-2 applies. The client loads the $t_{P2\_CAN\_Client\_Max}$ timer value into the $t_{P2\_CAN\_Client}$ timer.

d Commander node T_Data.req: the commander node starts the transmission of the request message by issuing a `T_Data.req` to its transport layer.

e Responder node T_DataSOM.ind: the start of the request message is indicated in the responder node via the `T_DataSOM.ind`.

f Responder node T_Data.con: the completion of the request message is indicated in the responder node via the `T_Data.con`. Now the response timing as specified in ISO 14229-2 applies. The responder node loads the $t_{P2\_LIN\_Responder\_Max}$ timer value into the $t_{P2\_LIN\_Responder}$ timer.

Commander node T_Data.con: the completion of the transmission of the response message is indicated in the commander node via `T_Data.con`. Now the server starts its $t_{P2\_LIN\_Commander\_Max}$ timer.

g Responder node T_Data.req: the start of the response message is indicated in the responder node via the `T_Data.req`.

h Commander node T_DataSOM.ind: the start of the response message is indicated in the commander node via the `T_DataSOM.ind`. The commander node stops the $t_{P2\_LIN\_Commander}$ timer.

i Responder node T_Data.con: the completion of the response message is indicated in the Responder node via `T_Data.con`.

Commander node T_Data.con: The completion of the response message is indicated in the commander node via `T_Data.con`.

j Server A_Data.req: the server starts the transmission of the response message by issuing a `T_Data.req` to its transport layer.

The server stops the $t_{P2\_CAN\_Server}$ timer.

[k] Client T_DataSOM.ind: the start of the response message is indicated in the client via the `T_DataSOM.ind`.

[l] Server T_Data.con: the completion of the response message is indicated in the server via the `T_Data.con`.

Client T_Data.con: the completion of the response message is indicated in the client via `T_Data.con`.

Figure 4 — Timing sequence chart with store and forward method

9 Presentation layer

The presentation layer specification is not in the scope of this document.

10 Session layer

10.1 Service primitive parameter definition

This document is part of the ISO 14229 series and therefore the service primitive parameter implementation follows the ISO 14229-2 specification.

REQ	5.1 UDSonLIN – Service primitive parameter definition
The service primitive parameters shall be implemented as specified in ISO 14229-2.	

10.2 S_Data.req, S_Data.ind, and S_Data.conf service interface

This document is part of the ISO 14229 series and therefore the service interface implementation follows the ISO 14229-1 specification.

REQ	5.2 UDSonLIN – S_Data.req, S_Data.ind, and S_Data.conf service interface
The `A_Data.req`, `A_Data.ind`, and `A_Data.conf` service interface shall be implemented as specified in ISO 14229-2.	

11 Transport layer

11.1 General

This document is part of the ISO 14229 series and therefore the service primitive parameter implementation follows the ISO 17987-2 specification.

NOTE The transport/network layer services are used to perform the application layer and diagnostic session management timing.

11.2 Service primitive parameters

This document follows the requirements of the transport layer and network layer as specified in ISO 17987-2 specification.

REQ	4.1 UDSonLIN – Service primitive parameters
The service primitive parameters shall be implemented as specified in ISO 17987-2.	

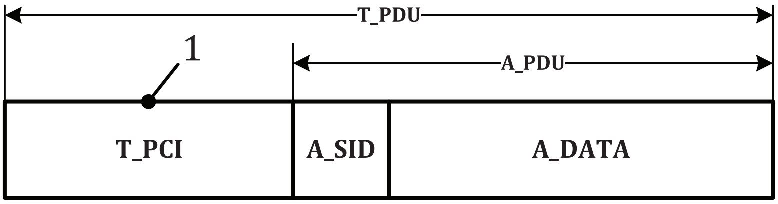
11.3 T_Data.req, T_Data.ind, and T_Data.conf service interface

This document supports the service interface definition as specified in ISO 17987-2.

REQ	4.2 UDSonLIN – T_Data.req, T_Data.ind, and T_Data.conf service interface
The `T_Data.req`, `T_Data.ind`, and `T_Data.conf` service interface implementation shall be implemented as specified in ISO 17987-2.	

11.4 T_PDU definition

Figure 5 shows the T_PDU.



Key

1 transport layer protocol control information

Figure 5 — T_PDU definition

11.5 LIN transport and network layer interface adaptation

11.5.1 Mapping of data link independent service primitives onto LIN data link-dependent service primitives

REQ	4.3 UDSonLIN – Mapping of data link independent service primitives onto LIN data link-dependent service primitives
The parameter mapping interface shall be implemented as specified in ISO 17987-2, DoCAN transport protocol and network layer services, and the session layer services as specified in ISO 14229-2 for the transmission and reception of diagnostic messages. The parameter mapping interface shall be implemented as specified in Table 4.	

Table 4 — Mapping of T_PDU service primitives onto network N_PDU service primitives

Session to transport/network layer service primitives (data link independent according to ISO 14229-2)	LIN network layer service primitives (data link dependent according to ISO 17987-2)
`T_Data.indication`	`N_USData.indication`
`T_DataSOM.indication`	`N_USDataFF.indication`
`T_Data.confirm`	`N_USData.confirm`
`T_Data.request`	`N_USData.request`

11.5.2 Mapping of T_PDU onto N_PDU

The service primitive parameters of the application layer PDU defined to request the transmission of a diagnostic service request message (client) and response message (server) are mapped onto the parameters of the transport/network layer PDU.

REQ	4.4 UDSonLIN – Mapping of T_PDU onto N_PDU
The service primitive parameter mapping of T_PDU onto N_PDU shall be implemented as specified in Table 5.	

Table 5 — Mapping of T_PDU parameter onto the network N_PDU parameter

T_PDU parameter (data link independent according to ISO 14229-2)	N_PDU parameter (LIN data link dependent according to ISO 17987-2)
T_Ptype	N_Ptype
T_SA	not applicable
T_TA	N_TA(NAD)
T_TAtype	N_TAtype
T_AE	not applicable
T_Data[ ]	N_Data[ ]
T_Length	N_Length
T_Result	N_Result

[a] If Ptype = diagnostics, then the address information shall consist of the parameters SA, TA, and TAtype.
If Ptype = remote diagnostics, then the address information shall consist of the parameters SA, TA, TAtype, and AE.

12 Network layer

12.1 Service primitive parameter definition

This document specifies the implementation of the network layer of the ISO 17987-2 and therefore the service primitive parameter implementation follows the ISO 17987-2 specification.

REQ	3.1 UDSonLIN – Service primitive parameter definition
The service primitive parameters shall be implemented as specified in ISO 17987-2.	

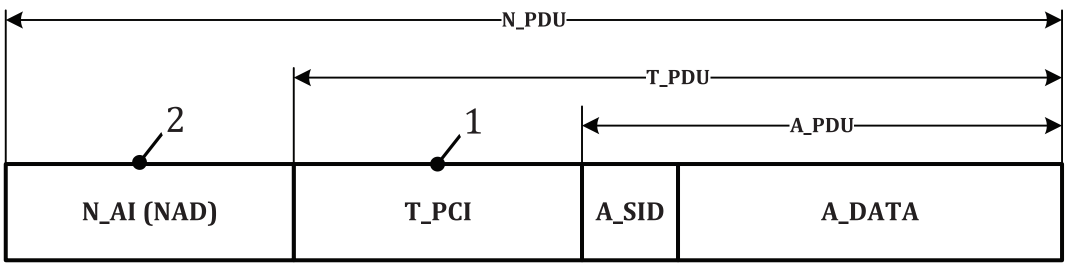
12.2 N_Data.req, N_Data.ind, and N_Data.conf service interface

This document follows the service interface definition as specified in ISO 17987-2.

REQ	3.2 UDSonLIN – N_Data.req, N_Data.ind, and N_Data.conf service interface
The N_Data.req, N_Data.ind, and N_Data.conf service interface implementation shall be implemented as specified in ISO 17987-2.	

12.3 N_PDU definition

Figure 6 shows the N_PDU.



Key

1 transport layer protocol control information

2 network layer address information (NAD)

Figure 6 — N_PDU definition

12.4 N_TAtype service primitive parameter

The parameter N_TAtype is a configuration attribute to the N_TA parameter. It is used to encode the communication by the peer entities.

REQ	3.3 UDSonLIN – N_TAtype service primitive parameter
The N_TAtype parameter shall be of data type Enum and shall be used to identify the target address type.	

ISO 14229-1-defined services use the same request and response message format.

REQ	3.4 UDSonLIN – Same N_TAtype request and associated response message format
If a server implements a particular N_TAtype for a received request message then it shall use the same N_TAtype as specified in Table 6 in the corresponding response message.	

Table 6 — N_Data service primitive parameter mapping

NL	.req	.ind	.conf	Description
N_AI[N_TAtype]	X[a]	X[a]	—	N_TAtype: physical
N_AI[TA]	X[a]	X[a]	—	target address to be added to PDU as N_AI(NAD) if N_TAtype: physical

12.5 LIN responder node requirements

This document specifies LIN responder node network layer requirements depending on the diagnostic class.

REQ	3.5 UDSonLIN – Responder node diagnostic class I – No additional network layer requirements
There shall be no additional network layer requirements defined for a LIN responder node which is compatible to diagnostic class I.	

REQ	3.6 UDSonLIN – Responder node diagnostic class II – Conform to ISO 17987-2
Each diagnostic class II compatible LIN responder node shall implement the network layer requirements as specified in ISO 17987-2.	

REQ	3.7 UDSonLIN – Responder node diagnostic class II – Fixed node address
Each diagnostic class II LIN responder node shall be assigned a fixed node address (NAD) during system design time.	

For LIN clusters using dynamic NAD assignment during network communication start-up, REQ 3.7 implies that after NAD assignment is finished, all LIN responder nodes have their NADs assigned as specified during system design time.

REQ	3.8 UDSonLIN – Responder node diagnostic class II – Ignore NAD $7E_{16}$ as broadcast address
Each diagnostic class II LIN responder node shall ignore NAD $7E_{16}$ as broadcast (functional) address for diagnostic communication on the LIN cluster.	

REQ	3.9 UDSonLIN – Responder node diagnostic class III – Conform to ISO 17987-2
Each diagnostic class III LIN responder node shall implement the network layer requirements as specified in ISO 17987-2.	

REQ	3.10 UDSonLIN – Responder node diagnostic class III – Fixed node address
Each diagnostic class III LIN responder node shall be assigned a fixed node address (NAD) during system design time.	

For LIN clusters using dynamic NAD assignment during network communication start-up, REQ 3.10 implies that after NAD assignment is finished, all LIN responder nodes have their NADs assigned as specified during system design time.

REQ	3.11 UDSonLIN – Responder node diagnostic class III – Ignore NAD $7E_{16}$ as broadcast address
Each diagnostic class II LIN responder node shall ignore NAD $7E_{16}$ as broadcast (functional) address for diagnostic communication on the LIN cluster.	

12.6 LIN commander node requirements

12.6.1 Network address requirements

This document specifies the LIN commander node network layer requirements.

REQ	3.12 UDSonLIN – Commander node diagnostic class III – Conform to ISO 17987-2
Each diagnostic class III compatible LIN commander node shall implement the network layer requirements as specified in ISO 17987-2.	

REQ	3.13 UDSonLIN – Commander node diagnostic class III – Usage of NAD assignment in LIN cluster
If a dynamic NAD assignment shall be used on a LIN cluster, the LIN commander node shall ensure that after communication start-up, all LIN responder nodes shall have their NADs assigned as specified according to REQ 3.7 and REQ 3.10. The start-up configuration time necessary to complete NAD assignment shall be documented in the diagnostic specification of the commander node.	

This implies that diagnostic communication between external diagnostic test equipment and a LIN responder node might not be possible until the LIN commander node has completed the NAD assignment sequence.

REQ	3.14 UDSonLIN – Commander node diagnostic class III – Assignment of subnet number
The LIN clusters connected to the LIN commander node shall be assigned the subnet number 1_{16} which is used by the diagnostic service CommunicationControl.	

12.6.2 Use of functional addressing

Functional addressing is used by the external test equipment (client to LIN commander node) if the client does not know about the LIN architecture.

13 Data link layer

13.1 Service primitive parameter definition

This document follows the service primitive parameter specification of ISO 17987-4.

REQ	2.1 UDSonLIN – Service primitive parameter definition
The service primitive parameters shall be implemented as specified in ISO 17987-4.	

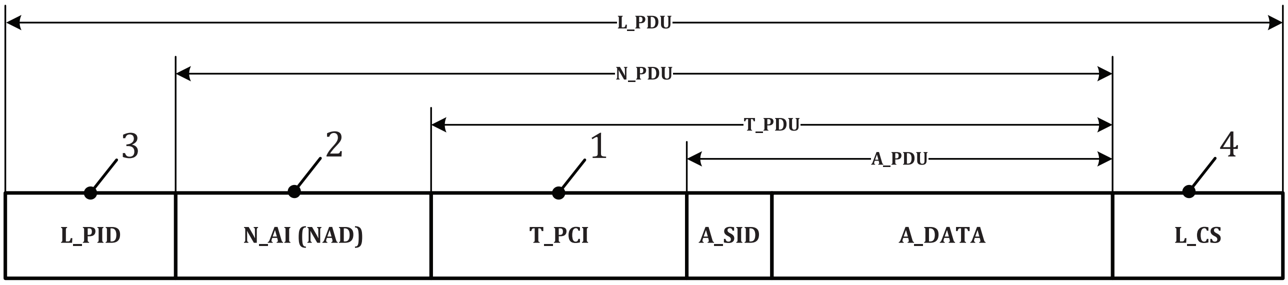
13.2 L_Data.req, L_Data.ind, and L_Data.conf service interface

This document implements the service interface of the data link layer in accordance with ISO 17987-3.

REQ	2.2 UDSonLIN – L_Data.req, L_Data.ind, and L_Data.conf service interface
The `L_Data.req`, `L_Data.ind`, and `L_Data.conf` service interface shall be implemented as specified in ISO 17987-3.	

13.3 L_PDU definition

Figure 7 shows the L_PDU.



Key

1 transport layer protocol control information

2 network layer address information (NAD)

3 LIN data link layer protected identifier

4 LIN data link layer checksum

Figure 7 — L_PDU definition

13.4 L_PID definition

REQ	2.3 UDSonLIN – L_PID definition
The `L_PID` protected identifier shall be implemented as specified in ISO 17987-3.	

13.5 L_CS definition

REQ	2.4 UDSonLIN – L_CS definition
The `L_CS` checksum shall be implemented as specified in ISO 17987-3.	

14 Physical layer

The physical layer specification is not in the scope of this document.

Bibliography

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

[2] ISO/IEC 10731:1994, *Information technology — Open Systems Interconnection — Basic Reference Model — Conventions for the definition of OSI services*

[3] ISO/TR 17987-5, *Road vehicles — Local Interconnect Network (LIN) — Part 5: Application programmers interface (API)*

ICS 43.180

Price based on 18 pages