INTERNATIONAL STANDARD

ISO 23150

First edition
2021-05

Road vehicles — Data communication between sensors and data fusion unit for automated driving functions — Logical interface

Véhicules routiers — Communication de données entre capteurs et unité de fusion de données pour les fonctions de conduite automatisée — Interface logique

Reference number
ISO 23150:2021(E)

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

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.

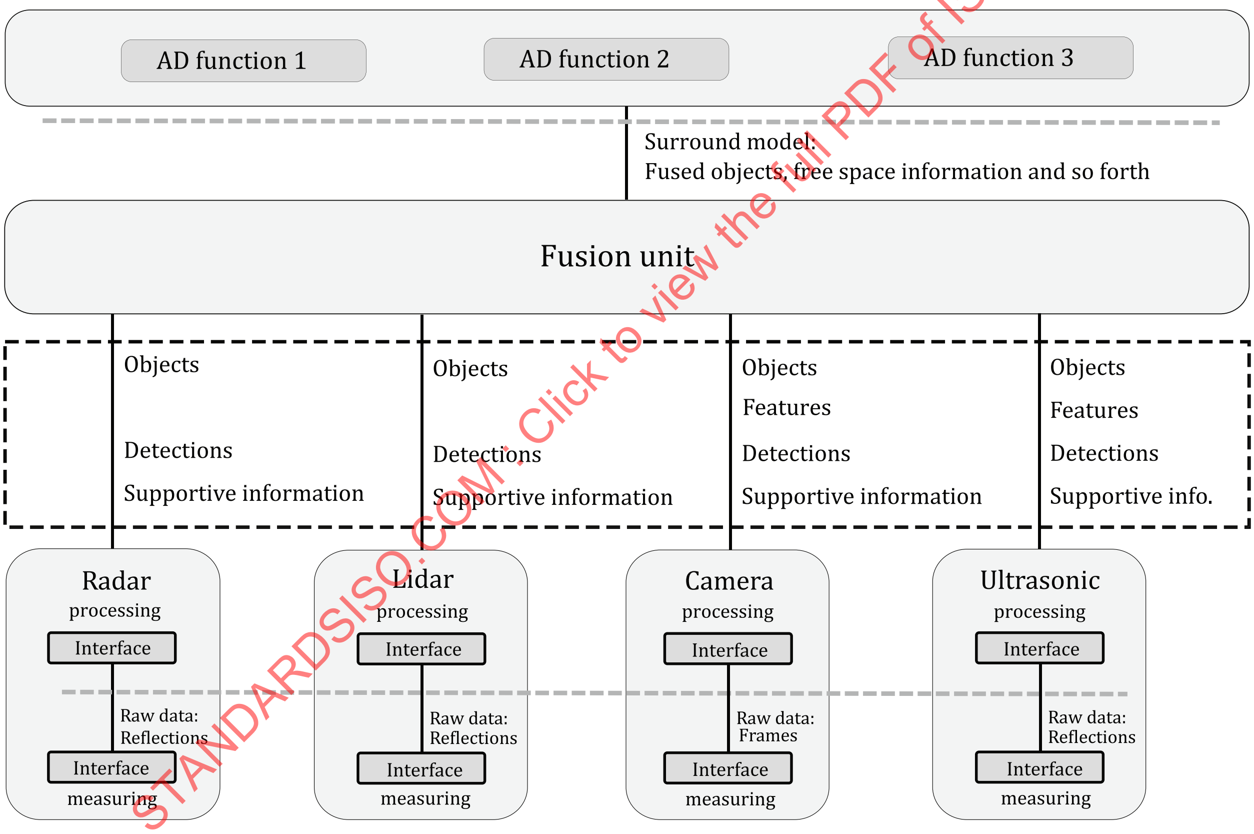
This document was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 31, *Data communication*.

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

Highly-automated driving (AD) functions for road vehicles require a situation awareness of the surroundings of the vehicle and a, preferably, comprehensive scene understanding. For the fast and reliable recognition of real-world objects, a sensor suite is necessary to provide information for the fusion unit. Utilisation of different sensor technologies like radar, lidar, camera and ultrasonic with different detection capabilities is indispensable to ensure both complementary and redundant information. The fusion unit analyses and evaluates the different sensor signals and finally generates a dynamic surround model with sufficient scene understanding.

While current partly-automated functions utilise only particular objects (for example, vehicles, pedestrians, road markings) to generate a simple surround model, it is necessary for future highly-automated driving functions to merge not only the recognised objects but also to include other sensor-specific properties and characteristics of these objects for the generation of a coherent model of the surroundings. To minimise the development efforts for the sensors and the fusion unit and to maximise the reusability of development and validation efforts for the different functions on the sensor and fusion unit side, a standardised logical interface layer between the sensor suite and the fusion unit is worthwhile and beneficial for both the sensor and the system supplier.



Key

1 logical interface layer between the fusion unit and automated driving functions

2 logical interface layer between a single sensor as well as a single sensor cluster and the fusion unit

3 interface layer on raw data level of a sensor's sensing element

Figure 1 — Architecture: sensors/sensor clusters – fusion unit – automated driving functions

The logical interface layer between a single sensor as well as a single sensor cluster and the fusion unit [see key 2 in Figure 1] addresses the encapsulation of technical complexity as well as objects, features and detections to enable object-level, feature-level and detection-level fusion. Additional supportive information of the sensor as well as the sensor cluster will supplement the data for the fusion unit.

Road vehicles — Data communication between sensors and data fusion unit for automated driving functions — Logical interface

1 Scope

This document is applicable to road vehicles with automated driving functions. The document specifies the logical interface between in-vehicle environmental perception sensors (for example, radar, lidar, camera, ultrasonic) and the fusion unit which generates a surround model and interprets the scene around the vehicle based on the sensor data. The interface is described in a modular and semantic representation and provides information on object level (for example, potentially moving objects, road objects, static objects) as well as information on feature and detection levels based on sensor technology specific information. Further supportive information is available.

This document does not provide electrical and mechanical interface specifications. Raw data interfaces are also excluded.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

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

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1 Architectural components

3.1.1
fusion
act of uniting *signals* (3.3.1) from two or more *sensors* (3.1.5) as well as *sensor clusters* (3.1.6) to create a *surround model* (3.1.7)

3.1.2
fusion unit
computing unit where the *fusion* (3.1.1) of *sensor* (3.1.5) data as well as a *sensor cluster* (3.1.6) data is performed

3.1.3
interface
shared boundary between two functional units, defined by various characteristics pertaining to the functions, physical interconnections, *signal* (3.3.1) exchanges and other characteristics of the units, as appropriate

[SOURCE: ISO/IEC 2382:2015, 2124351, modified — Notes to entry have been removed.]

3.1.4
logical interface
interface (3.1.3) between a *sensor* (3.1.5) as well as a *sensor cluster* (3.1.6) and the *fusion unit* (3.1.2), defined by logical characteristics

Note 1 to entry: Logical means a semantic description of the interface.

Note 2 to entry: Mechanical and electrical interfaces are excluded.

Note 3 to entry: This document uses the term interface as a shortcut for the term logical interfaces.

3.1.5
sensor
in-vehicle unit which detects entities external of the vehicle with preprocessing capabilities serving at least one *logical interface* (3.1.4)

Note 1 to entry: A sensor may use one or more sensing elements.

3.1.6
sensor cluster
group of *sensors* (3.1.5) of the same technology serving a common *logical interface* (3.1.4)

Note 1 to entry: A sensor cluster can exceptionally consist of only one sensor.

EXAMPLE A stereo camera, a surround-view camera, an ultrasonic sensor array, a corner radar system.

3.1.7
surround model
representation of the real world adjacent to the ego-vehicle

3.2 Level of detail terms

3.2.1
detection
sensor technology specific entity represented in the *sensor coordinate system* (3.7.18) based on a single *measurement* (3.4.1) of a *sensor* (3.1.5)

Note 1 to entry: A small amount of history can be used for some detection *signals* (3.3.1), for example, model-free filtering may be used in track-before-detect algorithms.

3.2.2
detection level
set of *logical interfaces* (3.1.4) that provides *detections* (3.2.1)

3.2.3
feature
sensor technology specific entity represented in the *vehicle coordinate system* (3.7.16) based on multiple *measurements* (3.4.1)

Note 1 to entry: Multiple measurements can originate from a *sensor cluster* (3.1.6).

Note 2 to entry: Multiple measurements can originate from multiple *measurement cycles* (3.4.2).

Note 3 to entry: The term feature is used in this document not as function or group of functions as specified in ISO/SAE PAS 22736[1].

3.2.4
feature level
set of *logical interfaces* (3.1.4) that provides *features* (3.2.3)

1) Under preparation. Stage at the time of publication: ISO/SAE DPAS 22736:2021.

3.2.5
object
representation of a real-world entity with defined boundaries and characteristics in the *vehicle coordinate system* (3.7.16)

Note 1 to entry: The geometric description of the object is in the vehicle coordinate system.

Note 2 to entry: Object *signals* (3.3.1) are basically sensor technology independent. Sensor technology specific signals may extend the object signals.

EXAMPLE A *potentially moving object* (3.6.1), a *road object* (3.6.2), a *static object* (3.6.3).

3.2.6
object level
set of *logical interfaces* (3.1.4) that provides *objects* (3.2.5)

3.3 Structure terms

3.3.1
signal
entity consisting of one or more values and which is part of a *logical interface* (3.1.4)

3.3.2
logical signal group
grouping of *signals* (3.3.1) that has a logical relationship and a name for the grouping

3.3.3
classification
attribute-based differentiation

Note 1 to entry: An attribute is defined by a list of enumerators.

3.4 Measurement terms

3.4.1
measurement
measuring and processing result of a *measurement cycle* (3.4.2)

3.4.2
measurement cycle
time period from the start of a data acquisition event to the start of the next data acquisition event

Note 1 to entry: A measurement cycle of one *sensor* (3.1.5) is a consistent view of an observed scene and not overlapping in time.

3.4.3
accuracy
closeness of agreement between a measured quantity value and a true quantity value

Note 1 to entry: The concept accuracy is not a quantity and is not given a numerical quantity value. A *measurement* (3.4.1) is said to be more accurate when it offers a smaller *error* (3.4.6).

Note 2 to entry: The term accuracy should not be used for *trueness* (3.4.4) and the term *precision* (3.4.5) should not be used for accuracy, which, however, is related to both these concepts.

Note 3 to entry: Accuracy is sometimes understood as closeness of agreement between measured quantity values that are being attributed to the measurand.

[SOURCE: ISO/IEC Guide 99:2007, 2.13, modified — The terms "measurement accuracy" and "accuracy of measurement" were deleted and the Notes to entry have been adapted.]

3.4.4
trueness
closeness of agreement between the average of an infinite number of replicated measured quantity values and a reference quantity value

Note 1 to entry: Trueness is not a quantity and thus cannot be expressed numerically, but measures for closeness of agreement are given in the ISO 5725 series.

Note 2 to entry: Trueness is inversely related to systematic error, but is not related to random error.

Note 3 to entry: The term *accuracy* (3.4.3) should not be used for trueness.

[SOURCE: ISO/IEC Guide 99:2007, 2.14, modified — The terms "measurement trueness" and "trueness of measurement" were deleted and the Notes to entry have been adapted.]

3.4.5
precision
closeness of agreement between indications or measured quantity values obtained by replicate *measurements* (3.4.1) on the same or similar measurands under specified conditions

Note 1 to entry: Precision is usually expressed numerically by measures of imprecision, such as standard deviation, variance, or coefficient of variation under the specified conditions of measurement.

Note 2 to entry: The specified conditions can be, for example, repeatability conditions of measurement, intermediate precision conditions of measurement, or reproducibility conditions of measurement (see ISO 5725-1:1994).

Note 3 to entry: Precision is used to define measurement repeatability, intermediate measurement precision and measurement reproducibility.

Note 4 to entry: Sometimes precision is erroneously used to mean *accuracy* (3.4.3).

Note 5 to entry: Precision is inversely related to random error, but is not related to systematic error.

[SOURCE: ISO/IEC Guide 99:2007, 2.15, modified — The term "measurement precision" was deleted, the word “objects” was replaced by “measurands”, the Notes to entry have been adapted and Note 5 to entry has been added.]

3.4.6
error
measured quantity value minus a reference quantity value

Note 1 to entry: The concept of error can be used both:

Note 2 to entry: a) when there is a single reference quantity value to refer to, which occurs if a calibration is made by means of a measurement standard with a measured quantity value having a negligible measurement uncertainty or if a conventional quantity value is given, in which case the error is known, and

Note 3 to entry: b) if a measurand is supposed to be represented by a unique true quantity value or a set of true quantity values of negligible range, in which case the error is not known.

Note 4 to entry: Error should not be confused with production error or mistake.

[SOURCE: ISO/IEC Guide 99:2007, 2.16, modified — The terms "measurement error" and "error of measurement" were deleted and the Notes to entry have been adapted.]

3.5 Requirement level terms

3.5.1
conditional
required under certain specified conditions

Note 1 to entry: One of three obligation statuses applied to a *requirement level* (3.5.4) of a *logical interface* (3.1.4) specification, indicating the conditions under which the *signal* (3.3.1) or *logical signal group* (3.3.2) is required. In other cases, the signal or logical signal group is optional. See also *mandatory* (3.5.2) and *optional* (3.5.3).

[SOURCE: ISO/IEC 11179-3:2013, 3.2.22, modified — Notes to entry have been adapted.]

3.5.2
mandatory
always required

Note 1 to entry: One of three obligation statuses applied to a *requirement level* (3.5.4) of a *logical interface* (3.1.4) specification, indicating the conditions under which the *signal* (3.3.1) or *logical signal group* (3.3.2) is required. See also *conditional* (3.5.1) and *optional* (3.5.3).

[SOURCE: ISO/IEC 11179-3:2013, 3.2.71, modified — Notes to entry have been adapted.]

3.5.3
optional
permitted but not required

Note 1 to entry: One of three obligation statuses applied to a *requirement level* (3.5.4) of a *logical interface* (3.1.4) specification, indicating the conditions under which the *signal* (3.3.1) or *logical signal group* (3.3.2) is required. See also *conditional* (3.5.1) and *mandatory* (3.5.2).

[SOURCE: ISO/IEC 11179-3:2013, 3.2.89, modified — Notes to entry have been adapted.]

3.5.4
requirement level
definition of the obligation status of a *logical interface's* (3.1.4) *logical signal group* (3.3.2), *signal* (3.3.1) as well as a signal's identifier or signal's enumerator

Note 1 to entry: Each requirement level entry has one of three possible obligation statuses applied: *conditional* (3.5.1), *mandatory* (3.5.2) or *optional* (3.5.3).

3.6 Road user relevant entity types

3.6.1
potentially moving object
real-world entity which potentially can move and is relevant for driving situations

Note 1 to entry: A representation of a potentially moving object is part of *object level* (3.2.6) *logical interfaces* (3.1.4).

EXAMPLE A vehicle, a bicycle, a pedestrian, an obstacle.

3.6.2
road object
marking or structure of a road which is relevant for driving situations

Note 1 to entry: A representation of a road object is part of *object level* (3.2.6) *logical interfaces* (3.1.4).

EXAMPLE A *road marking* (3.6.2.1), a *road boundary* (3.6.2.2), the *road surface* (3.6.2.3).

3.6.2.1
road marking
line, symbol or other mark on the surface of a road or a structure intended to limit, regulate, warn, guide or inform road users

Note 1 to entry: Other marks could be text, numbers, arrows or combinations.

EXAMPLE A lane marking, Botts' dots.

[SOURCE: ISO 6707-1:2020, 3.3.5.80, modified — "user" was modified to "road users", "a road surface" was modified to "the surface of a road" and the Note 1 to entry and example have been added.]

3.6.2.2
road boundary
structure that limits the road

EXAMPLE A curb stone, a guard rail, the end of the surface of the road.

3.6.2.3
road surface
surface supporting the tyre and providing friction necessary to generate shear forces in the *road plane* (3.7.6)

Note 1 to entry: The surface may be flat, curved, undulated or of other shape.

[SOURCE: ISO 8855:2011, 2.6]

3.6.3
static object
real-world stationary entity which can be used for information and/or localisation

Note 1 to entry: A representation of a static object is part of *object level* (3.2.6) *logical interfaces* (3.1.4).

EXAMPLE A *general landmark* (3.6.3.1), a *traffic sign* (3.6.3.2), a *traffic light* (3.6.3.3).

3.6.3.1
general landmark
real-world stationary entity which can be used for localisation

Note 1 to entry: A stationary *traffic sign* (3.6.3.2) or *traffic light* (3.6.3.3) is also regarded as a general landmark.

EXAMPLE A building, a tunnel, a bridge, a sign gantry structure, a tree.

3.6.3.2
traffic sign
traffic relevant, authorised sign that limits, regulates, warns, guides or informs road users

Note 1 to entry: One traffic sign usually consists of one *main sign* (3.6.3.2.1) and none, one or several *supplementary signs* (3.6.3.2.2).

EXAMPLE A speed limit which is restricted for trucks.

3.6.3.2.1
main sign
traffic sign (3.6.3.2) which gives a general message, obtained by a combination of colour and geometric shape and which, by the addition of a graphical symbol or text, gives a particular message for road users

[SOURCE: ISO 3864-1:2011, 3.12, modified — The original term was "safety sign", "sign" has been replaced by "traffic sign" and the phrases "or text" and "for road users" have been added to the definition.]

3.6.3.2.2
supplementary sign
traffic sign (3.6.3.2) that is supportive of a *main sign* (3.6.3.2.1) and the main purpose of which is to provide additional clarification

[SOURCE: ISO 3864-1:2011, 3.14, modified — "traffic sign" now replaces "sign" and "main sign" replaces "traffic sign".]

3.6.3.3
traffic light
traffic relevant, official lights

Note 1 to entry: One traffic light consists of one or several light spots with different light colours and shapes.

EXAMPLE A pedestrian traffic light.

3.7 Axis and coordinate system terms

3.7.1
reference frame
geometric environment in which all points remain fixed with respect to each other at all times

[SOURCE: ISO 8855:2011, 2.1]

3.7.2
axis system
set of three orthogonal directions associated with *X*, *Y* and *Z* axes

Note 1 to entry: A right-handed axis system is assumed throughout this document, where: $\vec{Z} = \vec{X} \times \vec{Y}$.

[SOURCE: ISO 8855:2011, 2.3, modified — Notes to entry have been adapted.]

3.7.3
coordinate system
numbering convention used to assign a unique ordered trio of values to each point in a *reference frame* (3.7.1) and which consists of an *axis system* (3.7.2) plus an origin point

[SOURCE: ISO 8855:2011, 2.4, modified — "(*x*, *y*, *z*)" has been removed from the definition.]

3.7.4
cartesian coordinate system
set of numerical coordinates (*x*, *y*, *z*), which are the signed distances to the *YZ*-, *ZX*- and *XY*-planes

3.7.5
spherical coordinate system
set of two angles and a distance vector associated with radial distance, azimuth and elevation

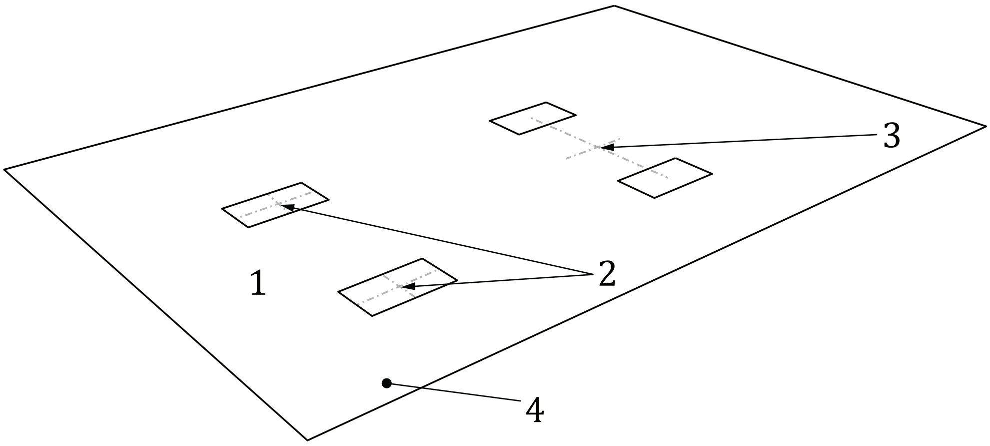
Note 1 to entry: The azimuth angle is the angle in *XY*-plane of the *axis system* (3.7.2) counted from the *X*-axis. The elevation angle is the angle from the azimuth direction in the *XY*-plane of the axis system towards the direction of the distance vector, that is *XY*-plane has an elevation angle = 0 rad.

Note 2 to entry: The angles of the spherical coordinate system have increasing values in counter-clockwise direction.

3.7.6
road plane
plane representing the *road surface* (3.6.2.3) within the front tyre contact patches and the *vehicle road-level reference point* (3.7.13)

Note 1 to entry: See Figure 2.

Note 2 to entry: For tyre contact patches, see ISO 8855:2011, 4.1.5.



Key

1 vehicle front

2 vehicle's front tyre contact patches

3 *vehicle road-level reference point* (3.7.13)

4 vehicle *road plane* (3.7.6)

Figure 2 — Road plane

[SOURCE: ISO 8855:2011, 2.7, modified — The phrase "and the vehicle road-level reference point" and the figure have been added, and the Notes to entry have been modified.]

3.7.7
road level
point related to a *road plane* (3.7.6)

3.7.8
vehicle unsprung mass
unsprung mass
mass that is not carried by the suspension, but is supported directly by the tyres

[SOURCE: ISO 8855:2011, 4.11, modified — The term "vehicle unsprung mass" has been added.]

3.7.9
vehicle sprung mass
sprung mass
mass that is supported by the suspension, that is the total vehicle mass less the *vehicle unsprung mass* (3.7.8)

[SOURCE: ISO 8855:2011, 4.12, modified — The term vehicle sprung mass has been added and Note 1 to entry has been removed.]

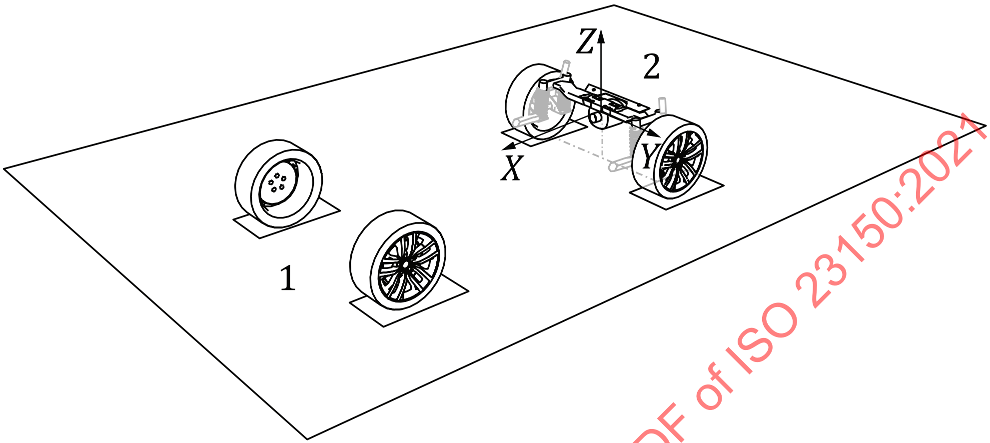
3.7.10
vehicle rear-axle reference point
point fixed in the *vehicle sprung mass* (3.7.9) and located at the centre of the rear-axle

3.7.11
vehicle sprung mass axis system
axis system (3.7.2) fixed in the *reference frame* (3.7.1) of the *vehicle sprung mass* (3.7.9), so that the *X*-axis is substantially horizontal and forwards (with the vehicle at rest), and is parallel to the vehicle's longitudinal plane of symmetry, and the *Y*-axis is perpendicular to the vehicle's longitudinal plane of symmetry and points to the left with the *Z*-axis pointing upward

3.7.12
vehicle rear-axle coordinate system
coordinate system (3.7.3) based on the *vehicle sprung mass axis system* (3.7.11) with the origin located at the *vehicle rear-axle reference point* (3.7.10)

Note 1 to entry: The vehicle rear-axle coordinate system is a *vehicle coordinate system* (3.7.16).

Note 2 to entry: See Figure 3.



Key

1 vehicle front

2 *vehicle rear-axle reference point* (3.7.10)

Figure 3 — Vehicle rear-axle coordinate system

3.7.13
vehicle road-level reference point
point at *road level* (3.7.7) located in the middle of the rear tyre contact patches

Note 1 to entry: For tyre contact patches, see ISO 8855:2011, 4.1.5.

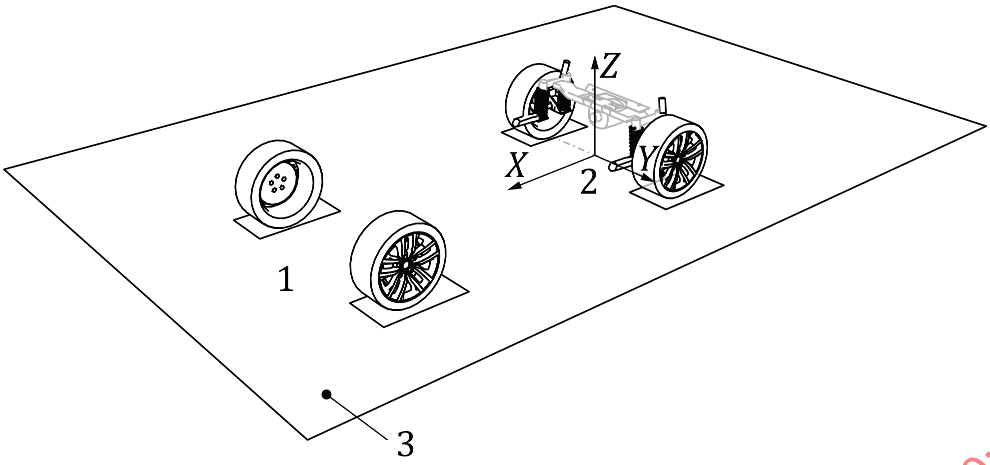
3.7.14
vehicle road-level axis system
axis system (3.7.2) fixed in the *reference frame* (3.7.1) of the *vehicle unsprung mass* (3.7.8), so that the *X*-axis is parallel to the vehicle's longitudinal plane of symmetry and points into forward moving direction and the *Y*-axis is perpendicular to the vehicle's longitudinal plane of symmetry and points to the left with the *Z*-axis pointing upward

Note 1 to entry: Vehicle road-level axis system's *XY*-plane is parallel to the ego-vehicle's *road plane* (3.7.6).

3.7.15
vehicle road-level coordinate system
coordinate system (3.7.3) based on the *vehicle road-level axis system* (3.7.14) with the origin located at the *vehicle road-level reference point* (3.7.13) at the vehicle *road level* (3.7.7)

Note 1 to entry: The vehicle road-level coordinate system is a *vehicle coordinate system* (3.7.16).

Note 2 to entry: See Figure 4.



Key

1 vehicle front

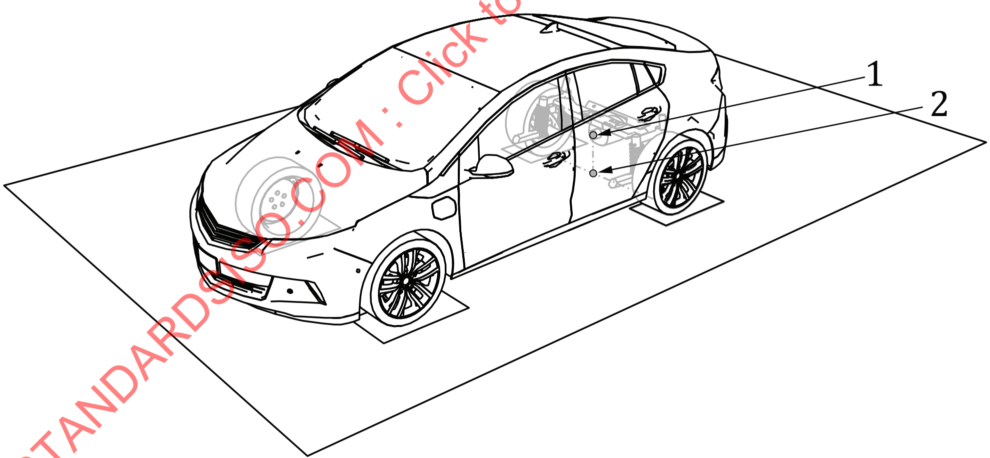
2 *vehicle road-level reference point* (3.7.13)

3 vehicle *road plane* (3.7.6)

Figure 4 — Vehicle road-level coordinate system

3.7.16
vehicle coordinate system
cartesian coordinate system (3.7.4) which is either the *vehicle rear-axle coordinate system* (3.7.12) or the *vehicle road-level coordinate system* (3.7.15)

Note 1 to entry: See Figure 5.



Key

1 *vehicle rear-axle reference point* (3.7.10)

2 *vehicle road-level reference point* (3.7.13)

Figure 5 — Vehicle coordinate systems

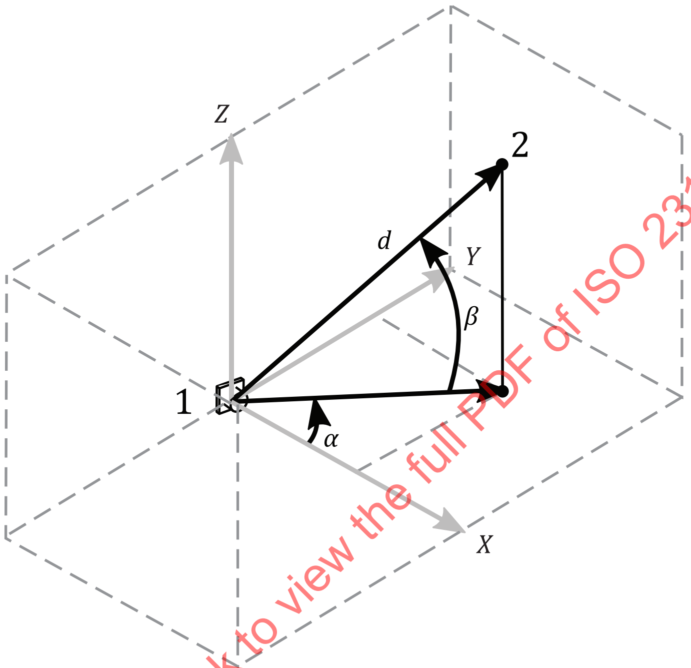
3.7.17
sensor axis system
axis system (3.7.2) fixed in the *reference frame* (3.7.1) of the *sensor* (3.1.5)

Note 1 to entry: The *X*-axis is in viewing direction of the sensor and the *Z*-axis pointing upward.

3.7.18
sensor coordinate system
spherical coordinate system (3.7.5) based on the *sensor axis system* (3.7.17) at a defined origin point of the *sensor* (3.1.5)

Note 1 to entry: The origin point of the sensor coordinate system has to be selected in a way that *detections* (3.2.1) could easily be specified in a spherical coordinate system. For example, the origin point of a camera sensor is the virtual projection centre of the camera's optics.

Note 2 to entry: See Figure 6.



Key

1 origin point of the *sensor* (3.1.5)
2 *detection* (3.2.1)
d radial distance
α azimuth
β elevation

Figure 6 — Sensor coordinate system

4 Abbreviated terms

AD	automated driving
C	conditional
CDI	camera detection interface
CFI	camera feature interface
D	dimensional
DLI	detection level interface
ECU	electronic control unit

FFT	fast Fourier transform
FLI	feature level interface
FOV	field-of-view
FWHM	full width at half maximum
HW	hardware
ID	identifier
IQR	interquartile range
IRI	international roughness index
LDI	lidar detection interface
LSG	logical signal group
M	mandatory
O	optional
OLI	object level interface
PMOI	potentially moving object interface
RCS	radar cross section
RDI	radar detection interface
RDOI	road object interface
RL	requirement level
SHII	sensor health information interface
SNR	signal to noise ratio
SOI	static object interface
SPI	sensor performance interface
SSI	supportive sensor interface
UDI	ultrasonic detection interface
UFI	ultrasonic feature interface

5 Structure of the interface description

5.1 General

The scope of the document are logical interfaces. Therefore, the following parts of the document use the term interface as a shortcut for the defined term "logical interface" (3.1.4). The interface descriptions have the following structure:

— a description of each interface by logical signal group (LSGs), signal names and requirement levels (RLs) (including conditional requirements) and additional options;

— the requirement level attribute “conditional” shall be an element used in this document;

— the logical interlink as well as complex dependencies between signals of an interface shall be defined in an interface’s subclause as a profile subclause of the interface;

— a detailed description of each signal in Annex A;

— the options and conditional requirements in Annex B.

5.2 Signal

Table 1 provides an overview of a signal used by an interface which is defined in subclause “Interface” (5.3). All signals are specified in Annex A. Signal names shall be unique. Similar signals on different interface levels have the interface level added to the signal name to guarantee a unique signal naming.

Table 1 — Signal: <Signal name>

Name	Name of the signal and optional list of signal identifiers Each signal respectively signal identifier is implemented by exactly one value.		
Description	Definition of the signal		
Value type	Definition of the signal’s value type and optionally the number of values, specifically the number of dimensions defined by the number of identifiers (default dimension: 1D vector)	**Unit**	Unit definition of the signal value In this document, the unit defines no application or network representation of the signal value (for example, a scaling factor, a fix point representation, linear or non-linear increasing values).

In general, for each signal with value type “enumeration”, a list of enumerators or exemplary enumerators is defined in an additional table (see Table 2) in Annex A. Each exemplary list of enumerators is informative. During the system design phase, the concrete enumeration lists shall be defined for the sensor as well as the sensor cluster, specifically based upon the exemplary enumeration list in Annex A. This can be done by using additional new, using a subset of exemplarily defined or adapted existing exemplary enumerators. The enumerator “<Enumeration>_Unknown” is always one exemplary enumerator and therefore, it is not added to the exemplary enumeration lists. At least one enumerator shall be defined for each implemented exemplary enumeration list.

Table 2 — Enumeration: <Signal name>

Name	**Description**	**RL enumerator**
Name of the enumerator	Definition of the enumerator	Requirement level of the enumerator (mandatory / conditional / optional) or “exemplary”

5.3 Interface

Table 3 provides an overview of a logical interface. An interface is comprised of signals. Signals can be part of a logical signal group (LSG) for ease of logical grouping.

Table 3 — Interface: <Interface name>

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
Name for logical signal group (LSG) – for a logical group of multiple signals	Requirement level for LSG: **m**andatory, **c**onditional or **o**ptional Conditional RL with a reference to a normative clause "Annex B"	Name of the signal, optional: {list of signal identifiers} with a reference to a normative clause "Annex A"	Requirement level for signal: **m**andatory, **c**onditional or **o**ptional is specified for the signal, including all signal identifiers, or for each signal identifier individually Conditional RL with a reference to a normative clause "Annex B"	Additional options for the signal: — a reference to a normative clause "Annex B" — a reference to a profile subclause

Areas in the interface table marked in grey indicate that the enclosed signals are part of a list of entities. The valid number of entities is defined at the beginning of the grey area (specifically, grey area header).

If an interface provides multiple interface entity types (for example, the road object interface (RDOI) provides road surface, road markings and road boundaries), the interface may require not all entity types at once [see specified in "Multiple entity types for one interface" (B.2.3)].

An LSG shall have at least one signal that is mandatory. If a signal of an optional LSG is used, all signals of the LSG shall be considered as defined on their signal requirement levels.

A conditional LSG requirement level shall define the conditions. Under all other conditions, the LSG is optional [see the definition "conditional" (3.5.1)].

In case of a list, the requirement level of the signal "Number of valid <...>" and its LSG are the base requirement level of all signals in the list.

Symbol "↪" indicates that LSG belongs to another LSG.

If only one signal requirement level is defined for a signal with signal identifiers, the requirement level of each signal identifier individually corresponds to the requirement level for the signal.

Signals with RL mandatory and static or fixed values, which are defined during the system design phase and/or are implicitly known by the sensor and the fusion unit may be optimised. In this case the applicability of optimisation methods of subclause "Options for interface optimisation" (B.1) are allowed which essentially means that, for example, the mandatory signals will not be explicitly transferred by the interface message.

5.4 Specific signal grouping

Table 4 provides an overview of a specific signal grouping, for example, an LSG, which refers a specific signal grouping of another subclause and extends the specific signal grouping to achieve similar interfaces.

Table 4 — Specific signal grouping: <Name for LSG(s)>

Signal	RL signal	Option
Name of the signal, optional: {list of signal identifiers} with a reference to a normative clause "Annex A"	Requirement level for signal: mandatory, conditional or optional is specified for the signal, including all signal identifiers, or for each signal identifier individually Conditional RL with a reference to a normative clause "Annex B" Optional: information of the RL for the signal's LSG, if it is not mandatory	Additional options for the signal: — a reference to a normative clause "Annex B" — a reference to a profile subclause

Symbol "↪" indicates that the signal belongs to a list of signals.

Unchanged, inherited signals, signal requirement levels and signal options of another specific subclause's signal grouping are marked in grey.

5.5 Profile

If multiple signals of an interface have an interface-specific semantic interdependency, this interdependency is described in a profile subclause and additional complex conditions are defined. Each signal which is contained in the profile has an annotation of the profile in the signal's option-column definition of the interface (see column "Option" in Table 3 and Table 4).

6 Logical interface from a sensor as well as a sensor cluster to a fusion unit

6.1 General

The aim of this document is to support different levels of fusion. Therefore, the document provides three logical interface levels (detection-, feature- and object level). Supportive sensor interfaces (SSIs) complement the interfaces of the three logical interface levels. Detections, features, objects and supportive sensor information are the basis for a fusion unit. A tracking algorithm within the sensor solely or the sensor cluster using their own information, results in objects with limited quality. The fusion unit can improve these and is able to merge the information on different levels, using object-level fusion as well as feature- and/or detection-level fusion. In order to perform object-level fusion from a sensor suite[2)], sufficient evidence in the form of correlated detections belonging to a particular real-world entity shall be provided by each sensor and sensor cluster. An advantage of feature-level fusion is the possible creation of an object in the common fusion unit out of detections, features or objects coming from a sensor suite, which would not have resulted in an object recognition within each sensor or each sensor cluster itself. Detection-level fusion may use machine learning algorithms to handle complexity. Future vehicles will integrate a sensor suite with many sensors and sensor clusters using different sensor technologies. Different technologies will complement each other to achieve a robust and reliable fusion.

Supportive sensor interfaces are provided so that the fusion is able to process the reliability of the used measurement method or data. These interfaces are sensor technology independent, specifically there are uniform interfaces for various sensor technologies. These supportive sensor interfaces are also used for the implementation of safety concepts.

The Figure 7 shows the interfaces that are in the scope of this document.

2) It consists of two or more different and independent sensors or sensor clusters.

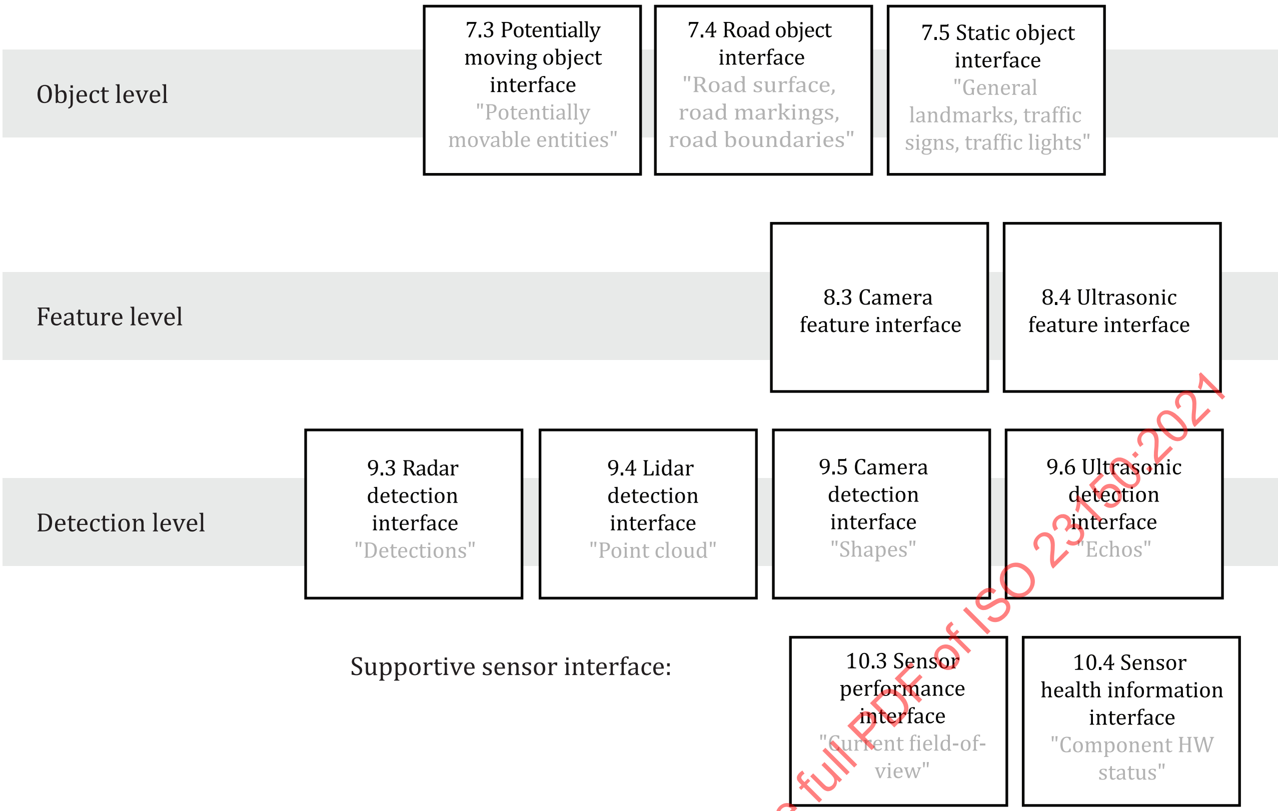


Figure 7 — Interfaces in the scope of this document

The interfaces covered in the scope of this document are the following:

— objects:

— potentially moving object interface (7.3): with specific attributes for, for example, lights, pedestrian;

— road object interface (7.4): road surface, road markings, road boundaries;

— static object interface (7.5): general landmarks, traffic signs, traffic lights;

— sensor or sensor cluster specific features:

— camera feature interface (8.3);

— ultrasonic feature interface (8.4);

— sensor or sensor cluster specific detections:

— radar detection interface (9.3);

— lidar detection interface (9.4);

— camera detection interface (9.5);

— ultrasonic detection interface (9.6);

— supportive information:

— sensor performance interface (10.3): with specific sensor-attributes for, for example, current field-of-view (FOV) segments, object recognition rates, reference target recognition rates;

— sensor health information interface (10.4): with specific attributes for, for example, generic sensor statuses, calibration information, sensor cluster definition.

A sensor as well as a sensor cluster can provide several logical interfaces. They can provide none, one or several interfaces on object level. Additionally, a sensor as well as a sensor cluster can provide a sensor technology specific interface on feature as well as detection level. Each sensor as well as each sensor cluster shall provide at least one interface either on object, feature or detection level to fulfil this document. Supportive sensor interfaces can be provided optionally. Sensors and sensor clusters may serve interfaces on all levels. Sensors mainly serve detection level interfaces (DLIs) whereas sensor clusters mainly serve feature level interfaces (FLIs) and/or object level interfaces (OLIs). Sensor clusters are based on sensors and therefore sensor clusters without a group of defined sensors should be avoided. If a sensor cluster offers no DLIs, the sensor cluster can be considered as a single sensor with multiple sensing elements. A sensor cluster provides consistent data over all its interfaces. The fusion unit does not have to use each provided interface. The provided interfaces can be used by other applications (for example, AD functions), not only by the fusion unit alone.

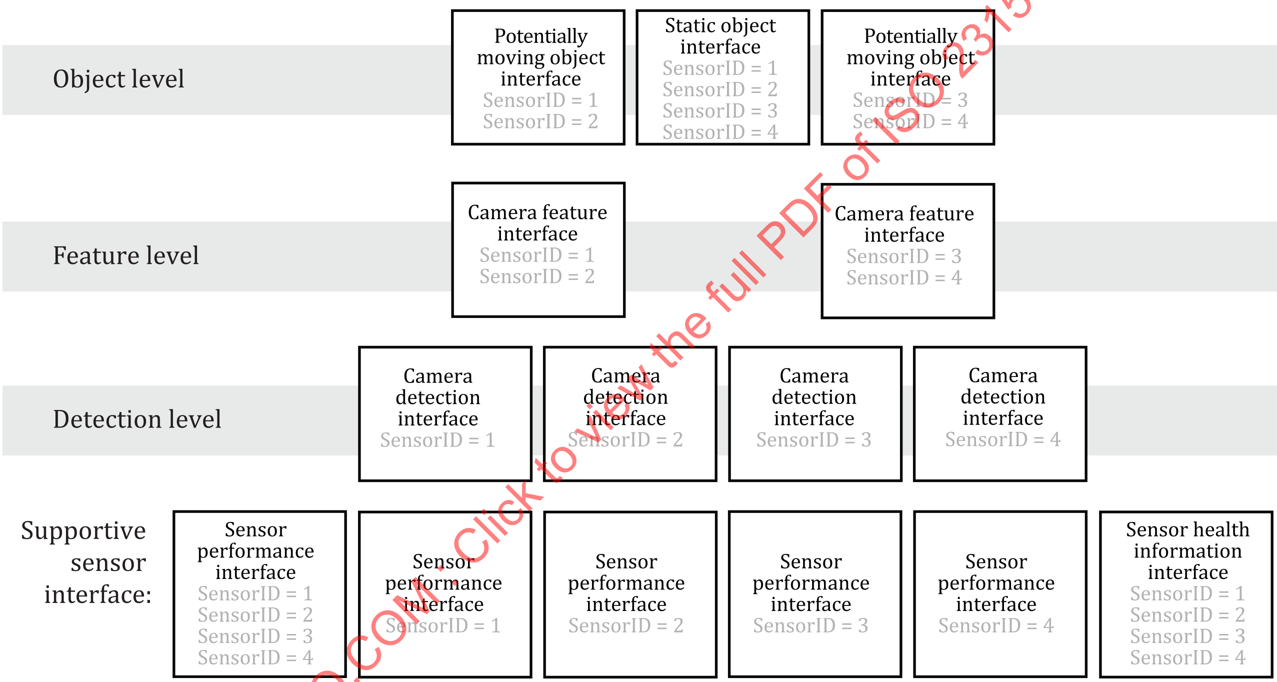


Figure 8 — Example of a camera sensor cluster and its sensor and sensor cluster interfaces

In Figure 8 sensor 1, sensor 2, sensor 3, sensor 4 and sensor sub-cluster {1, 2} and {3, 4} belong to the sensor cluster {1, 2, 3, 4}. The information of all sensors and sensor clusters belongs together and may be referenced between the interfaces. In this example, different combinations of sensors provide interfaces:

— the sensor cluster with the list of signal “Sensor ID” (A.1.3) values {1, 2, 3, 4} provides the static object interface (SOI), sensor performance interface (SPI) and sensor health information interface (SHII);

— the sensor sub-cluster with the list of signal “Sensor ID” (A.1.3) values {1, 2} provides potentially the moving object interface (PMOI), camera feature interface (CFI);

— the sensor sub-cluster with the list of signal “Sensor ID” (A.1.3) values {3, 4} provides PMOI, CFI;

— the sensors with signal “Sensor ID” (A.1.3) value 1, 2, 3 or 4 provide each camera detection interface (CDI) and SPI.

6.2 Generic interface header

Table 5 defines the structure of the subclause “Generic interface header” (6.2) as a specific signal grouping. This header is used by each interface and is adapted or extended accordingly. However, all the interface headers have at least these signals.

Table 5 — Specific signal grouping: Generic interface header

Signal	RL signal	Option
Interface version ID {major, minor, patch} (A.1.1)	Mandatory	Profile: Uniqueness of interface versioning (6.4)
Number of valid serving sensors (A.1.2)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
↪ Sensor ID (A.1.3)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
Interface ID (A.1.4)	Optional	Profile: Uniqueness of interface versioning (6.4)
Time stamp – <...> (A.1.5)	Mandatory	
<...> counter (A.1.6)	Optional	Redundancy (B.1.2) Signal: Time stamp – <...> (A.1.5)
Interface cycle time (A.1.7)	Optional	
Interface cycle time – variation (A.1.8)	Optional	
Data qualifier (A.1.9)	Mandatory	
Interface specific header extensions		
Recognised <...> – capability (A.1.10)	Optional	
Recognised <...> – status (A.1.11)	Optional	Redundancy (B.1.2) Signal: Recognised <...> – capability (A.1.10)
Number of valid <...> (A.1.12)	Mandatory	Optimise (B.1.1) List length optimisation

6.3 Generic interface entity

The interface entities define the set of signals without the header signals of the interface. Each logical interface provides at least one list of interface specific entities (except SHII). The number of valid entities of each interface entity type list is defined by the signal “Number of valid <...>” (A.1.12) (see Table 5). Interfaces with multiple entity types may not implement all interface entity type lists [see subclause “Multiple entity types for one interface” (B.2.3)]. The signals “Recognised <...> – capability” (A.1.10) and “Recognised <...> – status” (A.1.11) determine the minimal performance of the sensor for the interface entity type list, for example, at full load.

6.4 Profile: Uniqueness of interface versioning

The concatenation of the signal’s “Interface version ID {major, minor, patch}” (A.1.1) value, the signal’s “Sensor ID” (A.1.3) values and the signal’s “Interface ID” (A.1.4) value of an interface is a unique identifier for the interpretation of the interface message implementation. The signal “Number of valid serving sensors” (A.1.2) is also included. The order of the signal’s “Sensor ID” (A.1.3) values is also relevant. If a sensor could provide an interface more than once, this profile is adjusted in the corresponding interface to distinguish each interface implementation of one interface.

7 Object level

7.1 General

The object level interfaces (OLI) originate from several sensors. Therefore, at object level, the term sensor cluster is always used, even if a single sensor is serving the interface. The OLIs provide recognised real-world entities and recognition is based on historical data. Recognised objects are tracked and filtered over time. Model-based algorithms classify these objects and rate them with an existence probability value. The sensor cluster refers its recognised objects to one of two predefined vehicle coordinate systems [see signal “Vehicle coordinate system type” (A.1.21)]. The properties, and therefore the signals of the recognised objects, are generally independent of the sensor cluster’s sensor technology. Additional properties of the recognised objects can depend on the sensor technology and complement the object description of the interface. These sensor technology dependent signals are grouped in LSGs for each sensor technology.

To link recognised objects with non-OLI interface entities of the sensor cluster or its sensors, each recognised object has a unique signal’s “Object ID” (A.2.2) value that does not change over time and which is referenced by the entities. The signal’s “Object ID” (A.2.2) value can only be reused after the object has not been recognised for a defined amount of time and the sensor cluster is therefore certain that the object is no longer visible. Specifically, a reidentification of a previous recognised object with the same signal’s “Object ID” (A.2.2) value, after it has not been visible to the sensor cluster for a long time, is not guaranteed.

To link recognised objects of the sensor cluster’s OLIs, the set of related objects use the same consistent unique signal’s “Object grouping ID” (A.2.3) value to link all related objects. One dedicated value may indicate that the object is not linked to another object.

The object recognition algorithm is based on multiple measurements of the sensor cluster and is updated over time. For this reason, the entire object properties are not assigned to the time of one single measurement, but to an estimated point in time at which the properties of the interface’s objects fit together consistently. The object existence probability relates to this consistent set of estimated signals of the object at this estimated point in time.

On object level the following interfaces are available:

— potentially moving object interface (7.3);

— road object interface (7.4);

— static object interface (7.5).

7.2 Generic object level interface

Table 6 defines the generic structure of an OLI.

Table 6 — Generic object level interface

Structure	Multiplicity	Option
Generic object level header (7.2.1)	1	
Generic object level entity (7.2.2), specifically an individual object level entity list	Multiple	Size type: dynamic/fixed

Each OLI’s individual object level entity list can also be a set of different, specific object type lists [for example, for subclause “Road object interface” (7.4) there are two entity lists for the object types: road marking and road boundary and a single entity for the object type: road surface].

7.2.1 Generic object level header

Table 7 defines the interface header in subclause "Generic object level header" (7.2.1) and the changes due to the adaptation in comparison to the generic interface header in subclause "Generic interface header" (6.2).

Table 7 — Specific signal grouping: Generic object level header

Signal	RL signal	Option
Interface version ID {major, minor, patch} (A.1.1)	Mandatory	Profile: Uniqueness of interface versioning (6.4)
Number of valid serving sensors (A.1.2)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
↪ Sensor ID (A.1.3)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
Interface ID (A.1.4)	Optional	Profile: Uniqueness of interface versioning (6.4)
Time stamp – predicted (A.1.5.1)	Mandatory	
Cycle counter (A.1.6.1)	Optional	Redundancy (B.1.2) Signal: Time stamp – predicted (A.1.5.1)
Interface cycle time (A.1.7)	Optional	
Interface cycle time – variation (A.1.8)	Optional	
Data qualifier (A.1.9)	Mandatory	
OLI specific header extensions		
Recognised <...> – capability (A.1.10)	Optional	
Recognised <...> – status (A.1.11)	Optional	Redundancy (B.1.2) Signal: Recognised <...> – capability (A.1.10)
Number of valid <...> (A.1.12)	Mandatory	Optimise (B.1.1) List length optimisation

The following LSGs, which normally shall be provided by the sensor cluster in another interface, may be added to this header, if, for example, the other interface is not implemented by the sensor cluster [see subclause "Need of logical signal group" (B.3.3)]:

— Interface: SPI; LSG "Information: vehicle coordinate system" (see Table 46);

— Interface: SHII; LSG "Calibration" (see Table 48);

— Interface: SHII; LSG "Sensor cluster" (see Table 48).

7.2.2 Generic object level entity

Table 8 defines the generic signal grouping "Generic object level entity status" for each OLI object type, except of road surface.

Table 8 — Specific signal grouping: Generic object level entity status

Signal	RL signal	Option
Existence probability – object level (A.2.1)	Mandatory	
Object ID (A.2.2)	Mandatory	Alternative (B.3.2)
Object grouping ID (A.2.3)	Optional	

Table 8 *(continued)*

Signal	RL signal	Option
Age (A.2.4)	Mandatory	
Number of valid observations – object level (A.2.5)	Optional	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – object level (A.2.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
↪ Time stamp reference – object level (A.2.6)	Conditional (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Alternative (B.1.3)
↪ Observation status – object level (A.2.7)	Conditional (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Redundancy (B.1.2) Signal: Object ID (A.2.2), Time stamp – predicted (A.1.5.1)
Track quality (A.2.8)	Optional	
Measurement status – object level (A.2.9)	Mandatory	

Each interface type adds individual properties to the interface entity types. Geometric information of the object references the vehicle coordinate system [see signal "Vehicle coordinate system type" (A.1.21)]. Detections and features of the sensor cluster and its sensors shall reference objects of the OLIs with the signal "Object ID" (A.2.2). Potentially connected objects of a sensor cluster shall be linked together and the sensor cluster shall use the same signal's "Object grouping ID" (A.2.3) value for all objects of this connected object group.

7.3 Potentially moving object interface

At object level, the potentially moving object interface (PMOI) contains the list of objects that could potentially move and are relevant for driving situations. This includes all objects that currently move, have moved (see Figure 9) and that are not definitely static three-dimensional structures. Table 9 defines the logical structure of the potentially moving object interface. The object interface defines sets of signals for more detailed potentially moving object types (for example, lights, person poses). Their existence is indicated by the signals "Number of valid lights" (A.2.32) and "Number of valid person's poses" (A.2.35) of at most one person.

Table 9 — Potentially moving object interface

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
Information: interface	M	Interface version ID {major, minor, patch} (A.1.1)	M	Profile: Uniqueness of interface versioning (6.4)
		Number of valid serving sensors (A.1.2)	M	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
		Size type: dynamic/fixed Size #: Number of valid serving sensors (A.1.2)		
		Sensor ID (A.1.3)	M	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
		Interface ID (A.1.4)	O	Profile: Uniqueness of interface versioning (6.4)
		Time stamp – predicted (A.1.5.1)	M	
		Cycle counter (A.1.6.1)	O	Redundancy (B.1.2) Signal: Time stamp – predicted (A.1.5.1)
		Interface cycle time (A.1.7)	O	
		Interface cycle time – variation (A.1.8)	O	
		Data qualifier (A.1.9)	M	
		Tracking motion model (A.1.13)	O	
		Motion type (A.1.14)	M	Profile: Motion (7.3.3)
Potentially moving objects	M	Recognised potentially moving objects – capability (A.1.10.1)	O	
		Recognised potentially moving objects – status (A.1.11.1)	O	Redundancy (B.1.2) Signal: Recognised potentially moving objects – capability (A.1.10.1)
		Number of valid potentially moving objects (A.1.12.1)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid potentially moving objects (A.1.12.1)		

Table 9 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ Potentially moving object status	M	Existence probability – object level (A.2.1)	M	
		Object ID (A.2.2)	M	Alternative (B.3.2)
		Object grouping ID (A.2.3)	O	
		Age (A.2.4)	M	
		Number of valid observations – object level (A.2.5)	O	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – object level (A.2.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
		Size type: dynamic/fixed Size #: Number of valid observations – object level (A.2.5)		
		Time stamp reference – object level (A.2.6)	C (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Alternative (B.1.3)
		Observation status – object level (A.2.7)	C (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Redundancy (B.1.2) Signal: Object ID (A.2.2), Time stamp – predicted (A.1.5.1)
		Track quality (A.2.8)	O	
		Measurement status – object level (A.2.9)	M	

Table 9 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ Potentially moving object information	M	Number of valid potentially moving object classifications (A.2.10)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Potentially moving object classification type (A.2.11)
		Size type: dynamic/fixed Size #: Number of valid potentially moving object classifications (A.2.10)		
		Potentially moving object classification type (A.2.11)	M	Alternative (B.1.3)
		Potentially moving object classification type – confidence (A.2.12)	M	
↪ Potentially moving object position	M	Position – object level {x, y, z} (A.2.13)	{M, M, O}	
		Position – object level {x, y, z} – error (A.2.14)	{M, M, O}	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Orientation {yaw, pitch, roll} (A.2.15)	{C (B.2.1) Relevant: camera, O, O}	
		Orientation {yaw, pitch, roll} – error (A.2.16)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Reference point (A.2.17)	O	
		Road level (A.2.18)	O	
↪ Potentially moving object bounding box	C (B.2.1) Relevant: camera	Bounding box extent {length, width, height} (A.2.19)	{M, M, O}	
		Bounding box extent {length, width, height} – error (A.2.20)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Bounding box ground clearance (A.2.21)	O	
		Included geometric structures (A.2.22)	O	

Table 9 *(continued)*

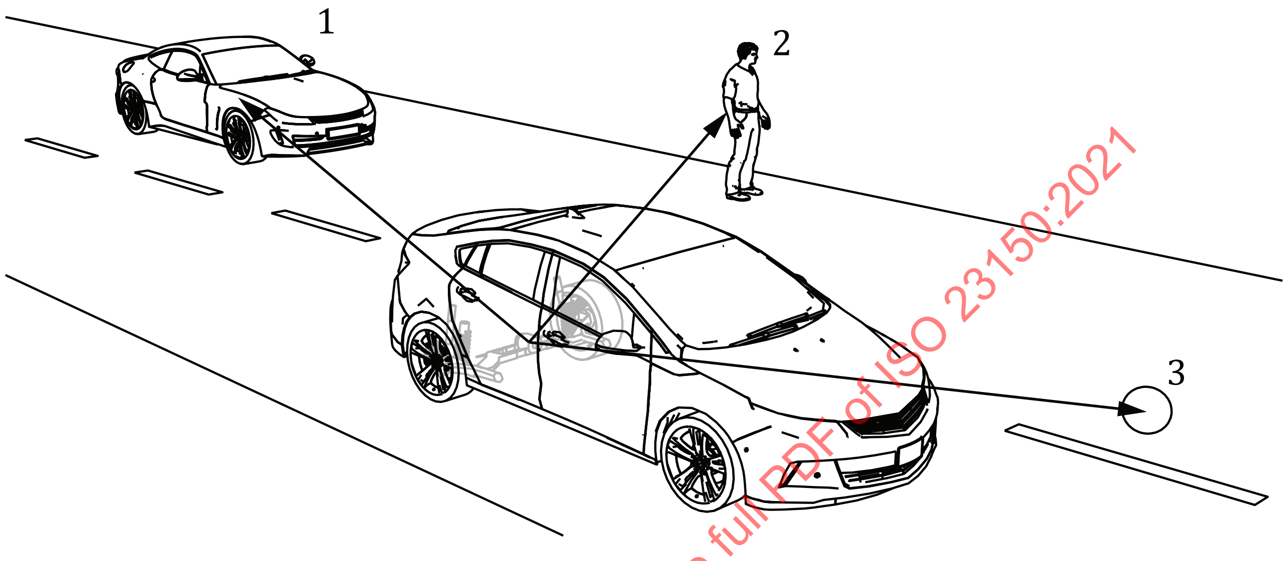
LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ Potentially moving object dynamics	M	Velocity {x, y, z} – object level (A.2.23)	{C, C, C (B.2.2) Signal: Position {z} (A.2.13)} (B.2.1) Relevant: except ultrasonic	Profile: Motion (7.3.3)
		Velocity {x, y, z} – object level – error (A.2.24)	O	Profile: Motion (7.3.3) Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Acceleration {x, y, z} (A.2.25)	O	Profile: Motion (7.3.3)
		Acceleration {x, y, z} – error (A.2.26)	O	Profile: Motion (7.3.3) Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Instantaneous centre of rotation {x, y} (A.2.27)	C (B.2.2) Signal: Rotation rate at instantaneous centre of rotation {yaw} (A.2.29)	
		Instantaneous centre of rotation {x, y} – error (A.2.28)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Rotation rate at instantaneous centre of rotation {yaw} (A.2.29)	O	Profile: Motion (7.3.3)
		Rotation rate at instantaneous centre of rotation {yaw} – error (A.2.30)	O	Profile: Motion (7.3.3) Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Movement status (A.2.31)	O	

Table 9 *(continued)*

LSG	RL LSG M/C/O		Signal	RL signal M/C/O	Option
↪ Potentially moving object's lights	C (B.2.1) Relevant: camera		Number of valid lights (A.2.32)	M	Optimise (B.1.1) List length optimisation) Alternative Unrolling tuple-lists (B.1.6) Key: Light type (A.2.33)
			Size type: dynamic/fixed Size #: Number of valid lights (A.2.32)		
			Light type (A.2.33)	M	Alternative (B.1.3)
			Light status (A.2.34)	M	
↪ Potentially moving object: person	O		Number of valid person's poses (A.2.35)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Person pose type (A.2.36)
			Size type: dynamic/fixed Size #: Number of valid person's poses (A.2.35)		
			Person pose type (A.2.36)	M	Alternative (B.1.3)
			Person pose {yaw, pitch, roll} (A.2.37)	M	
			Person pose {yaw, pitch, roll} – error (A.2.38)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
↪ Lane related information	O		Object lane association (A.2.39)	M	
			Angle between object edge and lane {left edge right lane, right edge left lane} (A.2.40)	O	
			Angle between object edge and lane {left edge right lane, right edge left lane} – error (A.2.41)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
			Percentage side lane {left, right} (A.2.42)	O	
↪ Motion related information	O		Angular position {azimuth} (A.2.43)	M	
			Angular velocity {azimuth} (A.2.44)	M	
↪ Camera sensor technology specific	C (B.2.1) Relevant: camera		Scale change – object level (A.2.45)	M	
↪ Radar sensor technology specific	C (B.2.1) Relevant: radar		Entity radar cross section (A.2.46)	M	

Table 9 *(continued)*

LSG	RL LSG M/C/O		Signal	RL signal M/C/O	Option
↪ Lidar sensor technology specific	C (B.2.1) Relevant: lidar		Entity lidar reflectivity (A.2.47)	M	



Key

1 vehicle

2 pedestrian

3 unknown potentially moving object

Figure 9 — Example for potentially moving objects

7.3.1 Potentially moving object header

Table 10 defines the interface header in subclause "Potentially moving object header" (7.3.1) and the changes due to the adaptation in comparison to the generic interface header in subclause "Generic object level header" (7.2.1). The header of the PMOI can contain a list of valid potentially moving object entities [see subclause "Potentially moving object entity" (7.3.2)].

Table 10 — Specific signal grouping: Potentially moving object header

Signal	RL signal	Option
Interface version ID {major, minor, patch} (A.1.1)	Mandatory	Profile: Uniqueness of interface versioning (6.4)
Number of valid serving sensors (A.1.2)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
↪ Sensor ID (A.1.3)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
Interface ID (A.1.4)	Optional	Profile: Uniqueness of interface versioning (6.4)
Time stamp – predicted (A.1.5.1)	Mandatory	

Table 10 *(continued)*

Signal	RL signal	Option
Cycle counter (A.1.6.1)	Optional	Redundancy (B.1.2) Signal: Time stamp – predicted (A.1.5.1)
Interface cycle time (A.1.7)	Optional	
Interface cycle time – variation (A.1.8)	Optional	
Data qualifier (A.1.9)	Mandatory	
Tracking motion model (A.1.13)	Optional	
Motion type (A.1.14)	Mandatory	Profile: Motion (7.3.3)
Recognised potentially moving objects – capability (A.1.10.1)	Optional	
Recognised potentially moving objects – status (A.1.11.1)	Optional	Redundancy (B.1.2) Signal: Recognised potentially moving objects – capability (A.1.10.1)
Number of valid potentially moving objects (A.1.12.1)	Mandatory	Optimise (B.1.1) List length optimisation

The following LSGs, which normally shall be provided by the sensor cluster in another interface, may be added to this header, if, for example, the other interface is not implemented by the sensor cluster [see subclause “Need of logical signal group” (B.3.3)]:

— interface: SPI; LSG “Information: vehicle coordinate system” (see Table 46);

— interface: SHII; LSG “Calibration” (see Table 48);

— interface: SHII; LSG “Sensor cluster” (see Table 48).

7.3.2 Potentially moving object entity

Each potentially moving object describes a recognised real-world object and consists of several LSGs. Each object, which may move, even if it might have not moved yet since first recognition of the real-world object, is an entity of potentially moving objects.

A potentially moving object is described by the following LSGs for potentially moving objects with attributes for different parts (for example, vehicle, pedestrian).

— **Status:** the status describes general information of the tracked potentially moving object. This information is based on history of the object tracking. Table 11 defines the signal grouping “Potentially moving object entity status”. It defines this status LSG of the potentially moving object and redefines the signal grouping “Generic object level entity status” (see Table 8).

— **Information:** the information describes properties such as the confidence of an object type.

— **Position:** the position describes the geometric position of the object relative to the vehicle coordinate system [see signal “Vehicle coordinate system type” (A.1.21)].

— **Bounding box:** the object bounding box describes the 3D rectangular hull that encloses the recognised object.

— **Dynamics:** the dynamics describes the motion information of the tracked potentially moving object with respect to the motion type defined by the signal “Motion type” (A.1.14) [see Profile: Motion (7.3.3)].

— **Lights:** the lights describe the status of the tracked potentially moving object’s lights.

— **Person:** the person describes poses of a tracked pedestrian if the tracked potentially moving object is a pedestrian.

— **Lane related information:** the lane related information describes the tracked potentially moving object in relation with real-world lanes.

— **Motion related information:** the motion related information describes additional motion information of the tracked potentially moving object.

— **Camera sensor technology specific:** the camera sensor technology specific data describe dynamic information in the camera projection plane.

— **Radar sensor technology specific:** the radar sensor technology specific data describe different surface and material properties of the potentially moving object with respect to the radar.

— **Lidar sensor technology specific:** the lidar sensor technology specific data describes different surface and material properties of the potentially moving object with respect to the lidar (for example, reflection strength of a surface).

Table 11 — Specific signal grouping: Potentially moving object entity status

Signal	RL signal	Option
Existence probability – object level (A.2.1)	Mandatory	
Object ID (A.2.2)	Mandatory	Alternative (B.3.2)
Object grouping ID (A.2.3)	Optional	
Age (A.2.4)	Mandatory	
Number of valid observations – object level (A.2.5)	Optional	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – object level (A.2.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
↪ Time stamp reference – object level (A.2.6)	Conditional (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Alternative (B.1.3)
↪ Observation status – object level (A.2.7)	Conditional (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Redundancy (B.1.2) Signal: Object ID (A.2.2), Time stamp – predicted (A.1.5.1)
Track quality (A.2.8)	Optional	
Measurement status – object level (A.2.9)	Mandatory	

Potentially moving objects may be linked together (for example, a truck with a trailer or a construction site vehicle with a traffic sign at the rear of the vehicle) by using the same signal's "Object grouping ID" (A.2.3) value.

7.3.3 Profile: Motion

The signal "Motion type" (A.1.14) defines the interpretation of the signal "Velocity {x, y, z} – object level (A.2.23)", "Velocity {x, y, z} – object level – error" (A.2.24), "Acceleration {x, y, z}" (A.2.25), "Acceleration {x, y, z} – error" (A.2.26), "Rotation rate at instantaneous centre of rotation {yaw}" (A.2.29) and "Rotation rate at instantaneous centre of rotation {yaw} – error" (A.2.30) values.

7.4 Road object interface

The road object interface (RDOI) consists of a general road surface information and two types of objects: road markings and road boundaries (see Figure 10). Table 12 defines the logical structure of the road object interface.

Table 12 — Road object interface

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
Information: interface	M	Interface version ID {major, minor, patch} (A.1.1)	M	Profile: Uniqueness of interface versioning (6.4)
		Number of valid serving sensors (A.1.2)	M	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
		Size type: dynamic/fixed Size #: Number of valid serving sensors (A.1.2)		
		Sensor ID (A.1.3)	M	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
		Interface ID (A.1.4)	O	Profile: Uniqueness of interface versioning (6.4)
		Time stamp – predicted (A.1.5.1)	M	
		Cycle counter (A.1.6.1)	O	Redundancy (B.1.2) Signal: Time stamp – predicted (A.1.5.1)
		Interface cycle time (A.1.7)	O	
		Interface cycle time – variation (A.1.8)	O	
		Data qualifier (A.1.9)	M	
		Colour model type (A.1.15)	C (B.2.1) Relevant: camera	Profile: Colour model for RDOI (7.4.3) Optimise (B.1.3) Use enumeration to define colours by defining colour values and the applied colour model for each enumerator.

Table 12 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
Road surface	C (B.2.3) multiple entity types	Road type (A.2.48)	M	
		Number of valid road surface classifications (A.2.49)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Road surface classification type (A.2.50)
		Size type: dynamic/fixed Size #: Number of valid road surface classifications (A.2.49)		
		Road surface classification type (A.2.50)	M	Alternative (B.1.3)
		Road surface classification type – confidence (A.2.51)	M	
		Road surface roughness (A.2.52)	O	
		Number of valid road surface condition classifications (A.2.53)	O	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Road surface condition classification type (A.2.54)
		Size type: dynamic/fixed Size #: Number of valid road surface condition classifications (A.2.53)		
		Road surface condition classification type (A.2.54)	C (B.2.2) Signal: Number of valid road surface condition classifications (A.2.53)	Alternative (B.1.3)
		Road surface condition classification type – confidence (A.2.55)	C (B.2.2) Signal: Number of valid road surface condition classifications (A.2.53)	
		Track quality (A.2.8)	O	
		Measurement status – object level (A.2.9)	O	

Table 12 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
Road markings	C (B.2.3) multiple entity types	Recognised road markings – capability (A.1.10.2)	O	
		Recognised road markings – status (A.1.11.2)	O	Redundancy (B.1.2) Signal: Recognised road markings – capability (A.1.10.2)
		Number of valid road markings (A.1.12.2)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid road markings (A.1.12.2)		
↪ Road marking status	M	Existence probability – object level (A.2.1)	M	
		Object ID (A.2.2)	M	Alternative (B.3.2)
		Object grouping ID (A.2.3)	O	
		Age (A.2.4)	M	
		Number of valid observations – object level (A.2.5)	O	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – object level (A.2.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
		Size type: dynamic/fixed Size #: Number of valid observations – object level (A.2.5)		
		Time stamp reference – object level (A.2.6)	C (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Alternative (B.1.3)
		Observation status – object level (A.2.7)	C (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Redundancy (B.1.2) Signal: Object ID (A.2.2), Time stamp – predicted (A.1.5.1)
		Track quality (A.2.8)	O	
		Measurement status – object level (A.2.9)	M	
↪ Road marking information	M	Number of valid road marking classifications (A.2.56)	M	Optimise (B.1.1) List length optimisation

Table 12 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
		Size type: dynamic/fixed Size #: Number of valid road marking classifications (A.2.56)		
		Road marking type (A.2.57)	M	
		Road marking type – confidence (A.2.58)	M	
		Road object lane association (A.2.59)	O	
		Road object lane association – confidence (A.2.60)	O	
		Arrow orientation (A.2.61)	O	
		Arrow direction (A.2.62)	C (B.2.2) * The signal “Road marking type” (A.2.57) has an enumerator “RoadMarkingType_Arrow” or a similar enumerator defined during the system desing phase.	
		Number of valid sign classifications (A.2.63)	O	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sign classification type (A.2.64)
		Size type: dynamic/fixed Size #: Number of valid sign classifications (A.2.63)		
		Sign classification type (A.2.64)	C (B.2.2) Signal: Number of valid sign classifications (A.2.63)	Alternative (B.1.3)
		Sign classification type – confidence (A.2.65)	C (B.2.2) Signal: Number of valid sign classifications (A.2.63)	
		Sign value (A.2.66)	O	
		Sign value unit (A.2.67)	O	
		Sign state (A.2.68)	O	

Table 12 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ ↪ Road marking colour tone	C (B.2.1) Relevant: camera	Size type: dynamic/fixed Size # * Implicit list length – Optimise (B.1.1): List length optimisation: depends on Colour model type (A.1.15)		
		Colour value – object level (A.2.69)	M	Profile: Colour model for RDOI (7.4.3)
		Colour tone – confidence – object level (A.2.70)	O	Profile: Colour model for RDOI (7.4.3)
↪ Road marking polynomials[a]	C (B.2.1) Relevant: camera, lidar	Number of valid connections (A.2.71)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Connection type (A.2.72)
		Size type: dynamic/fixed Size #: Number of valid connections (A.2.71)		
		Connection type (A.2.72)	M	Alternative (B.1.3)
		Connection grouping ID (A.2.73)	M	
		Number of valid polynomials (A.2.74)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid polynomials (A.2.74)		
		Polynomial coefficient y $\{c_0, c_1, c_2, c_3\}$ (A.2.75)	M	
		Polynomial coefficient z $\{c_0, c_1, c_2, c_3\}$ (A.2.76)	O	
		Polynomial y – error (A.2.77)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Polynomial z – error (A.2.78)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Polynomial range x {begin, end} (A.2.79)	M	
		Width – polynomial (A.2.80)	O	

Table 12 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
		Width – polynomial – error (A.2.81)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Width – polynomial – confidence (A.2.82)	O	
		Height – polynomial (A.2.83)	O	
		Height – polynomial – error (A.2.84)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Height – polynomial – confidence (A.2.85)	O	
		Number of valid data ranges (A.2.86)	O	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid data ranges (A.2.86)		
		Supported data range x {begin, end} (A.2.87)	C (B.2.2) Signal: Number of valid data ranges (A.2.86)	
		Supported axis (A.2.88)	C (B.2.2) Signal: Number of valid data ranges (A.2.86)	
↪ Road marking polylines[a]	C (B.2.1) Relevant: camera, lidar	Number of valid connections (A.2.71)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Connection type (A.2.72)
		Size type: dynamic/fixed Size #: Number of valid connections (A.2.71)		
		Connection type (A.2.72)	M	Alternative (B.1.3)
		Connection grouping ID (A.2.73)	M	
		Polyline interpolation method (A.2.89)	M	
		Number of valid polylines (A.2.90)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid polylines (A.2.90)		

Table 12 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
		Number of valid vertices (A.2.91)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid vertices (A.2.91)		
		Vertex point {x, y, z} (A.2.92)	{M, M, O}	
		Vertex point {x, y, z} – error (A.2.93)	{M, M, O}	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Vertex point – confidence {x, y, z} (A.2.94)	O	
		Width – vertex (A.2.95)	O	
		Width – vertex – error (A.2.96)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Width – vertex – confidence (A.2.97)	O	
		Height – vertex (A.2.98)	O	
		Height – vertex – error (A.2.99)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Height – vertex – confidence (A.2.100)	O	
Road boundaries	C (B.2.3) multiple entity types	Recognised road boundaries – capability (A.1.10.3)	O	
		Recognised road boundaries – status (A.1.11.3)	O	Redundancy (B.1.2) Signal: Recognised road boundaries – capability (A.1.10.3)
		Number of valid road boundaries (A.1.12.3)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid road boundaries (A.1.12.3)		
↪ Road boundary status	M	Existence probability – object level (A.2.1)	M	
		Object ID (A.2.2)	M	Alternative (B.3.2)
		Object grouping ID (A.2.3)	O	
		Age (A.2.4)	M	

Table 12 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
		Number of valid observations – object level (A.2.5)	O	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – object level (A.2.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
		Size type: dynamic/fixed Size #: Number of valid observations – object level (A.2.5)		
		Time stamp reference – object level (A.2.6)	C (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Alternative (B.1.3)
		Observation status – object level (A.2.7)	C (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Redundancy (B.1.2) Signal: Object ID (A.2.2), Time stamp – predicted (A.1.5.1)
		Track quality (A.2.8)	O	
		Measurement status – object level (A.2.9)	M	
↪ Road boundary information	M	Number of valid road boundary classifications (A.2.101)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Road boundary type (A.2.102)
		Size type: dynamic/fixed Size #: Number of valid road boundary classifications (A.2.101)		
		Road boundary type (A.2.102)	M	Alternative (B.1.3)
		Road boundary type – confidence (A.2.103)	M	
		Road object lane association (A.2.59)	O	
		Road object lane association – confidence (A.2.60)	O	

Table 12 *(continued)*

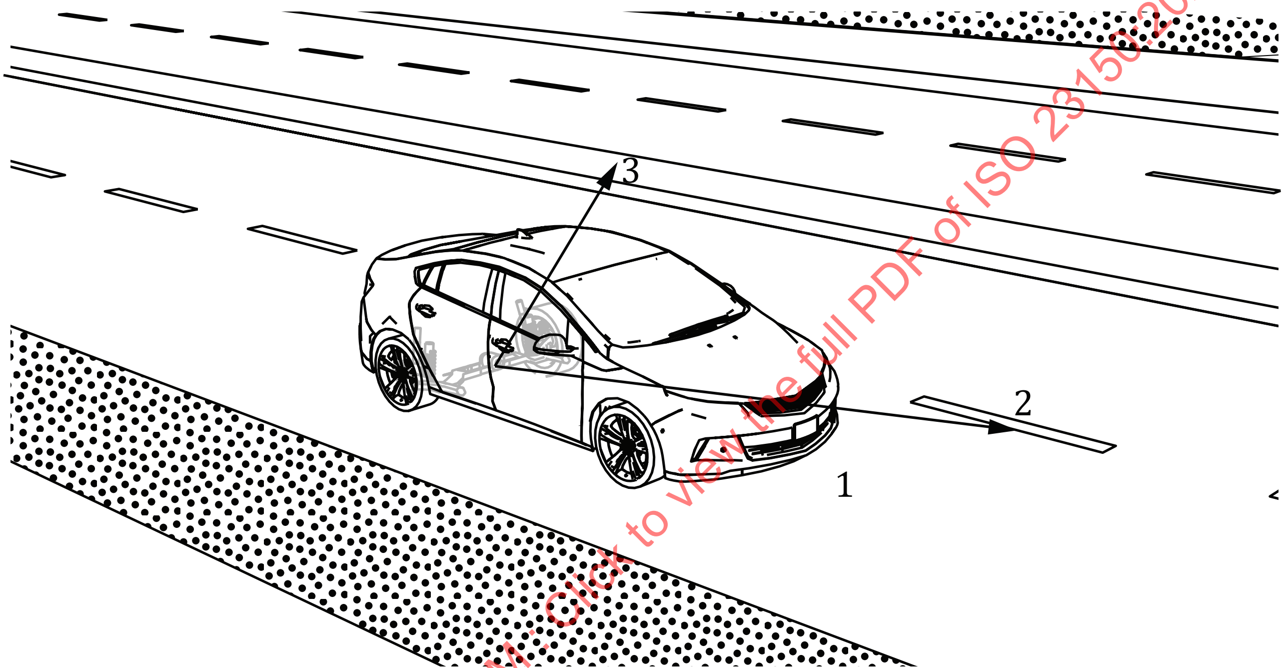
LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ ↪ Road boundary colour tone	C (B.2.1) Relevant: camera	Size type: dynamic/fixed Size # * Implicit list length – Optimise (B.1.1): List length optimisation: depends on Colour model type (A.1.15)		
		Colour value – object level (A.2.69)	M	Profile: Colour model for RDOI (7.4.3)
		Colour tone – confidence – object level (A.2.70)	O	Profile: Colour model for RDOI (7.4.3)
↪ Road boundary polynomials	C (B.2.1) Relevant: camera, lidar, radar	Number of valid polynomials (A.2.74)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid polynomials (A.2.74)		
		Polynomial coefficient y {c_0, c_1, c_2, c_3} (A.2.75)	M	
		Polynomial coefficient z {c_0, c_1, c_2, c_3} (A.2.76)	O	
		Polynomial y – error (A.2.77)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Polynomial z – error (A.2.78)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Polynomial range x {begin, end} (A.2.79)	M	
		Width – polynomial (A.2.80)	O	
		Width – polynomial – error (A.2.81)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Width – polynomial – confidence (A.2.82)	O	
		Height – polynomial (A.2.83)	O	
		Height – polynomial – error (A.2.84)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Height – polynomial – confidence (A.2.85)	O	

Table 12 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
		Number of valid data ranges (A.2.86)	O	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid data ranges (A.2.86)		
		Supported data range x {begin, end} (A.2.87)	C (B.2.2) Signal: Number of valid data ranges (A.2.86)	
		Supported axis (A.2.88)	C (B.2.2) Signal: Number of valid data ranges (A.2.86)	
↪ Road boundary polylines	C (B.2.1) Relevant: camera, lidar, radar	Polyline interpolation method (A.2.89)	M	
		Number of valid polylines (A.2.90)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid polylines (A.2.90)		
		Number of valid vertices (A.2.91)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid vertices (A.2.91)		
		Vertex point {x, y, z} (A.2.92)	{M, M, O}	
		Vertex point {x, y, z} – error (A.2.93)	{M, M, O}	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Vertex point – confidence {x, y, z} (A.2.94)	O	
		Width – vertex (A.2.95)	O	
		Width – vertex – error (A.2.96)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Width – vertex – confidence (A.2.97)	O	
		Height – vertex (A.2.98)	O	

Table 12 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
		Height – vertex – error (A.2.99)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Height – vertex – confidence (A.2.100)	O	
a Only one LSG (road marking polynomials or road marking polylines) should be implemented in an interface.				



Key

1 road surface; relevant for the ego-vehicle

2 road marking

3 road boundary

Figure 10 — Example for road objects

7.4.1 Road object header

Table 13 defines the interface header in subclause "Road object header" (7.4.1) and the changes due to the adaptation in comparison to the generic interface header in subclause "Generic object level header" (7.2.1). The header of the RDOI can contain a list of valid road marking entities, a list of valid road boundary entities and/or a single road surface entity [see subclause "Road object entity" (7.4.2)].

Table 13 — Specific signal grouping: Road object header

Signal	RL signal	Option
Interface version ID {major, minor, patch} (A.1.1)	Mandatory	Profile: Uniqueness of interface versioning (6.4)

Table 13 *(continued)*

Signal	RL signal		Option
Number of valid serving sensors (A.1.2)	Mandatory		Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
↪ Sensor ID (A.1.3)	Mandatory		Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
Interface ID (A.1.4)	Optional		Profile: Uniqueness of interface versioning (6.4)
Time stamp – predicted (A.1.5.1)	Mandatory		
Cycle counter (A.1.6.1)	Optional		Redundancy (B.1.2) Signal: Time stamp – predicted (A.1.5.1)
Interface cycle time (A.1.7)	Optional		
Interface cycle time – variation (A.1.8)	Optional		
Data qualifier (A.1.9)	Mandatory		
Colour model type (A.1.15)	Conditional (B.2.1) Relevant: camera		Profile: Colour model for RDOI (7.4.3) Optimise (B.1.3) Use enumeration to define colours by defining colour values and the applied colour model for each enumerator.
Recognised road markings – capability (A.1.10.2)	RL LSG conditional (B.2.3) multiple entity types	Optional	
Recognised road markings – status (A.1.11.2)		Optional	Redundancy (B.1.2) Signal: Recognised road markings – capability (A.1.10.2)
Number of valid road markings (A.1.12.2)		Mandatory	Optimise (B.1.1) List length optimisation
Recognised road boundaries – capability (A.1.10.3)		Optional	
Recognised road boundaries – status (A.1.11.3)		Optional	Redundancy (B.1.2) Signal: Recognised road boundaries – capability (A.1.10.3)
Number of valid road boundaries (A.1.12.3)		Mandatory	Optimise (B.1.1) List length optimisation

The following LSGs, which normally shall be provided by the sensor cluster in another interface, may be added to this header, if, for example, the other interface is not implemented by the sensor cluster [see subclause "Need of logical signal group" (B.3.3)]:

— interface: SPI; LSG "Information: vehicle coordinate system" (see Table 46);

— interface: SHII; LSG "Calibration" (see Table 48);

— interface: SHII; LSG "Sensor cluster" (see Table 48).

7.4.2 Road object entity

Each road object describes a recognised real-world road limitation (road marking or road boundary) or the relevant road surface for the ego-vehicle. They consist of several LSGs.

The road surface is described by the road surface LSG. The road surface provides no LSG “Status”.

A road marking is described by the following LSGs for road marking.

- **Status:** the status describes the general information of a tracked road marking. This information is based on history of the object tracking. Table 14 defines the signal grouping “Road object entity status”. It defines this status LSG of the road marking and redefines the signal grouping “Generic object level entity status” (see Table 8).
- **Information:** the information describes the road marking type and different visual properties related to the road marking.
- **Colour tone:** the colour tone describes the main colour of the road marking.
- The contour could be described by polynomials and/or polylines:
  - **Polynomials:** the polynomials of a road marking describe the detailed shape and bounds in a 3D vector space with polynomial lines. Different polynomials describe the y and z values of the contour independently.
  - **Polylines:** the polylines of a road marking describe the detailed shape and bounds in a 3D vector space with multiple line segments and a defined interpolation method.

A road boundary is described by the following LSGs for road boundary:

- **Status:** the status describes the general information of a tracked road boundary. This information is based on history of the object tracking. Table 14 defines the signal grouping “Road object entity status”. It defines this status LSG of the road boundary and redefines the signal grouping “Generic object level entity status” (see Table 8).
- **Information:** the information describes the road boundary type and different visual properties related to the road boundary.
- **Colour tone:** the colour tone describes the main colour of the road boundary.
- The contour could be described by polynomials and/or polylines:
  - **Polynomials:** the polynomials of a road boundary describe the detailed shape and bounds in a 3D vector space with polynomial lines. Different polynomials describe the y and z values of the contour independently.
  - **Polylines:** the polylines of a road boundary describe the detailed shape and bounds in a 3D vector space with multiple line segments and a defined interpolation method.

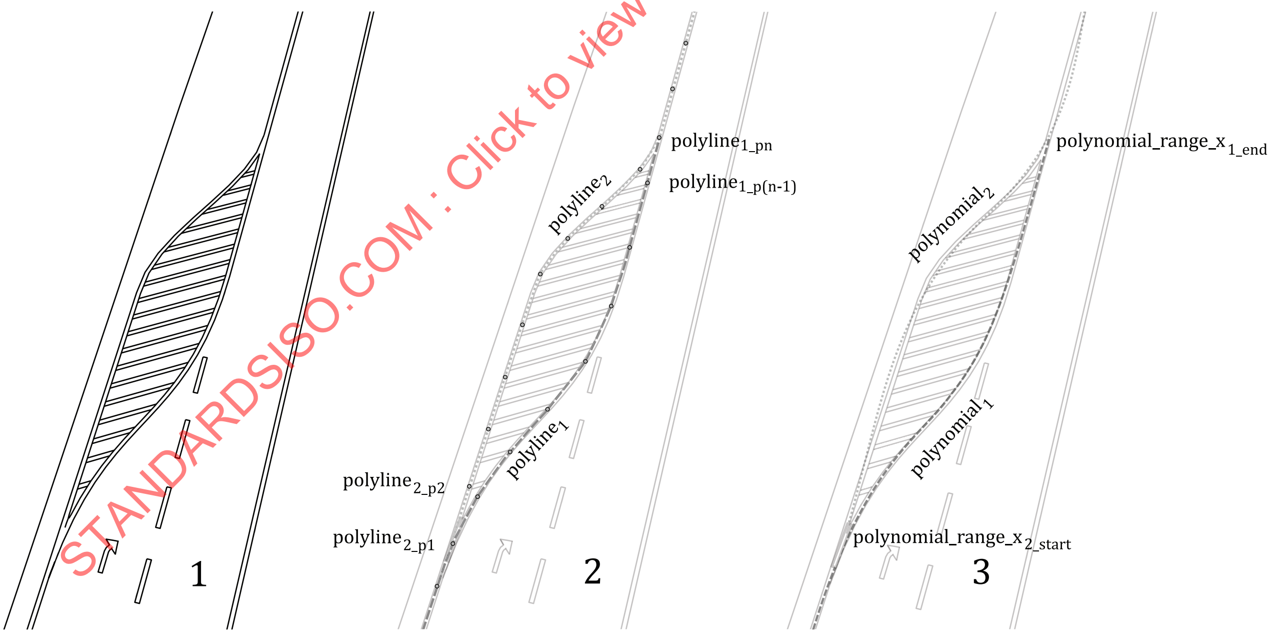
Table 14 — Specific signal grouping: Road object entity status

Signal	RL signal	Option
Existence probability – object level (A.2.1)	Mandatory	
Object ID (A.2.2)	Mandatory	Alternative (B.3.2)
Object grouping ID (A.2.3)	Optional	
Age (A.2.4)	Mandatory	

Table 14 *(continued)*

Signal	RL signal	Option
Number of valid observations – object level (A.2.5)	Optional	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – object level (A.2.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
↪ Time stamp reference – object level (A.2.6)	Conditional (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Alternative (B.1.3)
↪ Observation status – object level (A.2.7)	Conditional (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Redundancy (B.1.2) Signal: Object ID (A.2.2), Time stamp – predicted (A.1.5.1)
Track quality (A.2.8)	Optional	
Measurement status – object level (A.2.9)	Mandatory	

Road markings and road boundaries can be represented either by polynomials of degree 3 or by polylines consisting of one or more segments merged at the vertex points. See Figure 11 and see informative figures in Annex A for details: Figure A.6 and Figure A.9.



Key

1 real-world scene

2 polyline contour representation

3 polynomial contour representation

Figure 11 — Example for alternative contour representations

The road marking contours are specified with road marking polynomials and/or road marking polylines and refer to connections of road markings by the signal "Connection type" (A.2.72) and the signal “Connection grouping ID” (A.2.73). Each road marking connection is defined by a unique signal “Connection grouping ID” (A.2.73) value and connects two or more road marking contours.

7.4.3 Profile: Colour model for RDOI

The profile is relevant for camera sensor technology (B.2.1). The signal “Colour model type” (A.1.15) defines the colour model for the interface. Depending on the colour model, each colour is described by a fixed number of colour values [see signal “Colour value – object level” (A.2.69)] to define the colour tone. The confidence of the colour tone is provided by the signal “Colour tone – confidence – object level” (A.2.70).

The interface provides colour tones for road markings and road boundaries.

7.5 Static object interface

The static object interface (SOI) consists of three types of entities: general landmarks, traffic signs (traffic main sign and additional supplementary signs) and traffic lights (see Figure 12). Table 15 defines the logical structure of the static object interface.

Table 15 — Static object interface

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
Information: interface	M	Interface version ID {major, minor, patch} (A.1.1)	M	Profile: Uniqueness of interface versioning (6.4)
		Number of valid serving sensors (A.1.2)	M	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
		Size type: dynamic/fixed Size #: Number of valid serving sensors (A.1.2)		
		Sensor ID (A.1.3)	M	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
		Interface ID (A.1.4)	O	Profile: Uniqueness of interface versioning (6.4)
		Time stamp – predicted (A.1.5.1)	M	
		Cycle counter (A.1.6.1)	O	Redundancy (B.1.2) Signal: Time stamp – predicted (A.1.5.1)
		Interface cycle time (A.1.7)	O	

Table 15 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
		Interface cycle time – variation (A.1.8)	O	
		Data qualifier (A.1.9)	M	
		Colour model type (A.1.15)	C (B.2.1) Relevant: camera	Profile: Colour model for SOI (7.5.3) Optimise (B.1.3) Use enumeration to define colours by defining colour values and the applied colour model for each enumerator.
General landmarks	C (B.2.3) multiple entity types	Recognised general landmarks – capability (A.1.10.4)	O	
		Recognised general landmarks – status (A.1.11.4)	O	Redundancy (B.1.2) Signal: Recognised general landmarks – capability (A.1.10.4)
		Number of valid general landmarks (A.1.12.4)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid general landmarks (A.1.12.4)		
↪ General landmark status	M	Existence probability – object level (A.2.1)	M	
		Object ID (A.2.2)	M	Alternative (B.3.2)
		Object grouping ID (A.2.3)	O	
		Age (A.2.4)	M	
		Number of valid observations – object level (A.2.5)	O	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – object level (A.2.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
		Size type: dynamic/fixed Size #: Number of valid observations – object level (A.2.5)		

Table 15 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
		Time stamp reference – object level (A.2.6)	C (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Alternative (B.1.3)
		Observation status – object level (A.2.7)	C (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Redundancy (B.1.2) Signal: Object ID (A.2.2), Time stamp – predicted (A.1.5.1)
		Track quality (A.2.8)	O	
		Measurement status – object level (A.2.9)	M	
↪ General landmark information	M	Number of valid general landmark classifications (A.2.104)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: General landmark classification type (A.2.105)
		Size type: dynamic/fixed Size #: Number of valid general landmark classifications (A.2.104)		
		General landmark classification type (A.2.105)	M	Alternative (B.1.3)
		General landmark classification type – confidence (A.2.106)	M	

Table 15 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ General landmark position	M	Position – object level {x, y, z} (A.2.13)	{M, M, O}	
		Position – object level {x, y, z} – error (A.2.14)	{M, M, O}	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Orientation {yaw, pitch, roll} (A.2.15)	{C (B.2.1) Relevant: camera, O, O}	
		Orientation {yaw, pitch, roll} – error (A.2.16)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Reference point (A.2.17)	O	
↪ General landmark bounding box	C (B.2.1) Relevant: camera	Bounding box extent {length, width, height} (A.2.19)	{M, M, O}	
		Bounding box extent {length, width, height} – error (A.2.20)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
Traffic signs	C (B.2.3) multiple entity types	Recognised traffic signs – capability (A.1.10.5)	O	
		Recognised traffic signs – status (A.1.11.5)	O	Redundancy (B.1.2) Signal: Recognised traffic signs – capability (A.1.10.5)
		Number of valid traffic signs (A.1.12.5)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid traffic signs (A.1.12.5)		
↪ Traffic main sign status	M	Existence probability – object level (A.2.1)	M	
		Object ID (A.2.2)	M	Alternative (B.3.2)
		Object grouping ID (A.2.3)	O	
		Age (A.2.4)	M	

Table 15 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
		Number of valid observations – object level (A.2.5)	O	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – object level (A.2.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
		Size type: dynamic/fixed Size #: Number of valid observations – object level (A.2.5)		
		Time stamp reference – object level (A.2.6)	C (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Alternative (B.1.3)
		Observation status – object level (A.2.7)	C (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Redundancy (B.1.2) Signal: Object ID (A.2.2), Time stamp – predicted (A.1.5.1)
		Track quality (A.2.8)	O	
		Measurement status – object level (A.2.9)	M	
↪ Traffic main sign information	M	Number of valid sign classifications (A.2.63)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sign classification type (A.2.64)
		Size type: dynamic/fixed Size #: Number of valid sign classifications (A.2.63)		
		Sign classification type (A.2.64)	M	Alternative (B.1.3)
		Sign classification type – confidence (A.2.65)	M	
		Sign value (A.2.66)	M	
		Sign value unit (A.2.67)	M	
		Sign state (A.2.68)	M	

Table 15 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
		Sign geometry (A.2.107)	O	
		Number of valid lane relevance classifications (A.2.108)	O	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Lane relevance classification type (A.2.109)
		Size type: dynamic/fixed Size #: Number of valid lane relevance classifications (A.2.108)		
		Lane relevance classification type (A.2.109)	C (B.2.2) Signal:	Alternative (B.1.3)
		Lane relevance classification type – confidence (A.2.110)	C (B.2.2) Signal:	
↪ Traffic main sign colour tone	C (B.2.1) Relevant: camera	Size type: dynamic/fixed Size # * Implicit list length – Optimise (B.1.1): List length optimisation: depends on Colour model type (A.1.15)		
		Colour value – object level (A.2.69)	M	Profile: Colour model for SOI (7.5.3)
		Colour tone – confidence – object level (A.2.70)	O	Profile: Colour model for SOI (7.5.3)
↪ Traffic main sign position	M	Position – object level {x, y, z} (A.2.13)	{M, M, O}	
		Position – object level {x, y, z} – error (A.2.14)	{M, M, O}	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
↪ Supplementary signs	M	Number of valid traffic supplementary signs (A.2.111)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid traffic supplementary signs (A.2.111)		

Table 15 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ ↪ Supplementary sign status	M	Existence probability – object level (A.2.1)	Mandatory	
		Age (A.2.4)	M	
		Number of valid observations – object level (A.2.5)	O	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – object level (A.2.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
		Size type: dynamic/fixed Size #: Number of valid observations – object level (A.2.5)		
		Time stamp reference – object level (A.2.6)	C (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Alternative (B.1.3)
		Observation status – object level (A.2.7)	C (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Redundancy (B.1.2) Signal: Object ID (A.2.2), Time stamp – predicted (A.1.5.1)
		Track quality (A.2.8)	O	
		Measurement status – object level (A.2.9)	M	

Table 15 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ ↪ Supplementary sign information	M	Number of valid supplementary sign classifications (A.2.112)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Supplementary sign classification type (A.2.113)
		Size type: dynamic/fixed Size #: Number of valid supplementary sign classifications (A.2.112)		
		Supplementary sign classification type (A.2.113)	M	Alternative (B.3.2)
		Supplementary sign classification type – confidence (A.2.114)	M	
		Sign value (A.2.66)	M	
		Sign value unit (A.2.67)	M	
		Sign state (A.2.68)	M	
↪ ↪ Supplementary sign colour tone	C (B.2.1) Relevant: camera	Size type: dynamic/fixed Size # * Implicit list length – Optimise (B.1.1): List length optimisation: depends on Colour model type (A.1.15)		
		Colour value – object level (A.2.69)	M	Profile: Colour model for SOI (7.5.3)
		Colour tone – confidence – object level (A.2.70)	O	Profile: Colour model for SOI (7.5.3)
↪ ↪ Supplementary sign position	M	Relative position (A.2.115)	M	
		Relative position order (A.2.116)	M	
Traffic lights	C (B.2.3) multiple entity types	Recognised traffic lights – capability (A.1.10.6)	O	
		Recognised traffic lights – status (A.1.11.6)	O	Redundancy (B.1.2) Signal: Recognised traffic lights – capability (A.1.10.6)
		Number of valid traffic lights (A.1.12.6)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid traffic lights (A.1.12.6)		

Table 15 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ Traffic light status	M	Existence probability – object level (A.2.1)	M	
		Object ID (A.2.2)	M	Alternative (B.3.2)
		Object grouping ID (A.2.3)	O	
		Age (A.2.4)	M	
		Number of valid observations – object level (A.2.5)	O	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – object level (A.2.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
		Size type: dynamic/fixed Size #: Number of valid observations – object level (A.2.5)		
		Time stamp reference – object level (A.2.6)	C (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Alternative (B.1.3)
		Observation status – object level (A.2.7)	C (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Redundancy (B.1.2) Signal: Object ID (A.2.2), Time stamp – predicted (A.1.5.1)
		Track quality (A.2.8)	O	
		Measurement status – object level (A.2.9)	M	

Table 15 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ Traffic light information	M	Number of valid structure light classifications (A.2.117)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Structure light classification type (A.2.118)
		Size type: dynamic/fixed Size #: Number of valid structure light classifications (A.2.117)		
		Structure light classification type (A.2.118)	M	Alternative (B.1.3)
		Structure light classification type – confidence (A.2.119)	M	
↪ Traffic light position	M	Position – object level {x, y, z} (A.2.13)	{M, M, O}	
		Position – object level {x, y, z} – error (A.2.14)	{M, M, O}	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Orientation {yaw, pitch, roll} (A.2.15)	{C (B.2.1) Relevant: camera, O, O}	
		Orientation {yaw, pitch, roll} – error (A.2.16)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Reference point (A.2.17)	O	
		Minimum visibility distance (A.2.120)	M	
↪ Traffic light bounding box	C (B.2.1) Relevant: camera	Bounding box extent {length, width, height} (A.2.19)	{O, M, M}	
		Bounding box extent {length, width, height} – error (A.2.20)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)

Table 15 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ Traffic light spots	M	Total number of traffic light spots (A.2.121)	O	
		Total number of traffic light spots – confidence (A.2.122)	O	
		Number of valid traffic light spots (A.2.123)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid traffic light spots (A.2.123)		
↪ ↪ Traffic light spot status	M	Existence probability – object level (A.2.1)	M	
		Age (A.2.4)	M	
		Number of valid observations – object level (A.2.5)	O	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – object level (A.2.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
		Size type: dynamic/fixed Size #: Number of valid observations – object level (A.2.5)		
		Time stamp reference – object level (A.2.6)	C (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Alternative (B.1.3)
		Observation status – object level (A.2.7)	C (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Redundancy (B.1.2) Signal: Object ID (A.2.2), Time stamp – predicted (A.1.5.1)
		Track quality (A.2.8)	O	
		Measurement status – object level (A.2.9)	M	

Table 15 *(continued)*

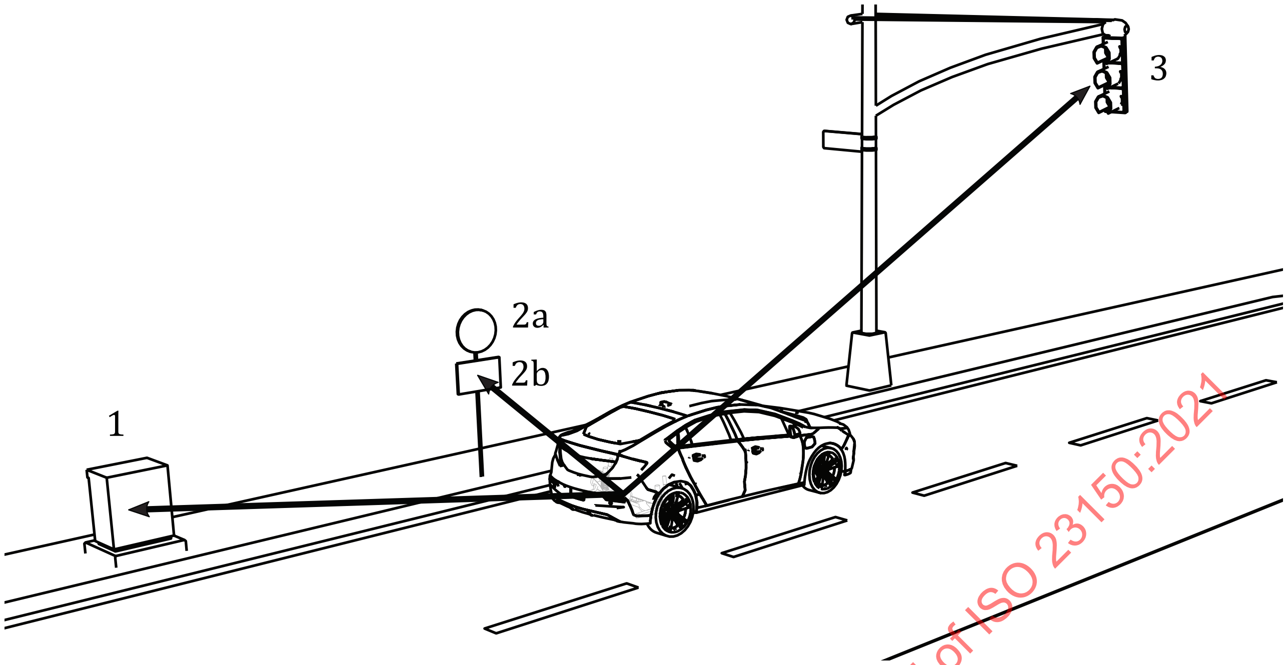
LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ ↪ Traffic light spot information	M	Number of valid light shape classifications (A.2.124)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Light shape classification type (A.2.125)
		Size type: dynamic/fixed Size #: Number of valid light shape classifications (A.2.124)		
		Light shape classification type (A.2.125)	M	Alternative (B.1.3)
		Light shape classification type – confidence (A.2.126)	M	
		Light shape value (A.2.127)	C (B.2.2) * The signal “Light shape classification type” (A.2.125) has an enumerator “LightShape_CountdownSecond”, “LightShape_CountdownPercent” or a similar enumerator defined during the system desing phase.	

Table 15 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ ↪ Traffic light spot colour	C (B.2.1) Relevant: camera	Number of valid colour classifications (A.2.128)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Colour classification type (A.2.129)
		Size type: dynamic/fixed Size #: Number of valid colour classifications (A.2.128)		
		Colour classification type (A.2.129)	M	Alternative (B.1.3)
		Colour classification type – confidence (A.2.130)	M	
		Number of valid light mode classifications (A.2.131)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Light mode classification type (A.2.132)
		Size type: dynamic/fixed Size #: Number of valid light mode classifications (A.2.131)		
		Light mode classification type (A.2.132)	M	Alternative (B.1.3)
		Light mode classification type – confidence (A.2.133)	M	

Table 15 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ ↪ Traffic light spot position	M	Position – object level {x, y, z} (A.2.13)	{M, M, O}	
		Position – object level {x, y, z} – error (A.2.14)	{M, M, O}	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Number of valid lane relevance classifications (A.2.108)	O	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Lane relevance classification type (A.2.109)
		Size type: dynamic/fixed Size #: Number of valid lane relevance classifications (A.2.108)		
		Lane relevance classification type (A.2.109)	C (B.2.2) Signal:	Alternative (B.1.3)
		Lane relevance classification type – confidence (A.2.110)	C (B.2.2) Signal:	



Key

1 general landmark

2a main sign of a traffic sign

2b supplementary sign of a traffic sign

3 traffic light (with 3 traffic light spots)

Figure 12 — Example for static objects

7.5.1 Static object header

Table 16 defines the interface header in subclause "Static object header" (7.5.1) and the changes due to the adaptation in comparison to the generic interface header in subclause "Generic object level header" (7.2.1). The header of the SOI can contain a list of general landmarks, a list of valid traffic signs and/or a list of valid traffic light entities [see subclause "Static object entity" (7.5.2)].

Table 16 — Specific signal grouping: Static object header

Signal	RL signal	Option
Interface version ID {major, minor, patch} (A.1.1)	Mandatory	Profile: Uniqueness of interface versioning (6.4)
Number of valid serving sensors (A.1.2)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
↪ Sensor ID (A.1.3)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
Interface ID (A.1.4)	Optional	Profile: Uniqueness of interface versioning (6.4)
Time stamp – predicted (A.1.5.1)	Mandatory	

Table 16 *(continued)*

Signal	RL signal		Option
Cycle counter (A.1.6.1)	Optional		Redundancy (B.1.2) Signal: Time stamp – predicted (A.1.5.1)
Interface cycle time (A.1.7)	Optional		
Interface cycle time – variation (A.1.8)	Optional		
Data qualifier (A.1.9)	Mandatory		
Colour model type (A.1.15)	Conditional (B.2.1) Relevant: camera		Profile: Colour model for SOI (7.5.3) Optimise (B.1.3) Use enumeration to define colours by defining colour values and the applied colour model for each enumerator.
Recognised general landmarks – capability (A.1.10.4)	RL LSG conditional (B.2.3) multiple entity types	Optional	
Recognised general landmarks – status (A.1.11.4)		Optional	Redundancy (B.1.2) Signal: Recognised general landmarks – capability (A.1.10.4)
Number of valid general landmarks (A.1.12.4)		Mandatory	Optimise (B.1.1) List length optimisation
Recognised traffic signs – capability (A.1.10.5)		Optional	
Recognised traffic signs – status (A.1.11.5)		Optional	Redundancy (B.1.2) Signal: Recognised traffic signs – capability (A.1.10.5)
Number of valid traffic signs (A.1.12.5)		Mandatory	Optimise (B.1.1) List length optimisation
Recognised traffic lights – capability (A.1.10.6)		Optional	
Recognised traffic lights – status (A.1.11.6)		Optional	Redundancy (B.1.2) Signal: Recognised traffic lights – capability (A.1.10.6)
Number of valid traffic lights (A.1.12.6)		Mandatory	Optimise (B.1.1) List length optimisation

The following LSGs, which normally shall be provided by the sensor cluster in another interface, may be added to this header, if, for example, the other interface is not implemented by the sensor cluster [see subclause "Need of logical signal group" (B.3.3)]:

— interface: SPI; LSG "Information: vehicle coordinate system" (see Table 46);

— interface: SHII; LSG "Calibration" (see Table 48);

— interface: SHII; LSG "Sensor cluster" (see Table 48).

7.5.2 Static object entity

Each static object describes a recognised real-world general landmark, traffic sign or traffic light. They consist of several LSGs. The traffic sign consists of a main sign and none, one or more supplementary signs. A main sign is a traffic sign that can be followed on its own (for example, a stop sign). A supplementary sign is a sign with conditions under which the main sign is valid (for example, under wet conditions only).

A general landmark is described by the following LSGs.

— **Status:** the status describes general information of the tracked general landmark. This information is based on history of the object tracking. Table 17 defines the signal grouping "Static object entity status". It defines this status LSG of the general landmark and redefines the signal grouping "Generic object level entity status" (see Table 8).

— **Information:** the information describes the general landmark type and confidence.

— **Position:** the position describes the position information of the tracked static object.

— **Bounding box:** the bounding box describes the 3D rectangular hull that encloses the recognised static object.

A traffic sign is described by the following LSGs.

— **Main sign status:** the main sign status describes general information of the tracked main sign. This information is based on history of the object tracking. Table 17 defines the signal grouping "Static object entity status". It defines this status LSG of the main sign and redefines the signal grouping "Generic object level entity status" (see Table 8).

— **Main sign information:** the main sign information describes the semantic of the main sign.

— **Main sign colour tone:** the main sign colour tone describes visual properties of the main sign.

— **Main sign position:** the main sign position describes the geometric position of the main sign.

— A list of supplementary signs assigned to the main sign:

  — **Supplementary sign status:** the supplementary sign status describes general information of the tracked supplementary sign. This information is based on history of the object tracking and is associated to the main sign. A supplementary sign uses a subset of signals of the signal grouping "Static object entity status" (Table 17) for this LSG.

  — **Supplementary sign information:** the supplementary sign information describes the supplementary sign type and confidence.

  — **Supplementary sign colour tone:** the supplementary sign colour tone describes visual properties of the supplementary sign.

  — **Supplementary sign position:** the supplementary sign position describes the qualitative position of the supplementary sign with respect to the main sign and other supplementary signs.

A traffic light is described by the following LSGs.

— **Light status:** the traffic light status describes general information of the tracked traffic light. This information is based on history of the object tracking. Table 17 defines the signal grouping "Static object entity status". It defines this status LSG of the traffic light and redefines the signal grouping "Generic object level entity status" (see Table 8).

— **Light information:** the information describes the certainty and the type of the tracked traffic light.

— **Light position:** the position describes the position information of the tracked traffic light.

— **Light bounding box:** the bounding box describes the 3D rectangular hull that encloses the tracked traffic light.

— A list of traffic light spots assigned to the traffic light:

  — **Spot status:** the traffic light spot status describes general information of the tracked light spot. This information is based on history of the object tracking and is associated to the traffic light.

A traffic light spot uses a subset of signals of the signal grouping “Static object entity status” (Table 17) for this LSG.

— **Spot information:** the traffic light spot information describes the semantic of the tracked light spot.

— **Spot colour:** the traffic light spot colour describes the basic light colour of the tracked light spot.

— **Spot position:** the traffic light spot position describes the position of the tracked light spot which is relative to the vehicle coordinate system [see signal “Vehicle coordinate system type” (A.1.21)]. All traffic light spots are inside the traffic light bounding box.

Table 17 — Specific signal grouping: Static object entity status

Signal	RL signal	Option
Existence probability – object level (A.2.1)	Mandatory	
Object ID (A.2.2)	Mandatory	Alternative (B.3.2)
Object grouping ID (A.2.3)	Optional	
Age (A.2.4)	Mandatory	
Number of valid observations – object level (A.2.5)	Optional	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – object level (A.2.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
↪ Time stamp reference – object level (A.2.6)	Conditional (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Alternative (B.1.3)
↪ Observation status – object level (A.2.7)	Conditional (B.2.2) Signal: Number of valid observations – object level (A.2.5)	Redundancy (B.1.2) Signal: Object ID (A.2.2), Time stamp – predicted (A.1.5.1)
Track quality (A.2.8)	Optional	
Measurement status – object level (A.2.9)	Mandatory	

7.5.3 Profile: Colour model for SOI

The profile is relevant for camera sensor technology (B.2.1). The signal “Colour model type” (A.1.15) defines the colour model for the interface. Depending on the colour model, each colour is described by a fixed number of colour values [see signal “Colour value – object level” (A.2.69)] to define the colour tone. The confidence of the colour tone is provided by the signal “Colour tone – confidence – object level” (A.2.70).

The interface provides colour tones for main signs and supplementary signs.

8 Feature level

8.1 General

The feature level interfaces (FLI) originate from several sensors of a sensor cluster. Therefore, at feature level, the term sensor cluster is always used, even if a single sensor is serving the interface. The FLIs provide recognised features, where recognition is based on multiple sensors of the sensor cluster, a small amount of historical data or no historical data at all. The sensor cluster refers its recognised features to one of two predefined vehicle coordinate systems [see signal “Vehicle coordinate system type” (A.1.21)]. The feature properties depend on the sensor technology of the sensor cluster. To bring recognised features in relation to entities of other interface levels, each recognised feature has a unique signal’s “Feature ID” (A.3.2) value. The uniqueness of the signal’s “Feature ID” (A.3.2) value is only guaranteed within the sensor cluster’s interfaces.

Wrong link-relations between detections and features shall be prevented. Specifically, the reuse of the signal’s “Feature ID” (A.3.2) values shall not lead to a misinterpretation of the link-relation between entities of the sensor cluster’s interfaces.

On feature level the following interfaces are available:

— camera feature interface (8.3);

— ultrasonic feature interface (8.4).

There is no FLI for radar or lidar sensor clusters defined.

8.2 Generic sensor cluster feature interface

Table 18 defines the generic structure of an FLI.

Table 18 — Generic sensor feature interface

Structure	Multiplicity	Option
Generic sensor cluster feature header (8.2.1)	1	
Generic sensor cluster feature entity (8.2.2), specifically an individual feature list interface entity	Multiple	Size type: dynamic/fixed

8.2.1 Generic sensor cluster feature header

Table 19 defines the interface header in subclause “Generic sensor cluster feature header” (8.2.1) and the changes due to the adaptation in comparison to the generic interface header in subclause “Generic interface header” (6.2).

Table 19 — Specific signal grouping: Generic sensor cluster feature header

Signal	RL signal	Option
Interface version ID {major, minor, patch} (A.1.1)	Mandatory	Profile: Uniqueness of interface versioning (6.4)
Number of valid serving sensors (A.1.2)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
↪ Sensor ID (A.1.3)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
Interface ID (A.1.4)	Optional	Profile: Uniqueness of interface versioning (6.4)
Time stamp – predicted (A.1.5.1)	Mandatory	
Cycle counter (A.1.6.1)	Optional	Redundancy (B.1.2) Signal: Time stamp – predicted (A.1.5.1)
Interface cycle time (A.1.7)	Optional	
Interface cycle time – variation (A.1.8)	Optional	
Data qualifier (A.1.9)	Mandatory	

Table 19 *(continued)*

Signal	RL signal	Option
FLI sensor technology specific header extensions		
Recognised features – capability (A.1.10.7)	Optional	
Recognised features – status (A.1.11.7)	Optional	Redundancy (B.1.2) Signal: Recognised features – capability (A.1.10.7)
Number of valid features (A.1.12.7)	Mandatory	Optimise (B.1.1) List length optimisation

The following LSGs, which normally shall be provided by the sensor cluster in another interface, may be added to this header, if, for example, the other interface is not implemented by the sensor cluster [see subclause “Need of logical signal group” (B.3.3)]:

— interface: SPI; LSG “Information: vehicle coordinate system” (see Table 46);

— interface: SHII; LSG “Calibration” (see Table 48);

— interface: SHII; LSG “Sensor cluster” (see Table 48).

8.2.2 Generic sensor cluster feature entity

Table 20 defines the generic signal grouping “Generic sensor cluster feature entity status” for each FLI and defines the status of a feature.

Table 20 — Specific signal grouping: Generic sensor cluster feature entity status

Signal	RL signal	Option
Existence probability – feature level (A.3.1)	Mandatory	
Feature ID (A.3.2)	Conditional (B.3.1) Exist: DLI	Alternative (B.3.2)
Object ID reference – feature level (A.3.3)	Conditional (B.3.1) Exist: OLI	Alternative (B.3.2)
Time stamp difference – feature level (A.3.4)	Mandatory	Optimise (B.1.5)
Number of valid observations – feature level (A.3.5)	Optional	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – feature level (A.3.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
↪ Time stamp reference – feature level (A.3.6)	Conditional (B.2.2) Signal: Number of valid observations – feature level (A.3.5)	Alternative (B.1.3)
↪ Observation status – feature level (A.3.7)	Conditional (B.2.2) Signal: Number of valid observations – feature level (A.3.5)	Redundancy (B.1.2) Signal: Feature ID (A.3.2) (if unique over time), Time stamp – predicted (A.1.5.1)

Each interface type adds individual properties to the interface entity type. Geometric information of the feature references the vehicle coordinate system [see signal “Vehicle coordinate system type” (A.1.21)]. Features reference objects of the sensor cluster. Likewise, detections from individual sensors of a sensor cluster are referencing features.

8.3 Camera feature interface

Table 21 defines the logical structure of the camera feature interface (CFI).

Table 21 — Camera feature interface

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
Information: interface	M	Interface version ID {major, minor, patch} (A.1.1)	M	Profile: Uniqueness of interface versioning (6.4)
		Number of valid serving sensors (A.1.2)	M	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
		Size type: dynamic/fixed Size #: Number of valid serving sensors (A.1.2)		
		Sensor ID (A.1.3)	M	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
		Interface ID (A.1.4)	O	Profile: Uniqueness of interface versioning (6.4)
		Time stamp – predicted (A.1.5.1)	M	
		Cycle counter (A.1.6.1)	O	Redundancy (B.1.2) Signal: Time stamp – predicted (A.1.5.1)
		Interface cycle time (A.1.7)	O	
		Interface cycle time – variation (A.1.8)	O	
		Data qualifier (A.1.9)	M	
		Colour model type (A.1.15)	M	Profile: Colour model for CFI (8.3.3) Optimise (B.1.3) Use enumeration to define colours by defining colour values and the applied colour model for each enumerator.

Table 21 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
Features	M	Recognised features – capability (A.1.10.7)	O	
		Recognised features – status (A.1.11.7)	O	Redundancy (B.1.2) Signal: Recognised features – capability (A.1.10.7)
		Number of valid features (A.1.12.7)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid features (A.1.12.7)		
↪ Status	M	Existence probability – feature level (A.3.1)	M	
		Feature ID (A.3.2)	C (B.3.1) Exist: DLI	Alternative (B.3.2)
		Feature grouping ID (A.3.8)	O	
		Object ID reference – feature level (A.3.3)	C (B.3.1) Exist: OLI	Alternative (B.3.2)
		Time stamp difference – feature level (A.3.4)	M	Optimise (B.1.5)
		Number of valid observations – feature level (A.3.5)	O	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – feature level (A.3.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
		Size type: dynamic/fixed Size #: Number of valid observations – feature level (A.3.5)		
		Time stamp reference – feature level (A.3.6)	Conditional (B.2.2) Signal: Number of valid observations – feature level (A.3.5)	Alternative (B.1.3)
		Observation status – feature level (A.3.7)	Conditional (B.2.2) Signal: Number of valid observations – feature level (A.3.5)	Redundancy (B.1.2) Signal: Feature ID (A.3.2) (if unique over time), Time stamp – predicted (A.1.5.1)

Table 21 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ Shape information	M	Number of valid shape classifications – feature level (A.3.9)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Shape classification type – feature level (A.3.10)
		Size type: dynamic/fixed Size #: Number of valid shape classifications – feature level (A.3.9)		
		Shape classification type – feature level (A.3.10)	M	Alternative (B.1.3)
		Shape classification type – confidence – feature level (A.3.11)	M	
↪ Shape colour tone	M	Size type: dynamic/fixed Size # * Implicit list length – Optimise (B.1.1): List length optimisation: depends on Colour model type (A.1.15)		
		Colour value – feature level (A.3.12)	M	Profile: Colour model for CFI (8.3.3)
		Colour tone – confidence – feature level (A.3.13)	O	Profile: Colour model for CFI (8.3.3)
↪ Shape points	M	Shape type – feature level (A.3.14)	M	
		Number of valid shape points – feature level (A.3.15)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid shape points – feature level (A.3.15)		
		Point existence probability – feature level (A.3.16)	M	
		Position – feature level {x, y, z} (A.3.17)	M	
		Position – feature level {x, y, z} – error (A.3.18)	M	Implementation (B.4.1) Error model implementation Alternative (B.1.3)

Table 21 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ Shape reference points	O	Number of valid shape reference points – feature level (A.3.19)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid shape reference points – feature level (A.3.19)		
		Point existence probability – feature level (A.3.16)	M	
		Position – feature level {x, y, z} (A.3.17)	M	
		Position – feature level {x, y, z} – error (A.3.18)	M	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Shape surface normal {x, y, z} (A.3.20)	O	
		Shape surface normal {x, y, z} – error (A.3.21)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Translation rate {x, y, z} – feature level (A.3.22)	O	
		Translation rate {x, y, z} – feature level – error (A.3.23)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Rotation rate {yaw, pitch, roll} (A.3.24)	O	
		Rotation rate {yaw, pitch, roll} – error (A.3.25)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Scale change – feature level (A.3.26)	O	
		Scale change – feature level – error (A.3.27)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)

8.3.1 Camera feature header

Table 22 defines the interface header in subclause "Camera feature header" (8.3.1) and the changes due to the adaptation in comparison to the generic interface header in subclause "Generic sensor cluster feature header" (8.2.1). The header of the CFI can contain a list of valid camera feature entities [see subclause "Camera feature entity" (8.3.2)].

Table 22 — Specific signal grouping: Camera feature header

Signal	RL signal	Option
Interface version ID {major, minor, patch} (A.1.1)	Mandatory	Profile: Uniqueness of interface versioning (6.4)
Number of valid serving sensors (A.1.2)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
↪ Sensor ID (A.1.3)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
Interface ID (A.1.4)	Optional	Profile: Uniqueness of interface versioning (6.4)
Time stamp – predicted (A.1.5.1)	Mandatory	
Cycle counter (A.1.6.1)	Optional	Redundancy (B.1.2) Signal: Time stamp – predicted (A.1.5.1)
Interface cycle time (A.1.7)	Optional	
Interface cycle time – variation (A.1.8)	Optional	
Data qualifier (A.1.9)	Mandatory	
Colour model type (A.1.15)	Mandatory	Profile: Colour model for CFI (8.3.3) Optimise (B.1.3) Use enumeration to define colours by defining colour values and the applied colour model for each enumerator.
Recognised features – capability (A.1.10.7)	Optional	
Recognised features – status (A.1.11.7)	Optional	Redundancy (B.1.2) Signal: Recognised features – capability (A.1.10.7)
Number of valid features (A.1.12.7)	Mandatory	Optimise (B.1.1) List length optimisation

The following LSGs, which normally shall be provided by the sensor cluster in another interface, may be added to this header, if, for example, the other interface is not implemented by the sensor cluster [see subclause “Need of logical signal group” (B.3.3)]:

— interface: SPI; LSG “Information: vehicle coordinate system” (see Table 46);

— interface: SHII; LSG “Calibration” (see Table 48);

— interface: SHII; LSG “Sensor cluster” (see Table 48).

8.3.2 Camera feature entity

For the camera, features are understood as shapes extracted from camera images. The shapes describe regions with the same semantic content (see Figure 13).

Each feature of a camera sensor cluster describes a recognised feature and consists of several LSGs.

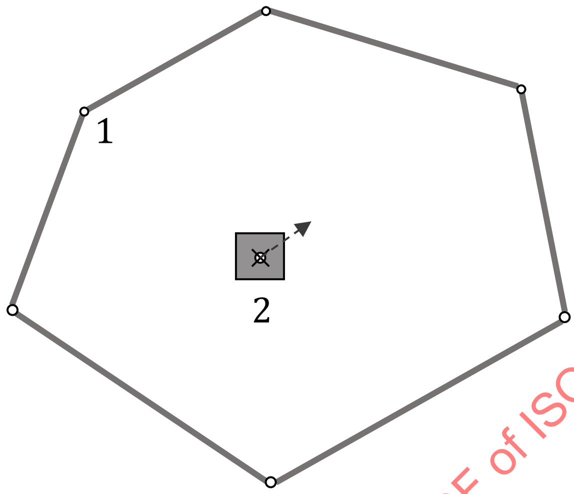
— **Status:** the status describes general information of the feature and provides an additional grouping ID to group interconnected camera features. This information is based on basic feature history information. Table 23 defines the signal grouping “Camera feature entity status”. It defines this status LSG of the feature and redefines the signal grouping “Generic sensor cluster feature entity status” (see Table 20).

— **Shape information:** the shape information describes the attributes of the tracked shape.

— **Shape colour tone:** the colour tone describes visual properties of the shape.

— **Shape points:** the shape points describe the geometric vertices of the hull for the feature shape.

— **Shape reference points:** the shape reference points are small distinctive trackable segments which are part of the feature's shape.



Key

1 shape with shape points

2 shape reference point with approximated tangential plane and orientation normal

Figure 13 — Example for a camera feature shape

Table 23 — Specific signal grouping: Camera feature entity status

Signal	RL signal	Option
Existence probability – feature level (A.3.1)	Mandatory	
Feature ID (A.3.2)	Conditional (B.3.1) Exist: DLI	Alternative (B.3.2)
Feature grouping ID (A.3.8)	Optional	
Object ID reference – feature level (A.3.3)	Conditional (B.3.1) Exist: OLI	Alternative (B.3.2)
Time stamp difference – feature level (A.3.4)	Mandatory	Optimise (B.1.5)
Number of valid observations – feature level (A.3.5)	Optional	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – feature level (A.3.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
↪ Time stamp reference – feature level (A.3.6)	Conditional (B.2.2) Signal: Number of valid observations – feature level (A.3.5)	Alternative (B.1.3)
↪ Observation status – feature level (A.3.7)	Conditional (B.2.2) Signal: Number of valid observations – feature level (A.3.5)	Redundancy (B.1.2) Signal: Feature ID (A.3.2) (if unique over time), Time stamp – predicted (A.1.5.1)

8.3.3 Profile: Colour model for CFI

The signal "Colour model type" (A.1.15) defines the colour model for the interface. Depending on the colour model, each colour is described by a fixed number of colour values [see signal "Colour value – feature level" (A.3.12)] to define the colour tone. The confidence of the colour tone is provided by the signal "Colour tone – confidence – feature level" (A.3.13).

The interface provides colour tones for features.

8.4 Ultrasonic feature interface

Table 24 defines the logical structure of the ultrasonic feature interface (UFI).

Table 24 — Ultrasonic feature interface

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
Information: interface	M	Interface version ID {major, minor, patch} (A.1.1)	M	Profile: Uniqueness of interface versioning (6.4)
		Number of valid serving sensors (A.1.2)	M	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
		Size type: dynamic/fixed Size #: Number of valid serving sensors (A.1.2)		
		Sensor ID (A.1.3)	M	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
		Interface ID (A.1.4)	O	Profile: Uniqueness of interface versioning (6.4)
		Time stamp – predicted (A.1.5.1)	M	
		Cycle counter (A.1.6.1)	O	Redundancy (B.1.2) Signal: Time stamp – predicted (A.1.5.1)
		Interface cycle time (A.1.7)	O	
		Interface cycle time – variation (A.1.8)	O	
		Data qualifier (A.1.9)	M	

Table 24 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
Features	M	Recognised features – capability (A.1.10.7)	O	
		Recognised features – status (A.1.11.7)	O	Redundancy (B.1.2) Signal: Recognised features – capability (A.1.10.7)
		Number of valid features (A.1.12.7)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid features (A.1.12.7)		
↪ Status	M	Existence probability – feature level (A.3.1)	M	
		Feature ID (A.3.2)	C (B.3.1) Exist: DLI	Alternative (B.3.2)
		Object ID reference – feature level (A.3.3)	C (B.3.1) Exist: OLI	Alternative (B.3.2)
		Time stamp difference – feature level (A.3.4)	M	Optimise (B.1.5)
		Number of valid observations – feature level (A.3.5)	O	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – feature level (A.3.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
		Size type: dynamic/fixed Size #: Number of valid observations – feature level (A.3.5)		
		Time stamp reference – feature level (A.3.6)	Conditional (B.2.2) Signal: Number of valid observations – feature level (A.3.5)	Alternative (B.1.3)
		Observation status – feature level (A.3.7)	Conditional (B.2.2) Signal: Number of valid observations – feature level (A.3.5)	Redundancy (B.1.2) Signal: Feature ID (A.3.2) (if unique over time), Time stamp – predicted (A.1.5.1)

Table 24 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
↪ Segment information	M	Number of valid ultrasonic feature classifications (A.3.28)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Ultrasonic feature classification type (A.3.29)
		Size type: dynamic/fixed Size #: Number of valid ultrasonic feature classifications (A.3.28)		
		Ultrasonic feature classification type (A.3.29)	M	Alternative (B.1.3)
		Ultrasonic feature classification type – confidence (A.3.30)	M	
↪ Segment points	M	Number of valid points (A.3.31)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid points (A.3.31)		
		Position – feature level {x, y, z} (A.3.17)	{M, M, O}	
		Position – feature level {x, y, z} – error (A.3.18)	{M, M, O}	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Orientation – feature level {pitch} (A.3.32)	O	
		Orientation – feature level {pitch} – error (A.3.33)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Height – feature level (A.3.34)	O	
		Height – feature level – error (A.3.35)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Velocity {x, y} – feature level (A.3.36)	O	

Table 24 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
		Velocity {x, y} – feature level – error (A.3.37)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Trilateration status (A.3.38)	M	
		Measurement status – feature level (A.3.39)	O	

8.4.1 Ultrasonic feature header

Table 25 defines the interface header in subclause "Ultrasonic feature header" (8.4.1) and the changes due to the adaptation in comparison to the generic interface header in subclause "Generic sensor cluster feature header" (8.2.1). The header of the UFI can contain a list of valid ultrasonic feature entities [see subclause "Ultrasonic feature entity" (8.4.2)].

Table 25 — Specific signal grouping: Ultrasonic feature header

Signal	RL signal	Option
Interface version ID {major, minor, patch} (A.1.1)	Mandatory	Profile: Uniqueness of interface versioning (6.4)
Number of valid serving sensors (A.1.2)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
↪ Sensor ID (A.1.3)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
Interface ID (A.1.4)	Optional	Profile: Uniqueness of interface versioning (6.4)
Time stamp – predicted (A.1.5.1)	Mandatory	
Cycle counter (A.1.6.1)	Optional	Redundancy (B.1.2) Signal: Time stamp – predicted (A.1.5.1)
Interface cycle time (A.1.7)	Optional	
Interface cycle time – variation (A.1.8)	Optional	
Data qualifier (A.1.9)	Mandatory	
Recognised features – capability (A.1.10.7)	Optional	
Recognised features – status (A.1.11.7)	Optional	Redundancy (B.1.2) Signal: Recognised features – capability (A.1.10.7)
Number of valid features (A.1.12.7)	Mandatory	Optimise (B.1.1) List length optimisation

The following LSGs, which normally shall be provided by the sensor cluster in another interface, may be added to this header, if, for example, the other interface is not implemented by the sensor cluster [see subclause "Need of logical signal group" (B.3.3)]:

— interface: SPI; LSG "Information: vehicle coordinate system" (see Table 46);

— interface: SHII; LSG "Calibration" (see Table 48);

— interface: SHII; LSG "Sensor cluster" (see Table 48).

8.4.2 Ultrasonic feature entity

Each feature of an ultrasonic sensor cluster describes a recognised feature and consists of several LSGs.

— **Status:** the status describes general information of the feature. This information is based on basic feature history information. Table 26 defines the signal grouping "Ultrasonic feature entity status". It defines this status LSG of the feature and redefines the signal grouping "Generic sensor cluster feature entity status" (see Table 20).

— **Segment information:** the segment information describes the attributes of the tracked segment.

— **Segment points:** the segment points describe the geometric vertices of the hull for the feature segment.

Table 26 — Specific signal grouping: Ultrasonic feature entity status

Signal	RL signal	Option
Existence probability – feature level (A.3.1)	Mandatory	
Feature ID (A.3.2)	Conditional (B.3.1) Exist: DLI	Alternative (B.3.2)
Object ID reference – feature level (A.3.3)	Conditional (B.3.1) Exist: OLI	Alternative (B.3.2)
Time stamp difference – feature level (A.3.4)	Mandatory	Optimise (B.1.5)
Number of valid observations – feature level (A.3.5)	Optional	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Time stamp reference – feature level (A.3.6) * Time stamp reference – object level (A.2.6) as a reference to the n.th last message of the interface
↪ Time stamp reference – feature level (A.3.6)	Conditional (B.2.2) Signal: Number of valid observations – feature level (A.3.5)	Alternative (B.1.3)
↪ Observation status – feature level (A.3.7)	Conditional (B.2.2) Signal: Number of valid observations – feature level (A.3.5)	Redundancy (B.1.2) Signal: Feature ID (A.3.2) (if unique over time), Time stamp – predicted (A.1.5.1)

9 Detection level

9.1 General

Normally the detection level interfaces (DLI) originate from one sensing element. Therefore, at detection level, the term sensor is always used, even if, for example, a combination of emitting as well as sensing elements are serving the interface. The DLIs provide recognised detections. The detections may be based on limited historical data and are rated with individual existence probability values. Detections are defined in the individual sensor coordinate system of the sensor's sensing element. The properties of the detections depend on the sensor technology. The uniqueness of the signal's "Feature ID" (A.3.2) values as well as the signal's "Object ID" (A.2.2) values is only guaranteed within the sensor cluster's interfaces of the sensor.

On detection level the following interfaces are available:

— radar detection interface (9.3);

— lidar detection interface (9.4);

— camera detection interface (9.5);

— ultrasonic detection interface (9.6).

9.2 Generic sensor detection interface

Table 27 defines the generic structure of a DLI.

Table 27 — Generic sensor detection interface

Structure	Multiplicity	Option
Generic sensor detections header (9.2.1)	1	
Generic sensor detections entity (9.2.2), specifically an individual detection list interface entity	Multiple	Size type: dynamic/fixed

9.2.1 Generic sensor detections header

Table 28 defines the interface header in subclause "Generic sensor detections header" (9.2.1) and the changes due to the adaptation in comparison to the generic interface header in subclause "Generic interface header" (6.2).

Table 28 — Specific signal grouping: Generic sensor detections header

Signal	RL signal	Option
Interface version ID {major, minor, patch} (A.1.1)	Mandatory	Profile: Uniqueness of interface versioning (6.4)
Number of valid serving sensors (A.1.2)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
↪ Sensor ID (A.1.3)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
Interface ID (A.1.4)	Optional	Profile: Uniqueness of interface versioning (6.4)
Time stamp – measurement (A.1.5.2)	Mandatory	
Cycle counter (A.1.6.1)	Optional	Redundancy (B.1.2) Signal: Time stamp – measurement (A.1.5.2)
Interface cycle time (A.1.7)	Optional	
Interface cycle time – variation (A.1.8)	Optional	

Table 28 *(continued)*

Signal	RL signal	Option
Data qualifier (A.1.9)	Mandatory	
DLI sensor technology specific header extensions		
Recognised detections – capability (A.1.10.8)	Optional	
Recognised detections – status (A.1.11.8)	Optional	Redundancy (B.1.2) Signal: Recognised detections – capability (A.1.10.8)
Number of valid detections (A.1.12.8)	Mandatory	Optimise (B.1.1) List length optimisation Optimise (B.1.4) Alternative implementation

The following LSGs, which normally shall be provided by the sensor in another interface, may be added to this header, if, for example, the other interface is not implemented by the sensor [see subclause "Need of logical signal group" (B.3.3)]:

— interface: SPI; LSG "Information: vehicle coordinate system" (see Table 46);

— interface: SPI; LSG "Information: sensor pose" (see Table 46);

— interface: SHII; LSG "Calibration" (see Table 48);

— interface: SHII; LSG "Sensor cluster" (see Table 48).

9.2.2 Generic sensor detections entity

Table 29 defines the sensor grouping "Generic sensor detections entity status" for each DLI and defines the status of the detection.

Table 29 — Specific signal grouping: Generic sensor detections entity status

Signal	RL signal	Option
Existence probability – detection level (A.4.1)	Mandatory	
Object ID reference – detection level (A.4.2)	Conditional (B.3.1) Exist: OLI	Alternative (B.3.2)
Feature ID reference (A.4.3)	Conditional (B.3.1) Exist: FLI	Alternative (B.3.2)
Time stamp difference – detection level (A.4.4)	Mandatory	Optimise (B.1.5)

Each interface type adds individual properties to the interface entity type. Detections are described in the sensor coordinate system. Detections reference entities of the OLI and FLI of the sensor as well as the sensor cluster.

9.3 Radar detection interface

Table 30 defines the logical structure of the radar detection interface (RDI).

Table 30 — Radar detection interface

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
Information: interface	M	Interface version ID {major, minor, patch} (A.1.1)	M	Profile: Uniqueness of interface versioning (6.4)
		Number of valid serving sensors (A.1.2)	M	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
		Size type: dynamic/fixed Size #: Number of valid serving sensors (A.1.2)		
		Sensor ID (A.1.3)	M	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
		Interface ID (A.1.4)	O	Profile: Uniqueness of interface versioning (6.4)
		Time stamp – measurement (A.1.5.2)	M	
		Cycle counter (A.1.6.1)	O	Redundancy (B.1.2) Signal: Time stamp – measurement (A.1.5.2)
		Interface cycle time (A.1.7)	O	
		Interface cycle time – variation (A.1.8)	O	
		Data qualifier (A.1.9)	M	
Information: ambiguity domain	M	Radial velocity ambiguity domain {begin, end} (A.1.16)	C (B.2.1)	Profile: Radar ambiguity (9.3.3)
		Range ambiguity domain {begin, end} (A.1.17)	C (B.2.1)	Profile: Radar ambiguity (9.3.3)
		Angle azimuth ambiguity domain {begin, end} (A.1.18)	C (B.2.1)	Profile: Radar ambiguity (9.3.3)
		Angle elevation ambiguity domain {begin, end} (A.1.19)	C (B.2.1)	Profile: Radar ambiguity (9.3.3)

Table 30 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
Detections	M	Recognised detections – capability (A.1.10.8)	O	
		Recognised detections – status (A.1.11.8)	O	Redundancy (B.1.2) Signal: Recognised detections – capability (A.1.10.8)
		Number of valid detections (A.1.12.8)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Position {radial distance, azimuth, elevation} (A.4.16) identifiers {azimuth, elevation} Optimise (B.1.4) Alternative implementation
		Size type: dynamic/fixed Size #: Number of valid detections (A.1.12.8)		
↪ Status	M	Existence probability – detection level (A.4.1)	M	
		Object ID reference – detection level (A.4.2)	C (B.3.1) Exist: OLI	Alternative (B.3.2)
		Time stamp difference – detection level (A.4.4)	M	Optimise (B.1.5)
↪ Information	M	Radar cross section (A.4.5)	M	
		Radar cross section – error (A.4.6)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Signal to noise ratio – detection level (A.4.7)	M	
		Signal to noise ratio – detection level – error (A.4.8)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Multi target probability (A.4.9)	O	
		Ambiguity grouping ID (A.4.10)	C (B.2.1) ambiguity	Profile: Radar ambiguity (9.3.3)
		Detection ambiguity probability (A.4.11)	C (B.2.1) ambiguity	Profile: Radar ambiguity (9.3.3)
		Free space probability (A.4.12)	O	

Table 30 *(continued)*

LSG	RL LSG M/C/O	Signal	RL signal M/C/O	Option
		Number of valid detection classifications (A.4.13)	O	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Detection classification type (A.4.14)
		Size type: dynamic/fixed Size #: Number of valid detection classifications (A.4.13)		
		Detection classification type (A.4.14)	C (B.2.2) Signal: Number of valid detection classifications (A.4.13)	Alternative (B.1.3)
		Detection classification type – confidence (A.4.15)	C (B.2.2) Signal: Number of valid detection classifications (A.4.13)	
↪ Position	M	Position {radial distance, azimuth, elevation} (A.4.16)	{M, M, O}	Alternative (B.1.3) Signal: {azimuth, elevation}
		Position {radial distance, azimuth, elevation} – error (A.4.17)	{M, M, O}	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
↪ Dynamics	M	Relative velocity {radial distance} (A.4.18)	M	
		Relative velocity {radial distance} – error (A.4.19)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)

9.3.1 Radar detections header

Table 31 defines the interface header in subclause “Radar detections header” (9.3.1) and the changes due to the adaptation in comparison to the generic interface header in subclause “Generic sensor detections header” (9.2.1). The header of the RDI can contain a list of valid radar detection entities [see subclause “Radar detections entity” (9.3.2)].

Table 31 — Specific signal grouping: Radar detections header

Signal	RL signal	Option
Interface version ID {major, minor, patch} (A.1.1)	Mandatory	Profile: Uniqueness of interface versioning (6.4)
Number of valid serving sensors (A.1.2)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)

Table 31 *(continued)*

Signal	RL signal		Option
↪ Sensor ID (A.1.3)	Mandatory		Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
Interface ID (A.1.4)	Optional		Profile: Uniqueness of interface versioning (6.4)
Time stamp – measurement (A.1.5.2)	Mandatory		
Cycle counter (A.1.6.1)	Optional		Redundancy (B.1.2) Signal: Time stamp – measurement (A.1.5.2)
Interface cycle time (A.1.7)	Optional		
Interface cycle time – variation (A.1.8)	Optional		
Data qualifier (A.1.9)	Mandatory		
Radial velocity ambiguity domain {begin, end} (A.1.16)	RL LSG mandatory	Conditional (B.2.1) technology specific	Profile: Radar ambiguity (9.3.3)
Range ambiguity domain {begin, end} (A.1.17)		Conditional (B.2.1) technology specific	Profile: Radar ambiguity (9.3.3)
Angle azimuth ambiguity domain {begin, end} (A.1.18)		Conditional (B.2.1) technology specific	Profile: Radar ambiguity (9.3.3)
Angle elevation ambiguity domain {begin, end} (A.1.19)		Conditional (B.2.1) technology specific	Profile: Radar ambiguity (9.3.3)
Recognised detections – capability (A.1.10.8)	Optional		
Recognised detections – status (A.1.11.8)	Optional		Redundancy (B.1.2) Signal: Recognised detections – capability (A.1.10.8)
Number of valid detections (A.1.12.8)	Mandatory		Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Position {radial distance, azimuth, elevation} (A.4.16) identifiers {azimuth, elevation} Optimise (B.1.4) Alternative implementation

The following LSGs, which normally shall be provided by the sensor in another interface, may be added to this header, if, for example, the other interface is not implemented by the sensor [see subclause "Need of logical signal group" (B.3.3)]:

— interface: SPI; LSG "Information: vehicle coordinate system" (see Table 46);

— interface: SPI; LSG "Information: sensor pose" (see Table 46);

— interface: SHII; LSG "Calibration" (see Table 48);

— interface: SHII; LSG "Sensor cluster" (see Table 48).

9.3.2 Radar detections entity

Each detection of a radar sensor describes a measured detection and consists of several LSGs.

— **Status:** the status describes general information of the detection. It is based on current information and does not include historical information. Table 32 defines the signal grouping "Radar detections entity status". It defines this status LSG of the detection and redefines the signal grouping "Generic sensor detections entity status" (see Table 29). There is no radar feature interface defined and therefore radar detections cannot reference radar features.

— **Information:** the information describes the radar specific information of the detection.

— **Position:** the position describes the position information of the detection in the sensor coordinate system of the detection.

— **Dynamics:** the dynamics describes the dynamic information of the detection in the sensor coordinate system of the detection.

Table 32 — Specific signal grouping: Radar detections entity status

Signal	RL signal	Option
Existence probability – detection level (A.4.1)	Mandatory	
Object ID reference – detection level (A.4.2)	Conditional (B.3.1) Exist: OLI	Alternative (B.3.2)
Time stamp difference – detection level (A.4.4)	Mandatory	Optimise (B.1.5)

9.3.3 Profile: Radar ambiguity

The signal "Ambiguity grouping ID" (A.4.10) and the signal "Detection ambiguity probability" (A.4.11) are mandatory, if the radar sensor technology has one or more technology depending ambiguities. The signals corresponding to the following domains of the signal are mandatory, if the sensor technology has the specific ambiguity: ambiguity corresponding to signal "Radial velocity ambiguity domain {begin, end}" (A.1.16), signal "Range ambiguity domain {begin, end}" (A.1.17), signal "Angle azimuth ambiguity domain {begin, end}" (A.1.18) and/or signal "Angle elevation ambiguity domain {begin, end}" (A.1.19).

9.4 Lidar detection interface

Table 33 defines the logical structure of the lidar detection interface (LDI).

Table 33 — Lidar detection interface

LSG	RL LSG M/C/O	Signal	RL Signal M/C/O	Option
Information: interface	M	Interface version ID {major, minor, patch} (A.1.1)	M	Profile: Uniqueness of interface versioning (6.4)
		Number of valid serving sensors (A.1.2)	M	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
		Size type: dynamic/fixed Size #: Number of valid serving sensors (A.1.2)		

Table 33 *(continued)*

LSG	RL LSG M/C/O	Signal	RL Signal M/C/O	Option
		Sensor ID (A.1.3)	M	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
		Interface ID (A.1.4)	O	Profile: Uniqueness of interface versioning (6.4)
		Time stamp – measurement (A.1.5.2)	M	
		Cycle counter (A.1.6.1)	O	Redundancy (B.1.2) Signal: Time stamp – measurement (A.1.5.2)
		Interface cycle time (A.1.7)	O	
		Interface cycle time – variation (A.1.8)	O	
		Data qualifier (A.1.9)	M	
Detections	M	Recognised detections – capability (A.1.10.8)	O	
		Recognised detections – status (A.1.11.8)	O	Redundancy (B.1.2) Signal: Recognised detections – capability (A.1.10.8)
		Number of valid detections (A.1.12.8)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Position {radial distance, azimuth, elevation} (A.4.16) identifiers {azimuth, elevation} Optimise (B.1.4) Alternative implementation
		Size type: dynamic/fixed Size #: Number of valid detections (A.1.12.8)		
↪ Status	M	Existence probability – detection level (A.4.1)	M	
		Object ID reference – detection level (A.4.2)	C (B.3.1) Exist: OLI	Alternative (B.3.2)
		Time stamp difference – detection level (A.4.4)	M	Optimise (B.1.5)

Table 33 *(continued)*

LSG	RL LSG M/C/O	Signal	RL Signal M/C/O	Option
↪ Information	M	Reflectivity (A.4.22)	M	
		Reflectivity – error (A.4.23)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Free space probability (A.4.12)	O	
		Number of valid detection classifications (A.4.13)	O	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Detection classification type (A.4.14)
		Size type: dynamic/fixed Size #: Number of valid detection classifications (A.4.13)		
		Detection classification type (A.4.14)	C (B.2.2) Signal: Number of valid detection classifications (A.4.13)	Alternative (B.1.3)
		Detection classification type – confidence (A.4.15)	C (B.2.2) Signal: Number of valid detection classifications (A.4.13)	
↪ Position	M	Position {radial distance, azimuth, elevation} (A.4.16)	M	Alternative (B.1.3) Signal: {azimuth, elevation}
		Position {radial distance, azimuth, elevation} – error (A.4.17)	M	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Height – lidar (A.4.20)	O	
		Height – lidar – error (A.4.21)	C (B.2.2) Signal: Height – lidar (A.4.20)	Implementation (B.4.1) Error model implementation Alternative (B.1.3)

Table 33 *(continued)*

LSG	RL LSG M/C/O	Signal	RL Signal M/C/O	Option
↪ Dynamics	O	Relative velocity {radial distance} (A.4.18)	M	
		Relative velocity {radial distance} – error (A.4.19)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)

9.4.1 Lidar detection header

Table 34 defines the interface header in subclause "Lidar detection header" (9.4.1) and the changes due to the adaptation in comparison to the generic interface header in subclause "Generic sensor detections header" (9.2.1). The header of the LDI can contain a list of valid lidar detection entities [see subclause "Lidar detection entity" (9.4.2)].

Table 34 — Specific signal grouping: Lidar detection header

Signal	RL signal	Option
Interface version ID {major, minor, patch} (A.1.1)	Mandatory	Profile: Uniqueness of interface versioning (6.4)
Number of valid serving sensors (A.1.2)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
↪ Sensor ID (A.1.3)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
Interface ID (A.1.4)	Optional	Profile: Uniqueness of interface versioning (6.4)
Time stamp – measurement (A.1.5.2)	Mandatory	
Cycle counter (A.1.6.1)	Optional	Redundancy (B.1.2) Signal: Time stamp – measurement (A.1.5.2)
Interface cycle time (A.1.7)	Optional	
Interface cycle time – variation (A.1.8)	Optional	
Data qualifier (A.1.9)	Mandatory	
Recognised detections – capability (A.1.10.8)	Optional	
Recognised detections – status (A.1.11.8)	Optional	Redundancy (B.1.2) Signal: Recognised detections – capability (A.1.10.8)
Number of valid detections (A.1.12.8)	Mandatory	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Position {radial distance, azimuth, elevation} (A.4.16) identifiers {azimuth, elevation} Optimise (B.1.4) Alternative implementation

The following LSGs, which normally shall be provided by the sensor in another interface, may be added to this header, if, for example, the other interface is not implemented by the sensor [see subclause "Need of logical signal group" (B.3.3)]:

— interface: SPI; LSG "Information: vehicle coordinate system" (see Table 46);

— interface: SPI; LSG "Information: sensor pose" (see Table 46);

— interface: SHII; LSG "Calibration" (see Table 48);

— interface: SHII; LSG "Sensor cluster" (see Table 48).

9.4.2 Lidar detection entity

Each detection of a lidar sensor describes a measured detection and consists of several LSGs.

— **Status:** the status describes general information of the detection. It is based on current information and does not include historical information. Table 35 defines the signal grouping "Lidar detection entity status". It defines this status LSG of the detection and redefines the signal grouping "Generic sensor detections entity status" (see Table 29). There is no lidar feature interface defined and therefore lidar detections cannot reference lidar features.

— **Information:** the information describes the lidar specific information of the detection.

— **Position:** the position describes the position information of the detection in the sensor coordinate system of the detection.

— **Dynamics:** the dynamics describes the dynamic information of the detection in the sensor coordinate system of the detection.

Table 35 — Specific signal grouping: Lidar detection entity status

Signal	RL signal	Option
Existence probability – detection level (A.4.1)	Mandatory	
Object ID reference – detection level (A.4.2)	Conditional (B.3.1) Exist: OLI	Alternative (B.3.2)
Time stamp difference – detection level (A.4.4)	Mandatory	Optimise (B.1.5)

9.5 Camera detection interface

Table 36 defines the logical structure of the camera detection interface (CDI). Camera detections use shapes to define detections.

Table 36 — Camera detection interface

LSG	RL LSG M/C/O	Signal	RL Signal M/C/O	Option
Information: interface	M	Interface version ID {major, minor, patch} (A.1.1)	M	Profile: Uniqueness of interface versioning (6.4)
		Number of valid serving sensors (A.1.2)	M	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
		Size type: dynamic/fixed Size #: Number of valid serving sensors (A.1.2)		
		Sensor ID (A.1.3)	M	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
		Interface ID (A.1.4)	O	Profile: Uniqueness of interface versioning (6.4)
		Time stamp – measurement (A.1.5.2)	M	
		Cycle counter (A.1.6.1)	O	Redundancy (B.1.2) Signal: Time stamp – measurement (A.1.5.2)
		Interface cycle time (A.1.7)	O	
		Interface cycle time – variation (A.1.8)	O	
		Data qualifier (A.1.9)	M	
		Colour model type (A.1.15)	M	Profile: Colour model for CDI (9.5.3) Optimise (B.1.3) Use enumeration to define colours by defining colour values and the applied colour model for each enumerator.

Table 36 *(continued)*

LSG	RL LSG M/C/O	Signal	RL Signal M/C/O	Option
Shapes	M	Recognised shapes – capability (A.1.10.9)	O	
		Recognised shapes – status (A.1.11.9)	O	Redundancy (B.1.2) Signal: Recognised shapes – capability (A.1.10.9)
		Number of valid shapes (A.1.12.9)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid shapes (A.1.12.9)		
↪ Status	M	Existence probability – detection level (A.4.1)	M	
		Object ID reference – detection level (A.4.2)	C (B.3.1) Exist: OLI	Alternative (B.3.2)
		Feature ID reference (A.4.3)	C (B.3.1) Exist: FLI	Alternative (B.3.2)
		Time stamp difference – detection level (A.4.4)	M	Optimise (B.1.5)
↪ Shape information	M	Free space probability (A.4.12)	O	
		Number of valid shape classifications – detection level (A.4.24)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Shape classification type – detection level (A.4.25)
		Size type: dynamic/fixed Size #: Number of valid shape classifications – detection level (A.4.24)		
		Shape classification type – detection level (A.4.25)	M	Alternative (B.1.3)
		Shape classification type – confidence – detection level (A.4.26)	M	
		Shape ambiguity grouping ID (A.4.27)	C (B.2.1) ambiguity	
↪ Shape colour tone	M	Size type: dynamic/fixed Size # * Implicit list length – Optimise (B.1.1): List length optimisation: depends on Colour model type (A.1.15)		
		Colour value – detection level (A.4.28)	M	Profile: Colour model for CDI (9.5.3)
		Colour tone – confidence – detection level (A.4.29)	O	Profile: Colour model for CDI (9.5.3)

Table 36 *(continued)*

LSG	RL LSG M/C/O	Signal	RL Signal M/C/O	Option
↪ Shape points	M	Shape type – detection level (A.4.30)	M	
		Number of valid shape points – detection level (A.4.31)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid shape points – detection level (A.4.31)		
		Point existence probability – detection level (A.4.32)	M	
		Position {radial distance, azimuth, elevation} (A.4.16)	{O, M, M}	
		Position {radial distance, azimuth, elevation} – error (A.4.17)	{C (B.2.2) * Signal: Position {radial distance} (A.4.16), M, M}	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
↪ Shape reference points	O	Number of valid shape reference points – detection level (A.4.33)	M	Optimise (B.1.1) List length optimisation
		Size type: dynamic/fixed Size #: Number of valid shape reference points – detection level (A.4.33)		
		Point existence probability – detection level (A.4.32)	M	
		Position {radial distance, azimuth, elevation} (A.4.16)	{O, M, M}	
		Position {radial distance, azimuth, elevation} – error (A.4.17)	{C (B.2.2) * Signal: Position {radial distance} (A.4.16), M, M}	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Translation rate {x, y, z} – detection level (A.4.34)	O	
		Translation rate {x, y, z} – detection level – error (A.4.35)	O	

9.5.1 Camera detection header

Table 37 defines the interface header in subclause “Camera detection header” (9.5.1) and the changes due to the adaptation in comparison to the generic interface header in subclause “Generic sensor detections header” (9.2.1). The header of the CDI can contain a list of valid camera detection entities [see subclause “Camera detection entity” (9.5.2)].

Table 37 — Specific signal grouping: Camera detection header

Signal	RL signal	Option
Interface version ID {major, minor, patch} (A.1.1)	Mandatory	Profile: Uniqueness of interface versioning (6.4)

Table 37 *(continued)*

Signal	RL signal	Option
Number of valid serving sensors (A.1.2)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
↪ Sensor ID (A.1.3)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
Interface ID (A.1.4)	Optional	Profile: Uniqueness of interface versioning (6.4)
Time stamp – measurement (A.1.5.2)	Mandatory	
Cycle counter (A.1.6.1)	Optional	Redundancy (B.1.2) Signal: Time stamp – measurement (A.1.5.2)
Interface cycle time (A.1.7)	Optional	
Interface cycle time – variation (A.1.8)	Optional	
Data qualifier (A.1.9)	Mandatory	
Colour model type (A.1.15)	Mandatory	Profile: Colour model for CDI (9.5.3) Optimise (B.1.3) Use enumeration to define colours by defining colour values and the applied colour model for each enumerator.
Recognised shapes – capability (A.1.10.9)	Optional	
Recognised shapes – status (A.1.11.9)	Optional	Redundancy (B.1.2) Signal: Recognised shapes – capability (A.1.10.9)
Number of valid shapes (A.1.12.9)	Mandatory	Optimise (B.1.1) List length optimisation

The following LSGs, which normally shall be provided by the sensor in another interface, may be added to this header, if, for example, the other interface is not implemented by the sensor [see subclause “Need of logical signal group” (B.3.3)]:

— interface: SPI; LSG “Information: vehicle coordinate system” (see Table 46);

— interface: SPI; LSG “Information: sensor pose” (see Table 46);

— interface: SHII; LSG “Calibration” (see Table 48);

— interface: SHII; LSG “Sensor cluster” (see Table 48).

9.5.2 Camera detection entity

Each detection of a camera sensor describes a measured detection and consists of several LSGs.

— **Status:** the status describes general information of the detection. It is based on current information and does not include historical information. Table 38 defines the signal grouping “Camera detection entity status”. It defines this status LSG of the detection and redefines the signal grouping “Generic sensor detections entity status” (see Table 29).

— **Shape information:** the information describes the camera specific information of the detection.

— **Shape colour tone:** the colour tone describes visual properties of the shape.

— **Shape points:** the shape points describe the geometric vertices of the hull for the detection in the sensor coordinate system.

— **Shape reference points:** the shape reference points are small distinctive segments which are part of the detection's shape.

Table 38 — Specific signal grouping: Camera detection entity status

Signal	RL signal	Option
Existence probability – detection level (A.4.1)	Mandatory	
Object ID reference – detection level (A.4.2)	Conditional (B.3.1) Exist: OLI	Alternative (B.3.2)
Feature ID reference (A.4.3)	Conditional (B.3.1) Exist: FLI	Alternative (B.3.2)
Time stamp difference – detection level (A.4.4)	Mandatory	Optimise (B.1.5)

9.5.3 Profile: Colour model for CDI

The signal “Colour model type” (A.1.15) defines the colour model for the interface. Depending on the colour model, each colour is described by a fixed number of colour values [see signal “Colour value – detection level” (A.4.28)] to define the colour tone. The confidence of the colour tone is provided by the signal “Colour tone – confidence – detection level” (A.4.29).

The interface provides colour tones for detections.

9.6 Ultrasonic detection interface

Table 39 defines the logical structure of the ultrasonic detection interface (UDI).

Table 39 — Ultrasonic detection interface

LSG	RL LSG M/C/O	Signal	RL Signal M/C/O	Option
Information: interface	M	Interface version ID {major, minor, patch} (A.1.1)	M	Profile: Uniqueness of interface versioning (6.4)
		Number of valid serving sensors (A.1.2)	M	Profile: Uniqueness of interface versioning (6.4) Profile: Ultrasonic sensor cluster (9.6.3) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
		Size type: dynamic/fixed Size #: Number of valid serving sensors (A.1.2)		
		Sensor ID (A.1.3)	M	Profile: Uniqueness of interface versioning (6.4) Profile: Ultrasonic sensor cluster (9.6.3) Alternative (B.1.3)
		Interface ID (A.1.4)	O	Profile: Uniqueness of interface versioning (6.4)
		Time stamp – measurement (A.1.5.2)	M	
		Cycle counter (A.1.6.1)	O	Redundancy (B.1.2) Signal: Time stamp – measurement (A.1.5.2)
		Interface cycle time (A.1.7)	O	
		Interface cycle time – variation (A.1.8)	O	
		Data qualifier (A.1.9)	M	

Table 39 *(continued)*

LSG	RL LSG M/C/O	Signal	RL Signal M/C/O	Option
Detections	M	Recognised detections – capability (A.1.10.8)	O	
		Recognised detections – status (A.1.11.8)	O	Redundancy (B.1.2) Signal: Recognised detections – capability (A.1.10.8)
		Number of valid detections (A.1.12.8)	M	Optimise (B.1.1) List length optimisation Optimise (B.1.4) Alternative implementation
		Size type: dynamic/fixed Size #: Number of valid detections (A.1.12.8)		
↪ Status	M	Existence probability – detection level (A.4.1)	M	
		Object ID reference – detection level (A.4.2)	C (B.3.1) Exist: OLI	Alternative (B.3.2)
		Feature ID reference (A.4.3)	C (B.3.1) Exist: FLI	Alternative (B.3.2)
		Time stamp difference – detection level (A.4.4)	M	Profile: Ultrasonic sensor cluster (9.6.3) Optimise (B.1.5)
↪ Information	M	Second sensor ID reference (A.4.36)	C Profile: Ultrasonic sensor cluster (9.6.3)	Profile: Ultrasonic sensor cluster (9.6.3)
		Reflectivity (A.4.22)	O	
↪ Position	M	Distance (A.4.37)	M	
		Distance – error (A.4.38)	M	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Height – ultrasonic (A.4.39)	C (B.2.1) technology specific	
		Height – ultrasonic – error (A.4.40)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)

9.6.1 Ultrasonic detection header

Table 40 defines the interface header in subclause “Ultrasonic detection header” (9.6.1) and the changes due to the adaptation in comparison to the generic interface header in subclause “Generic sensor detections header” (9.2.1). The header of the UDI can contain a list of valid ultrasonic detection entities [see subclause “Ultrasonic detection entity” (9.6.2)].

Table 40 — Specific signal grouping: Ultrasonic detection header

Signal	RL signal	Option
Interface version ID {major, minor, patch} (A.1.1)	Mandatory	Profile: Uniqueness of interface versioning (6.4)
Number of valid serving sensors (A.1.2)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Profile: Ultrasonic sensor cluster (9.6.3) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
↪ Sensor ID (A.1.3)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Profile: Ultrasonic sensor cluster (9.6.3) Alternative (B.1.3)
Interface ID (A.1.4)	Optional	Profile: Uniqueness of interface versioning (6.4)
Time stamp – measurement (A.1.5.2)	Mandatory	
Cycle counter (A.1.6.1)	Optional	Redundancy (B.1.2) Signal: Time stamp – measurement (A.1.5.2)
Interface cycle time (A.1.7)	Optional	
Interface cycle time – variation (A.1.8)	Optional	
Data qualifier (A.1.9)	Mandatory	
Recognised detections – capability (A.1.10.8)	Optional	
Recognised detections – status (A.1.11.8)	Optional	Redundancy (B.1.2) Signal: Recognised detections – capability (A.1.10.8)
Number of valid detections (A.1.12.8)	Mandatory	Optimise (B.1.1) List length optimisation Optimise (B.1.4) Alternative implementation

The UDI may provide all detections of either the emitting or the sensing ultrasonic sensor. Each detection will provide the corresponding sensor by the signal “Second sensor ID reference” (A.4.36). If the UDI provides detections for a sensor-sub-cluster, the first signal “Sensor ID” (A.1.3) provides the emitting sensor and the following signal “Sensor ID” (A.1.3) provides the sensing sensor. In case of a concurrent emitting and sensing sensor element the signal “Sensor ID” (A.1.3) provides only one ID for both elements [see subclause “Profile: Ultrasonic sensor cluster” (9.6.3)].

The following LSGs, which normally shall be provided by the sensor in another interface, may be added to this header, if, for example, the other interface is not implemented by the sensor [see subclause “Need of logical signal group” (B.3.3)]:

— interface: SPI; LSG “Information: vehicle coordinate system” (see Table 46);

— interface: SPI; LSG “Information: sensor pose” (see Table 46);

— interface: SHII; LSG “Calibration” (see Table 48);

— interface: SHII; LSG “Sensor cluster” (see Table 48).

9.6.2 Ultrasonic detection entity

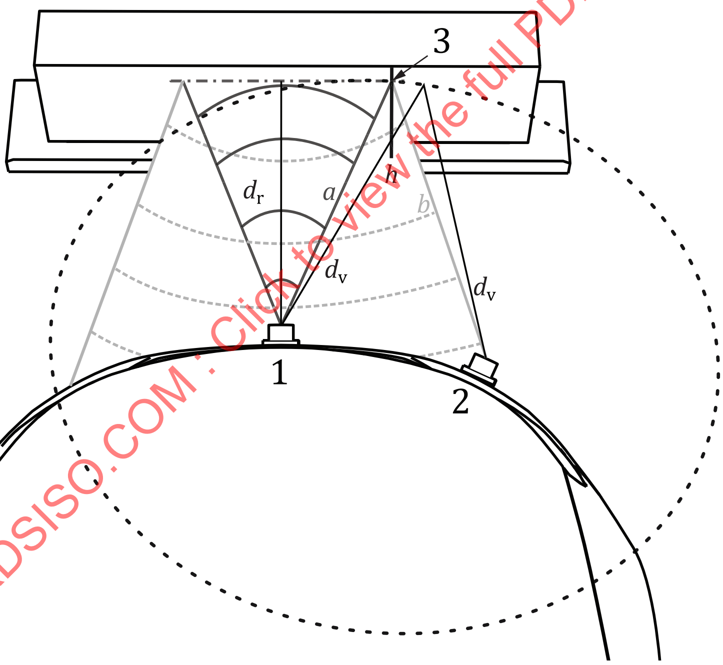
Each detection of an ultrasonic sensor describes a measured detection and consists of several LSGs.

— **Status:** the status describes general information of the detection. It is based on current information and does not include historical information. Table 41 defines the signal grouping “Ultrasonic detection entity status”. It defines this status LSG of the detection and redefines the signal grouping “Generic sensor detections entity status” (see Table 29).

— **Information:** The information describes the ultrasonic specific information of the echo (see Figure 14).

Table 41 — Specific signal grouping: Ultrasonic detection entity status

Signal	RL signal	Option
Existence probability – detection level (A.4.1)	Mandatory	
Object ID reference – detection level (A.4.2)	Conditional (B.3.1) Exist: OLI	Alternative (B.3.2)
Feature ID reference (A.4.3)	Conditional (B.3.1) Exist: FLI	Alternative (B.3.2)
Time stamp difference – detection level (A.4.4)	Mandatory	Optimise (B.1.5)



Key

1 ultrasonic sender

2 ultrasonic receiver

3 obstacle’s reflection point

a distance from the emitting ultrasonic element to the obstacle’s reflection point

b distance from the obstacle’s reflection point to the sensing ultrasonic element

d_v mean distance := $(a+b)/2$ of the echo

d_r distance of the direct reflection and the ultrasonic sensor (S)

h height of the obstacle

Figure 14 — Example for an ultrasonic sensor cluster with separate emitting and sensing elements

9.6.3 Profile: Ultrasonic sensor cluster

An ultrasonic sensor cluster (see Figure 15) may serve this interface in different ways (see Table 42):

1. Each virtual sensor (which is based on two real sensors, one for emmitting and one for sensing) and real sensor provide a separate interface. The emitting and sensing sensors are defined in the header of the interface message. Real sensors, which emit and sense their own echo, provide a DLI with the signal “Sensor ID” (A.1.3) of the sensor (one ID). Virtual sensors provide a DLI with the first signal “Sensor ID” (A.1.3) of the emitting sensor and the second of the sensing sensor (two IDs). The RL of the signal “Second sensor ID reference” (A.4.36) is optional and the signal would be redundant.

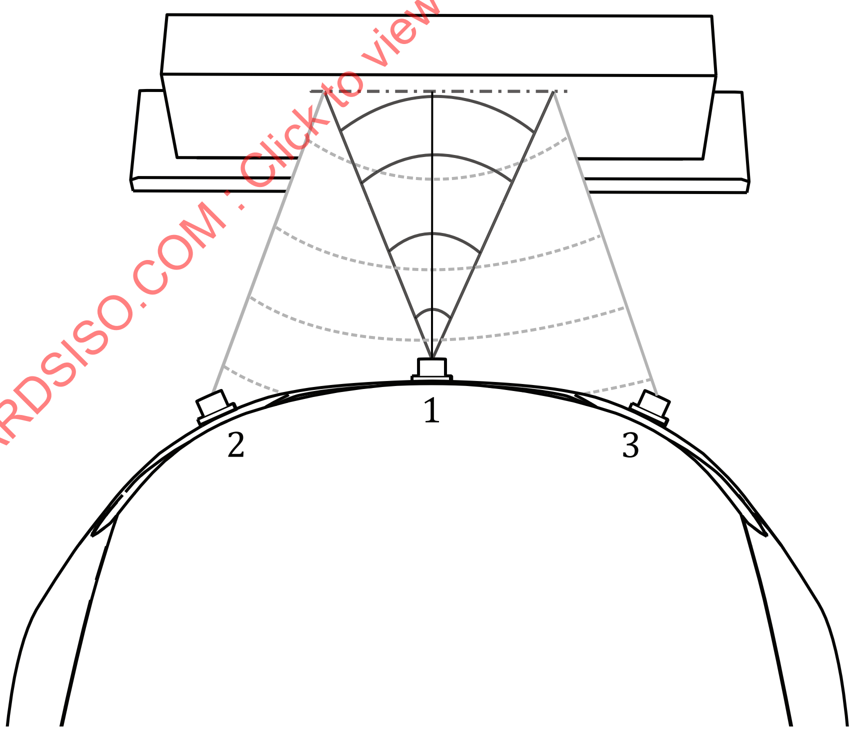
2. Each real sensor provides a separate interface, including all detections of the virtual sensors. Virtual sensors may serve no interface.

   a. The signal “Sensor ID” (A.1.3) provides the ID of the emitting sensor. The sensing sensor of the detection is referenced by the signal “Second sensor ID reference” (A.4.36), which is mandatory for a sensor cluster. The measurement cycle of the sensor cluster’s UDIs may not overlap in time.

   b. The signal “Sensor ID” (A.1.3) provides the ID of the sensing sensor. The emitting sensor of the detection is referenced by the signal “Second sensor ID reference” (A.4.36), which is mandatory for a sensor cluster. The measurement cycle of the sensor cluster’s UDIs may overlap in time.

The signal “Second sensor ID reference” (A.4.36) is optional for a single ultrasonic sensor.

All emitting and/or sensing ultrasonic sensors may provide the SPI sensor origin point, in detail the LSG “Information: sensor pose” (see Table 46).



Key

1 ultrasonic sender and receiver

2 ultrasonic receiver 1

3 ultrasonic receiver 2

Figure 15 — Example for an ultrasonic sensor cluster of 3 ultrasonic elements for one echo

Table 42 — Example of UDIs for 3 emitting and sensing elements; one echo – Profile relevant signals

Profile: Ultrasonic sensor cluster (9.6.3) – option 1 (9 UDI)	Profile: Ultrasonic sensor cluster (9.6.3) – option 2a (3 UDI)	Profile: Ultrasonic sensor cluster (9.6.3) – option 2b (3 UDI)

UDI {
{Sensor ID = SR}
Detection {}
}
	UDI {
	{Sensor ID = SR}
	Detection {
	Second sensor ID reference = SR
	Time stamp difference = dt_{SRSR}
	}
	Detection {
	Second sensor ID reference = R1
	Time stamp difference = dt_{SRR1}
	}
	Detection {
	Second sensor ID reference = R2
	Time stamp difference = dt_{SRR2}
	}
	}
		UDI {
		{Sensor ID = SR}
		...
		Detection {
		Second sensor ID reference = SR
		Time stamp difference = $dt_{SR}+ dt_{SRSR}$
		}
		...
		}

UDI {
{Sensor ID = SR,
Sensor ID = R1}
Detection {}
}
		UDI {
		{Sensor ID = R1}
		...
		Detection {
		Second sensor ID reference = SR
		Time stamp difference = $dt_{SR}+ dt_{SRR1}$
		}
		...
		}

UDI {
{Sensor ID = SR,
Sensor ID = R2}
Detection {}
}
	...	UDI {
		{Sensor ID = R2}
		...
		Detection {
		Second sensor ID reference = SR
		Time stamp difference = $dt_{SR}+ dt_{SRR2}$
		}
		...
		}

...		

10 Supportive sensor interfaces

10.1 General

Sensors and sensor clusters serve supportive sensor interfaces (SSI). A fusion unit may derive the sensor cluster's SSI's information of the sensor cluster's sensor's SSIs. Therefore, in this clause for SSI, the term sensor is always used, even if a sensor cluster is serving the interface. The SSIs provide additional, general information of a sensor's or sensor cluster's health and performance. Table 43 provides an overview over the differences and boundaries between the SSIs.

Table 43 — Brief overview over the differences and boundaries between the SSIs

Sensor performance		Sensor health information
Impairment on observed field	Impairment on sensor surface	Sensor internal
— rain, fog — snow — particles (air) — and so forth	— dirt, dust — condensation — scratch — and so forth	— operation — diagnosis — defects — cleaning — position calibration — and so forth
Classification into several measurement ranges Relevant for safety concept		Global information for the complete sensor Misalignment: hardware and software calibration Relevant E/E information

The following supportive sensor interfaces are available:

— sensor performance interface (10.3);

— sensor health information interface (10.4).

Supportive sensor interface messages [see subclause "Sensor performance interface" (10.3), "Sensor health information interface" (10.4)] may be required to correctly interpret information from object-, feature- and detection level. Data consistency over the sensor's interfaces shall be assured. Supportive sensor interfaces have LSGs which the sensor shall provide. If an SSI is not provided, the SSI relevant LSG will be provided by other interfaces as defined [see subclause "Need of logical signal group" (B.3.3)].

10.2 Generic supportive sensor interface

Table 44 defines the generic structure of an SSI.

Table 44 — Generic supportive sensor interface

Structure	Multiplicity	Option
Generic supportive sensor header (10.2.1)	1	
Generic supportive sensor entity (10.2.2), specifically an individual supportive sensor entity (list)	SPI: multiple as well as SHII: 1	Size type: dynamic/fixed

The multiplicity of the individual supportive sensor interface entity list depends on the individual SSI interface.

The geometric information of the SSIs references the vehicle coordinate system [see signal "Vehicle coordinate system type" (A.1.21)] as well as the sensor coordinate system.

10.2.1 Generic supportive sensor header

Table 45 defines the interface header in subclause "Generic supportive sensor header" (10.2.1) and the changes due to the adaptation in comparison to the generic interface header in subclause "Generic interface header" (6.2). Each individual SSI may define the signal "Time stamp – <…>" (A.1.5) during the system design phase.

Table 45 — Specific signal grouping: Generic supportive sensor header

Signal	RL signal	Option
Interface version ID {major, minor, patch} (A.1.1)	Mandatory	Profile: Uniqueness of interface versioning (6.4)
Number of valid serving sensors (A.1.2)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
↪ Sensor ID (A.1.3)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
Interface ID (A.1.4)	Optional	Profile: Uniqueness of interface versioning (6.4)
Time stamp – <...> (A.1.5)	Mandatory	
Message counter (A.1.6.2)	Optional	
Interface cycle time (A.1.7)	Optional	
Interface cycle time – variation (A.1.8)	Optional	
Data qualifier (A.1.9)	Mandatory	
SSI specific header extensions		
Number of valid <...> (A.1.12)	Mandatory	Optimise (B.1.1) List length optimisation

The LSGs, which normally shall be provided by the sensor as well as the sensor cluster in another interface, may be added to this header, if, for example, the other interface is not implemented by the sensor or sensor cluster [see subclause “Need of logical signal group” (B.3.3)]. For each SSI these LSGs are individual.

10.2.2 Generic supportive sensor entity

No generic LSG is defined for the interface entity types of all SSI.

10.3 Sensor performance interface

Table 46 defines the logical structure of the sensor performance interface (SPI).

Table 46 — Sensor performance interface

LSG	RL LSG M/C/O	Signal	RL Signal M/C/O	Option
Information: interface	M	Interface version ID {major, minor, patch} (A.1.1)	M	Profile: Uniqueness of interface versioning of SPIs (10.3.3)
		Number of valid serving sensors (A.1.2)	M	Profile: Uniqueness of interface versioning of SPIs (10.3.3) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
		Size type: dynamic/fixed Size #: Number of valid serving sensors (A.1.2)		
		Sensor ID (A.1.3)	M	Profile: Uniqueness of interface versioning of SPIs (10.3.3) Alternative (B.1.3)
		Interface ID (A.1.4)	O	Profile: Uniqueness of interface versioning of SPIs (10.3.3)
		Time stamp – <...> (A.1.5)	M	
		Message counter (A.1.6.2)	O	
		Interface cycle time (A.1.7)	O	
		Interface cycle time – variation (A.1.8)	O	
		Data qualifier (A.1.9)	M	
		Interface applicability (A.1.20)	O	Profile: Uniqueness of interface versioning of SPIs (10.3.3)
Information: vehicle coordinate system	M Need of logical signal group (B.3.3) 1. SHII Header, 2. Redundant on OLI, FLI and DLI Header	Vehicle coordinate system type (A.1.21)	M	Optimise Implicit values (B.1.5)

Table 46 *(continued)*

LSG	RL LSG M/C/O	Signal	RL Signal M/C/O	Option
Information: sensor pose	M Need of logical signal group (B.3.3) 1. SHII Header, 2. DLI Header	Sensor origin point {x, y, z} (A.1.22)	M	
		Sensor origin point {x, y, z} – error (A.1.23)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Sensor orientation {yaw, pitch, roll} (A.1.24)	M	
		Sensor orientation {yaw, pitch, roll} – error (A.1.25)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
Information: sensor surrounding	O	Vanishing point {azimuth, elevation} (A.1.26)	M	
		Vanishing point {azimuth, elevation} – error (A.1.27)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
Segments	M	Number of valid field-of-view segments (A.1.12.10)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Segment azimuth {begin, end} (A.5.1) and Segment elevation {begin, end} (A.5.2)
		Size type: Dynamic/fixed Size #: Number of valid field-of-view segments (A.1.12.10)		
↪ Status	M	Segment azimuth {begin, end} (A.5.1)	M	Alternative (B.1.3)
		Segment elevation {begin, end} (A.5.2)	M	Alternative (B.1.3)
		Measurement grid resolution {radial distance, azimuth, elevation} (A.5.3)	O	
		Beam divergence {azimuth, elevation} (A.5.4)	O	
		Range gain (A.5.5)	O	
		Blockage status (A.5.6)	M	

Table 46 *(continued)*

LSG	RL LSG M/C/O		Signal		RL Signal M/C/O	Option
↪ Field-of-view reduction	O		Number of valid field-of-view reduction reasons (A.5.7)		M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Field-of-view reduction reason type (A.5.8)
			Size type: Dynamic/fixed Size #: Number of valid field-of-view reduction reasons (A.5.7)			
				Field-of-view reduction reason type (A.5.8)	M	Alternative (B.1.3)
				Field-of-view reduction reason type – confidence (A.5.9)	M	
↪ Real-world object recognition capabilities	C (B.3.1) Exist: OLI and in scope of signal “Interface applicability” (A.1.20)		Number of valid recognisable object types (A.5.10)		M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Recognised object type (A.5.11)
			Size type: Dynamic/fixed Size #: Number of valid recognisable object types (A.5.10)			
				Recognised object type (A.5.11)	M	Alternative (B.1.3)
				Detection range radial distance {begin, end} (A.5.12)	M	
				True positive rate (A.5.13)	O	
				False positive rate (A.5.14)	O	
				Positive predictive value (A.5.15)	O	

Table 46 *(continued)*

LSG	RL LSG M/C/O	Signal	RL Signal M/C/O	Option
↪ Reference target recognition capabilities	C (B.3.1) Exist: FLI, DLI and in scope of signal “Interface applicability” (A.1.20)	Number of valid reference target types (A.5.16)	M	Optimise (B.1.1) List length optimisation
		Size type: Dynamic/fixed Size #: Number of valid reference target types (A.5.16)		
		Reference target type (A.5.17)	C (B.2.1) Relevant: camera	
		Radar cross section reference target (A.5.18)	C (B.2.1) Relevant: radar	
		Reflectivity reference target (A.5.19)	C (B.2.1) Relevant: lidar, ultrasonic	
		Detection range radial distance {begin, end} (A.5.12)	M	
		True positive rate (A.5.13)	O	
		Relative radial velocity range {begin, end} (A.5.20)	C (B.2.1) Relevant: radar	
		Signal to noise ratio – supportive level (A.5.21)	M	
		Spatial separability {radial distance, azimuth, elevation} (A.5.22)	O	
		Velocity separability {radial distance, azimuth, elevation} (A.5.23)	O	

The signals referring to angles as well as distances in the segments always refer to the sensor coordinate system and its origin. This affects the signals:

— segment azimuth {begin, end} (A.5.1);

— segment elevation {begin, end} (A.5.2);

— measurement grid resolution {radial distance, azimuth, elevation} (A.5.3);

— detection range radial distance {begin, end} (A.5.12);

— relative radial velocity range {begin, end} (A.5.20);

— spatial separability {radial distance, azimuth, elevation} (A.5.22);

— velocity separability {radial distance, azimuth, elevation} (A.5.23).

The correlation between vehicle- and sensor coordinate system is determined by the sensor origin point of the sensor [LSG “Information: sensor pose” (see Table 46)].

10.3.1 Sensor performance header

Table 47 defines the interface header in subclause “Sensor performance header” (10.3.1) and the changes due to the adaptation in comparison to the generic interface header in subclause “Generic supportive sensor header" (10.2.1). The header of the SPI can contain a list of valid segment entities [see subclause "Sensor performance entity" (10.3.2)].

Table 47 — Specific signal grouping: Sensor performance header

<table>
<tr><th>Signal</th><th colspan="2">RL signal</th><th>Option</th></tr>
<tr><td>Interface version ID {major, minor, patch} (A.1.1)</td><td colspan="2">Mandatory</td><td>Profile: Uniqueness of interface versioning of SPIs (10.3.3)</td></tr>
<tr><td>Number of valid serving sensors (A.1.2)</td><td colspan="2">Mandatory</td><td>Profile: Uniqueness of interface versioning of SPIs (10.3.3)
Optimise (B.1.1)
List length optimisation
Alternative Unrolling tuple-lists (B.1.6)
Key: Sensor ID (A.1.3)</td></tr>
<tr><td>↪ Sensor ID (A.1.3)</td><td colspan="2">Mandatory</td><td>Profile: Uniqueness of interface versioning of SPIs (10.3.3)
Alternative (B.1.3)</td></tr>
<tr><td>Interface ID (A.1.4)</td><td colspan="2">Optional</td><td>Profile: Uniqueness of interface versioning of SPIs (10.3.3)</td></tr>
<tr><td>Time stamp – <...> (A.1.5)</td><td colspan="2">Mandatory</td><td></td></tr>
<tr><td>Message counter (A.1.6.2)</td><td colspan="2">Optional</td><td></td></tr>
<tr><td>Interface cycle time (A.1.7)</td><td colspan="2">Optional</td><td></td></tr>
<tr><td>Interface cycle time – variation (A.1.8)</td><td colspan="2">Optional</td><td></td></tr>
<tr><td>Data qualifier (A.1.9)</td><td colspan="2">Mandatory</td><td></td></tr>
<tr><td>Interface applicability (A.1.20)</td><td colspan="2">Optional</td><td>Profile: Uniqueness of interface versioning of SPIs (10.3.3)</td></tr>
<tr><td>Vehicle coordinate system type (A.1.21)</td><td>RL LSG mandatory
Need of logical signal group (B.3.3)
1. SHII Header,
2. Redundant on OLI, FLI and DLI Header</td><td>Mandatory</td><td>Optimise Implicit values (B.1.5)</td></tr>
<tr><td>Sensor origin point {x, y, z} (A.1.22)</td><td rowspan="4">RL LSG mandatory
Need of logical signal group (B.3.3)
1. SHII Header,
2. DLI Header</td><td>Mandatory</td><td></td></tr>
<tr><td>Sensor origin point {x, y, z} – error (A.1.23)</td><td>Optional</td><td>Implementation (B.4.1)
Error model implementation
Alternative (B.1.3)</td></tr>
<tr><td>Sensor orientation {yaw, pitch, roll} (A.1.24)</td><td>Mandatory</td><td></td></tr>
<tr><td>Sensor orientation {yaw, pitch, roll} – error (A.1.25)</td><td>Optional</td><td>Implementation (B.4.1)
Error model implementation
Alternative (B.1.3)</td></tr>
<tr><td>Vanishing point {azimuth, elevation} (A.1.26)</td><td rowspan="2">RL LSG optional</td><td>Mandatory</td><td></td></tr>
<tr><td>Vanishing point {azimuth, elevation} – error (A.1.27)</td><td>Optional</td><td>Implementation (B.4.1)
Error model implementation
Alternative (B.1.3)</td></tr>
</table>

Table 47 *(continued)*

Signal	RL signal	Option
Number of valid field-of-view segments (A.1.12.10)	Mandatory	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Segment azimuth {begin, end} (A.5.1) and Segment elevation {begin, end} (A.5.2)

The following LSGs, which normally shall be provided by the sensor as well as the sensor cluster in another interface, may be added to this header, if, for example, the other interface is not implemented by the sensor or sensor cluster [see subclause "Need of logical signal group" (B.3.3)]:

— interface: SHII; LSG "Calibration" (see Table 48);

— interface: SHII; LSG "Sensor cluster" (see Table 48).

The following LSGs, which shall be provided by the sensor as well as the sensor cluster in this interface, may be added to other header [see subclause "Need of logical signal group" (B.3.3)]:

— interface: SPI; LSG "Information: vehicle coordinate system" (see Table 46);

— interface: SPI; LSG "Information: sensor pose" (see Table 46).

10.3.2 Sensor performance entity

Each sensor performance segment of a sensor describes the performance in this segment and consists of several LSGs.

— **Status:** the status describes the geometry of the FOV segment.

— **Field-of-view reduction causes:** the FOV reduction causes describe the causes (for example, extrinsic or intrinsic) for an FOV reduction.

— **Real-world object recognition rates:** the real-world object recognition rates describe the estimated recognition rates for real-world recognition of different real-world objects.

— **Reference target recognition rates:** the reference target recognition rates describe the estimated recognition rates for synthetic, well-defined recognition of different synthetic, well-defined objects.

For two valid but geometric overlapping FOV segments (azimuth-, elevation range), the values for the earlier defined segment will always stay valid. So typically, the segments will be defined from the inside to the outside (due to timing reasons, special regions of interest are more important than the general FOV segments and the special regions of interest need to be known to be subtracted from the general FOV segment).

10.3.3 Profile: Uniqueness of interface versioning of SPIs

This profile extends the profile of subclause "Profile: Uniqueness of interface versioning" (6.4). A sensor cluster may provide this interface more than once for a sensor. To identify the different interface implementations of the SPI the signal "Interface applicability" (A.1.20) is additionally used.

10.4 Sensor health information interface

The subclause "Sensor health information interface" (10.3) (SHII) provides a qualitive statement of the sensor's health status. More detailed information of a sensor statuses can be provided via other interfaces.

Table 48 defines the logical structure of the sensor health information interface.

Table 48 — Sensor health information interface

LSG	RL LSG M/C/O	Signal	RL Signal M/C/O	Option
Information: interface	M	Interface version ID {major, minor, patch} (A.1.1)	M	Profile: Uniqueness of interface versioning (6.4)
		Number of valid serving sensors (A.1.2)	M	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
		Size type: dynamic/fixed Size #: Number of valid serving sensors (A.1.2)		
		Sensor ID (A.1.3)	M	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
		Interface ID (A.1.4)	O	Profile: Uniqueness of interface versioning (6.4)
		Time stamp – <...> (A.1.5)	M	
		Message counter (A.1.6.2)	O	
		Interface cycle time (A.1.7)	O	
		Interface cycle time – variation (A.1.8)	O	
		Data qualifier (A.1.9)	M	

Table 48 *(continued)*

LSG	RL LSG M/C/O	Signal	RL Signal M/C/O	Option
Status	M	Number of valid sensor operation modes (A.5.24)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor operation mode (A.5.25)
		Size type: Dynamic/fixed Size #: Number of valid sensor operation modes (A.5.24)		
		Sensor operation mode (A.5.25)	M	Alternative (B.1.3)
		Sensor defect detected (A.5.26)	M	
		Sensor defect reason (A.5.27)	M	
		Status supply voltage (A.5.28)	M	
		Sensor temperature status (A.5.29)	M	
		Number of valid sensor input signal statuses (A.5.30)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor input signal – type (A.5.31)
		Size type: Dynamic/fixed Size #: Number of valid sensor input signal statuses (A.5.30)		
		Sensor input signal – type (A.5.31)	M	Alternative (B.1.3)
		Sensor input signal – status (A.5.32)	M	
		Sensor externally disturbed (A.5.33)	O	
		Sensor transmit power reduced (A.5.34)	O	
		Status sensor heating (A.5.35)	O	
		Status sensor cleaning (A.5.36)	O	
		Time stamp – <...> (A.1.5)	O	
		Sensor time sync offset value (A.5.38)	C (B.2.2) * The signal "Sensor time sync" (A.5.37) has an enumerator "SensorTimeSync_Offset" or a similar enumerator defined during the system desing phase.	

Table 48 *(continued)*

LSG	RL LSG M/C/O	Signal	RL Signal M/C/O	Option
Calibration	O Need of logical signal group (B.3.3) 1. SPI Header, 2. DLI Header, 3. FLI Header, 4. redundant on OLI Header	Number of valid sensor-calibratable components (A.5.39)	M	Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor-calibratable component (A.5.40)
		Size type: Dynamic/fixed Size #: Number of valid sensor-calibratable components (A.5.39)		
		Sensor-calibratable component (A.5.40)	M	Alternative (B.1.3)
		Sensor calibration status (A.5.41)	M	
		Calibration process state (A.5.42)	O	
		Sensor origin point – correction {x, y, z} (A.5.43)	O	
		Sensor origin point – correction {x, y, z} – error (A.5.44)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Sensor origin translation – correction limit {x_{begin}, x_{end}, y_{begin}, y_{end}, z_{begin}, z_{end}} (A.5.45)	O	
		Sensor orientation – correction {yaw, pitch, roll} (A.5.46)	O	
		Sensor orientation – correction {yaw, pitch, roll} – error (A.5.47)	O	Implementation (B.4.1) Error model implementation Alternative (B.1.3)
		Sensor pose angle – correction limit {yaw_{begin}, yaw_{end}, $pitch_{begin}$, $pitch_{end}$, $roll_{begin}$, $roll_{end}$} (A.5.48)	O	
Sensor cluster	O Need of logical signal group (B.3.3) 1. SPI Header, 2. Redundant on OLI, FLI and DLI Header	Number of valid sensors (A.5.49)	M	
		Size type: Dynamic/fixed Size #: Number of valid sensors (A.5.49)		
		Sensor ID reference (A.5.50)	M	

10.4.1 Sensor health information header

Table 49 defines the interface header in subclause "Sensor health information header" (10.4.1) and the changes due to the adaptation in comparison to the generic interface header in subclause "Generic supportive sensor header" (10.2.1). The header of the SHII contains only one sensor information entity [see subclause "Sensor health information entity" (10.4.2)].

Table 49 — Specific signal grouping: Sensor health information header

Signal	RL signal	Option
Interface version ID {major, minor, patch} (A.1.1)	Mandatory	Profile: Uniqueness of interface versioning (6.4)
Number of valid serving sensors (A.1.2)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Optimise (B.1.1) List length optimisation Alternative Unrolling tuple-lists (B.1.6) Key: Sensor ID (A.1.3)
↪ Sensor ID (A.1.3)	Mandatory	Profile: Uniqueness of interface versioning (6.4) Alternative (B.1.3)
Interface ID (A.1.4)	Optional	Profile: Uniqueness of interface versioning (6.4)
Time stamp – <...> (A.1.5)	Mandatory	
Message counter (A.1.6.2)	Optional	
Interface cycle time (A.1.7)	Optional	
Interface cycle time – variation (A.1.8)	Optional	
Data qualifier (A.1.9)	Mandatory	

There is only one global entity in the SHII. Therefore, the sensor provides no list of SHII entities and a signal "Number of valid <...>" (A.1.12) is additionally not provided.

The following LSGs, which normally shall be provided by the sensor as well as the sensor cluster in another interface, may be added to this header, if, for example, the other interface is not implemented by the sensor or sensor cluster [see subclause "Need of logical signal group" (B.3.3)]:

— interface: SPI; LSG "Information: vehicle coordinate system" (see Table 46);

— interface: SPI; LSG "Information: sensor pose" (see Table 46).

The following LSGs, which shall be provided by the sensor as well as the sensor cluster in this interface, may be added to other header [see subclause "Need of logical signal group" (B.3.3)]:

— interface: SHII; LSG "Calibration" (see Table 48);

— interface: SHII; LSG "Sensor cluster" (see Table 48).

10.4.2 Sensor health information entity

Sensor health information of a sensor describes global sensor statuses and consists of several LSGs. This interface provides only one sensor health information entity for the entire sensor and not a list of entities.

— **Status:** the status describes the global statuses of the sensor.

— **Calibration:** the status describes the statuses and information of the sensor calibration.

— **Sensor cluster:** the sensor cluster defines the sensor group which defines the sensor cluster.

Annex A
(normative)

Interface signals

A.1 Header signals

Interface header specific signals are defined in this subclause (see Tables A.1 - A.62). If different definitions are required for different interfaces, the interface-specific signal is defined in a subclause, for example, the signal “Time stamp – <...>” (A.1.5) and the specific signal “Time stamp – predicted” (A.1.5.1).

A.1.1 Interface version ID {major, minor, patch}

Table A.1 — Signal: Interface version ID {major, minor, patch}

Name	Interface version ID {major, minor, patch}		
Description	The signal “Interface version ID {major, minor, patch}” (A.1.1) provides the version information of the interface. Requirement: The signal “Interface version ID {major, minor, patch}” (A.1.1) is assigned in the system design phase.		
Value type	3D vector value	**Unit**	(1, 1, 1)

A.1.2 Number of valid serving sensors

Table A.2 — Signal: Number of valid serving sensors

Name	Number of valid serving sensors		
Description	The signal “Number of valid serving sensors” (A.1.2) provides the number of valid sensors serving the interface.		
Value type	[0...] integer value	**Unit**	1

A.1.3 Sensor ID

Table A.3 — Signal: Sensor ID

Name	Sensor ID

Table A.3 *(continued)*

Description	The signal "Sensor ID" (A.1.3) which uniquely identifies the sensor and link the data of different sensors' interfaces. Additional information: It is required to associate the logical interfaces of one sensor or a sensor cluster with its sensors. Each signal "Sensor ID" (A.1.3) value shall serve independently at least one interface. The signal "Sensor ID" (A.1.3) is necessary, for example, for a service-oriented architecture. Requirement: The signal's "Sensor ID" (A.1.3) value is assigned during the system design phase.		
Value type	integer value	**Unit**	1

A.1.4 Interface ID

Table A.4 — Signal: Interface ID

Name	Interface ID		
Description	The signal "Interface ID" (A.1.4) is used to uniquely identify the originated interface type of the message. Additional information: The enumerators may differentiate the sensor's sensor technology. The signal "Interface ID" (A.1.4) is necessary, for example, for a service-oriented architecture.		
Value type	enumeration	**Unit**	1

Table A.5 — Enumeration: Interface ID – Example enumerators

Name	**Description**	**RL enumerator**
InterfaceID_PotentiallyMovingObjects	The message was sent by PMOI.	"exemplary"
InterfaceID_RoadObjects	The message was sent by RDOI.	"exemplary"
InterfaceID_StaticObjects	The message was sent by SOI.	"exemplary"
InterfaceID_CameraFeatures	The message was sent by the sensor technology specific FLI. The sensor or sensor cluster uses camera sensing elements.	"exemplary"
InterfaceID_UltrasonicFeatures	The message was sent by the sensor technology specific FLI. The sensor or sensor cluster uses ultrasonic sensing elements.	"exemplary"
InterfaceID_RadarDetections	The message was sent by the sensor technology specific DLI. The sensor or sensor cluster uses radar sensing elements.	"exemplary"
InterfaceID_LidarDetections	The message was sent by the sensor technology specific DLI. The sensor or sensor cluster uses lidar sensing elements.	"exemplary"
InterfaceID_CameraDetections	The message was sent by the sensor technology specific DLI. The sensor or sensor cluster uses camera sensing elements.	"exemplary"
InterfaceID_UltrasonicDetections	The message was sent by the sensor technology specific DLI. The sensor or sensor cluster uses ultrasonic sensing elements.	"exemplary"

Table A.5 *(continued)*

Name	Description	RL enumerator
InterfaceID_SensorPerformance	The message was sent by SPI.	“exemplary”
InterfaceID_SensorHealthInformation	The message was sent by SHII.	“exemplary”

A.1.5 Time stamp – <...>

Time stamp is defined depending on the level of the interface. It is either the most consistent time of the measurement or the predicted time at which the message is consistent. An additional time stamp difference specifies an individual measurement point with a time offset of each entity relative to the time stamp of the message. The time stamps can also be used to link the data of different interfaces of a sensor cluster.

A.1.5.1 Time stamp – predicted

Table A.6 — Signal: Time stamp – predicted

Name	Time stamp – predicted		
Description	The signal “Time stamp – predicted” (A.1.5.1) provides the time stamp at which the entities of the interface are consistent (not the time at which it was processed or at which it is transmitted) in the vehicle-global synchronised time. Clock synchronisation shall be ensured, depending on bus technology. Additional information: The time stamp that was used for the prediction. Specifically, the time stamp can be the same as or differ from the base time stamp(s) of the, for example, underlying detection’s or feature’s entity list(s). Time stamps are consistent for all vehicle electronic control units (ECUs).		
Value type	real value	**Unit**	s

A.1.5.2 Time stamp – measurement

Table A.7 — Signal: Time stamp – measurement

Name	Time stamp – measurement		
Description	The signal “Time stamp – measurement” (A.1.5.2) provides the time stamp at which the measurement was taken (not the time at which it was processed or at which it is transmitted) in the vehicle-global synchronised time. Clock synchronisation shall be ensured, depending on bus technology. Additional information: The signal’s “Time stamp – measurement” (A.1.5.2) value shall be selected so that it is as consistent as possible with the measured entities and the associated time error is minimum. In case of a continuous or a subdivided measurement cycle, the signal “Time stamp difference – detection level” (A.4.4) shall be used to differentiate the real measured point in time of the entity acquisition. Each entity of a detection list may have, for example, the signal “Time stamp difference – detection level” (A.4.4) for its measurement (for example, for integrating sensor technologies). Time stamps are consistent for all vehicle ECUs.		
Value type	real value	**Unit**	s

A.1.6 <...> counter

A continuous value that defines the order of the logical interface's messages. The logical interface messages may be sent cyclically or non-cyclically, for example, event triggered.

A.1.6.1 Cycle counter

Table A.8 — Signal: Cycle counter

Name	Cycle counter		
Description	The signal "Cycle counter" (A.1.6.1) provides the continuous up counter to identify the message cycle of the interface. It counts up per every cycle of a complete prediction/single measurement. Additional information: This information can be determined from signal's "Time stamp – <...>" (A.1.5) value and the sensor's cycle period [see the signal "Interface cycle time" (A.1.7) and "Interface cycle time – variation" (A.1.8)]. Each sensor as well as each sensor cluster uses a local sensor cycle counter per interface.		
Value type	[0...] integer value	**Unit**	1

A.1.6.2 Message counter

Table A.9 — Signal: Message counter

Name	Message counter		
Description	The signal "Message counter" (A.1.6.2) provides the continuous up counter to identify the message sequence of the interface. It counts up per every sent message. Additional information: Interfaces with a signal "Message counter" (A.1.6.2) may send messages non-cyclical, for example, event triggered. Each sensor as well as each sensor cluster uses a local sensor message counter per interface.		
Value type	[0...] integer value	**Unit**	1

A.1.7 Interface cycle time

Table A.10 — Signal: Interface cycle time

Name	Interface cycle time		
Description	The signal "Interface cycle time" (A.1.7) provides the representative cycle time of the interface's messages. Normally, each interface sends the complete interface message cyclically. The period between two messages can vary. Each interface header defines the interface's cycle time and the variation of the cycle time [see signal "Interface cycle time – variation" (A.1.8)]. Additional information: If the signal "Interface cycle time – variation" (A.1.8) is 100 %, the message is sent in the case one or several signals have changed their value.		
Value type	[0...] real value	**Unit**	s

A.1.8 Interface cycle time – variation

Table A.11 — Signal: Interface cycle time – variation

Name	Interface cycle time – variation		
Description	The signal “Interface cycle time – variation” (A.1.8) provides the possible variation of the expected representative cycle time [see signal “Interface cycle time” (A.1.7)] of the interface’s messages. Additional information: If the signal “Interface cycle time – variation” (A.1.8) is 100 %, the message is sent in the case one or several signals have changed their value, for example, specifically for the n.th cycle: 0 s <= n.th cycle time <= infinity s. EXAMPLE The duration between two sequentially sent messages of an interface is in the range [interface cycle time × (1 - interface cycle time variation / 100 %), interface cycle time / (1 - interface cycle time variation/100 %)].		
Value type	[0...] real value	**Unit**	%

A.1.9 Data qualifier

Table A.12 — Signal: Data qualifier

Name	Data qualifier		
Description	The signal “Data qualifier” (A.1.9) expresses whether the content of the entity list(s) of the interface can be used or not. It can change dynamically over time and inform about temporal restrictions.		
Value type	enumeration	**Unit**	1

Table A.13 — Enumeration: Data qualifier – Example enumerators

Name	Description	RL enumerator
DataQualifier_Normal	The entities and information of the interface can be used without restrictions.	“exemplary”
DataQualifier_NotAvailable	The entities and information of the interface are not available.	“exemplary”
DataQualifier_ReduceInCoverage	The entities are not complete due to a restricted coverage (for example, restricted view).	“exemplary”
DataQualifier_ReduceInPerformance	The entities are not complete due to a restricted performance.	“exemplary”
DataQualifier_ReduceInCoverageAndPerformance	The entities are not complete due to a restricted coverage and restricted performance.	“exemplary”
DataQualifier_TestMode	The sensor cluster is running in test mode and outputs differ from normal operation mode.	“exemplary”
DataQualifier_Invalid	The measurement cycle was invalid, no predicted or measured object will be transferred.	“exemplary”

A.1.10 Recognised <...> – capability

The performance of the sensor as well as the sensor cluster determines how many entities the sensor cluster’s interface can recognise, for example, at full load. For example, a value of 0 informs that the sensor may not recognise any object/feature/detection at the moment.

A.1.10.1 Recognised potentially moving objects – capability

Table A.14 — Signal: Recognised potentially moving objects – capability

Name	Recognised potentially moving objects – capability		
Description	The signal "Recognised potentially moving objects – capability" (A.1.10.1) provides the performance of the sensor cluster's interface. It determines how many entities the sensor cluster is guaranteed to recognise, for example, at full load with respect to the current scenario.		
Value type	[0...] integer value	**Unit**	1

A.1.10.2 Recognised road markings – capability

Table A.15 — Signal: Recognised road markings – capability

Name	Recognised road markings – capability		
Description	The signal "Recognised road markings – capability" (A.1.10.2) provides the performance of the sensor cluster's interface. It determines how many entities the sensor cluster is guaranteed to recognise, for example, at full load with respect to the current scenario.		
Value type	[0...] integer value	**Unit**	1

A.1.10.3 Recognised road boundaries – capability

Table A.16 — Signal: Recognised road boundaries – capability

Name	Recognised road boundaries – capability		
Description	The signal "Recognised road boundaries – capability" (A.1.10.3) provides the performance of the sensor cluster's interface. It determines how many entities the sensor cluster is guaranteed to recognise, for example, at full load with respect to the current scenario.		
Value type	[0...] integer value	**Unit**	1

A.1.10.4 Recognised general landmarks – capability

Table A.17 — Signal: Recognised general landmarks – capability

Name	Recognised general landmarks – capability		
Description	The signal "Recognised general landmarks – capability" (A.1.10.4) provides the performance of the sensor cluster's interface. It determines how many entities the sensor cluster is guaranteed to recognise, for example, at full load with respect to the current scenario.		
Value type	[0...] integer value	**Unit**	1

A.1.10.5 Recognised traffic signs – capability

Table A.18 — Signal: Recognised traffic signs – capability

Name	Recognised traffic signs – capability		
Description	The signal "Recognised traffic signs – capability" (A.1.10.5) provides the performance of the sensor cluster's interface. It determines how many entities the sensor cluster is guaranteed to recognise, for example, at full load with respect to the current scenario.		
Value type	[0...] integer value	**Unit**	1

A.1.10.6 Recognised traffic lights – capability

Table A.19 — Signal: Recognised traffic lights – capability

Name	Recognised traffic lights – capability		
Description	The signal “Recognised traffic lights – capability” (A.1.10.6) provides the performance of the sensor cluster’s interface. It determines how many entities the sensor cluster is guaranteed to recognise, for example, at full load with respect to the current scenario.		
Value type	[0...] integer value	**Unit**	1

A.1.10.7 Recognised features – capability

Table A.20 — Signal: Recognised features – capability

Name	Recognised features – capability		
Description	The signal “Recognised features – capability” (A.1.10.7) provides the performance of the sensor cluster’s interface. It determines how many entities the sensor cluster is guaranteed to recognise, for example, at full load with respect to the current scenario.		
Value type	[0...] integer value	**Unit**	1

A.1.10.8 Recognised detections – capability

Table A.21 — Signal: Recognised detections – capability

Name	Recognised detections – capability		
Description	The signal “Recognised detections – capability” (A.1.10.8) provides the performance of the sensor cluster’s interface. It determines how many entities the sensor cluster is guaranteed to recognise, for example, at full load with respect to the current scenario.		
Value type	[0...] integer value	**Unit**	1

A.1.10.9 Recognised shapes – capability

Table A.22 — Signal: Recognised shapes – capability

Name	Recognised shapes – capability		
Description	The signal “Recognised shapes – capability” (A.1.10.9) provides the performance of the sensor cluster’s interface. It determines how many entities the sensor cluster is guaranteed to recognise, for example, at full load with respect to the current scenario.		
Value type	[0...] integer value	**Unit**	1

A.1.11 Recognised <...> – status

The performance of the sensor cluster’s interface determines a state of the recognition process under, for example, full load.

All entities use the same list of example enumerators.

Table A.23 — Enumeration: Recognised <...> – status – Example enumerators

Name	Description	RL enumerator
Recognised<...>Status_Normal	The performance of the sensor cluster’s interface is enough to process all recognised entities.	“exemplary”
Recognised<...>Status_PreLimits	The performance of the sensor cluster’s interface is close to the limits to process all recognised entities.	“exemplary”

Table A.23 *(continued)*

Name	Description	RL enumerator
Recognised<...>Status_Limits	The performance of the sensor cluster's interface is not enough to process all recognised entities.	"exemplary"

A.1.11.1 Recognised potentially moving objects – status

Table A.24 — Signal: Recognised potentially moving objects – status

Name	Recognised potentially moving objects – status		
Description	The signal "Recognised potentially moving objects – status" (A.1.11.1) determines whether the sensor cluster's interface can process all recognised entities or whether it works at the sensor cluster's performance limits.		
Value type	enumeration	**Unit**	1

A.1.11.2 Recognised road markings – status

Table A.25 — Signal: Recognised road markings – status

Name	Recognised road markings – status		
Description	The signal "Recognised road markings – status" (A.1.11.2) determines whether the sensor cluster's interface can process all recognised entities or whether it works at the sensor cluster's performance limits.		
Value type	enumeration	**Unit**	1

A.1.11.3 Recognised road boundaries – status

Table A.26 — Signal: Recognised road boundaries – status

Name	Recognised road boundaries – status		
Description	The signal "Recognised road boundaries – status" (A.1.11.3) determines whether the sensor cluster's interface can process all recognised entities or whether it works at the sensor cluster's performance limits.		
Value type	enumeration	**Unit**	1

A.1.11.4 Recognised general landmarks – status

Table A.27 — Signal: Recognised general landmarks – status

Name	Recognised general landmarks – status		
Description	The signal "Recognised general landmarks – status" (A.1.11.4) determines whether the sensor cluster's interface can process all recognised entities or whether it works at the sensor cluster's performance limits.		
Value type	enumeration	**Unit**	1

A.1.11.5 Recognised traffic signs – status

Table A.28 — Signal: Recognised traffic signs – status

Name	Recognised traffic signs – status		
Description	The signal "Recognised traffic signs – status" (A.1.11.5) determines whether the sensor cluster's interface can process all recognised entities or whether it works at the sensor cluster's performance limits.		
Value type	enumeration	**Unit**	1

A.1.11.6 Recognised traffic lights – status

Table A.29 — Signal: Recognised traffic lights – status

Name	Recognised traffic lights – status		
Description	The signal "Recognised traffic lights – status" (A.1.11.6) determines whether the sensor cluster's interface can process all recognised entities or whether it works at the sensor cluster's performance limits.		
Value type	enumeration	**Unit**	1

A.1.11.7 Recognised features – status

Table A.30 — Signal: Recognised features – status

Name	Recognised features – status		
Description	The signal "Recognised features – status" (A.1.11.7) determines whether the sensor cluster's interface can process all recognised entities or whether it works at the sensor cluster's performance limits.		
Value type	enumeration	**Unit**	1

A.1.11.8 Recognised detections – status

Table A.31 — Signal: Recognised detections – status

Name	Recognised detections – status		
Description	The signal "Recognised detections – status" (A.1.11.8) determines whether the sensor cluster's interface can process all recognised entities or whether it works at the sensor cluster's performance limits.		
Value type	enumeration	**Unit**	1

A.1.11.9 Recognised shapes – status

Table A.32 — Signal: Recognised shapes – status

Name	Recognised shapes – status		
Description	The signal "Recognised shapes – status" (A.1.11.9) determines whether the sensor cluster's interface can process all recognised entities or whether it works at the sensor cluster's performance limits.		
Value type	enumeration	**Unit**	1

A.1.12 Number of valid <...>

Each interface (except SHII) sends one or more lists of entities with defined entity types. For each list, the number of valid transferred entities is known.

A.1.12.1 Number of valid potentially moving objects

Table A.33 — Signal: Number of valid potentially moving objects

Name	Number of valid potentially moving objects

Table A.33 *(continued)*

Description	The signal's "Number of valid potentially moving objects" (A.1.12.1) value defines the number of valid entities in the potentially moving object list of the PMOI. Additional information: The entries in the list may or may not be sorted. All those elements of the list which can be used by the fusion unit are regarded as valid. The list includes a complete and prioritised set of valid entities for the signal's "Time stamp – <...>" (A.1.5) value and not only a differential list between two sequential time stamps.		
Value type	[0...] integer value	**Unit**	1

A.1.12.2 Number of valid road markings

Table A.34 — Signal: Number of valid road markings

Name	Number of valid road markings		
Description	The signal's "Number of valid road markings" (A.1.12.2) value defines the number of valid entities in the road marking list of the RDOI. Additional information: The entries in the list may or may not be sorted. All those elements of the list which can be used by the fusion unit are regarded as valid. The list includes a complete and prioritised set of valid entities for the signal's "Time stamp – <...>" (A.1.5) value and not only a differential list between two sequential time stamps. Each road marking has polynomial or polyline, for example, containing >= 2 vertices.		
Value type	[0...] integer value	**Unit**	1

A.1.12.3 Number of valid road boundaries

Table A.35 — Signal: Number of valid road boundaries

Name	Number of valid road boundaries		
Description	The signal's "Number of valid road boundaries" (A.1.12.3) value defines the number of valid entities in the road boundary list of the RDOI. Additional information: The entries in the list may or may not be sorted. All those elements of the list which can be used by the fusion unit are regarded as valid. The list includes a complete and prioritised set of valid entities for the signal's "Time stamp – <...>" (A.1.5) value and not only a differential list between two sequential time stamps. Each road boundary has polynomial or polyline, for example, containing >= 2 vertices.		
Value type	[0...] integer value	**Unit**	1

A.1.12.4 Number of valid general landmarks

Table A.36 — Signal: Number of valid general landmarks

Name	Number of valid general landmarks

Table A.36 *(continued)*

Description	The signal's "Number of valid general landmarks" (A.1.12.4) value defines the number of valid entities in the general landmark list of the SOI. Additional information: The entries in the list may or may not be sorted. All those elements of the list which can be used by the fusion unit are regarded as valid. The list includes a complete and prioritised set of valid entities for the signal's "Time stamp – <...>" (A.1.5) value and not only a differential list between two sequential time stamps.		
Value type	[0...] integer value	**Unit**	1

A.1.12.5 Number of valid traffic signs

Table A.37 — Signal: Number of valid traffic signs

Name	Number of valid traffic signs		
Description	The signal's "Number of valid traffic signs" (A.1.12.5) value defines the number of valid entities in the traffic sign list of the SOI. Additional information: The entries in the list may or may not be sorted. All those elements of the list which can be used by the fusion unit are regarded as valid. The list includes a complete and prioritised set of valid entities for the signal's "Time stamp – <...>" (A.1.5) value and not only a differential list between two sequential time stamps.		
Value type	[0...] integer value	**Unit**	1

A.1.12.6 Number of valid traffic lights

Table A.38 — Signal: Number of valid traffic lights

Name	Number of valid traffic lights		
Description	The signal's "Number of valid traffic lights" (A.1.12.6) value defines the number of valid entities in the traffic light list of the SOI. Additional information: The entries in the list may or may not be sorted. All those elements of the list which can be used by the fusion unit are regarded as valid. The list includes a complete and prioritised set of valid entities for the signal's "Time stamp – <...>" (A.1.5) value and not only a differential list between two sequential time stamps.		
Value type	[0...] integer value	**Unit**	1

A.1.12.7 Number of valid features

Table A.39 — Signal: Number of valid features

Name	Number of valid features

Table A.39 *(continued)*

Description	The signal's "Number of valid features" (A.1.12.7) value defines the number of valid entities in the feature list of an FLI. One measurement can generate not only unique features, if there are, for example, ambiguities in camera shape or colour classification. Additional information: The entries in the list may or may not be sorted. All those elements of the list which can be used by the fusion unit are regarded as valid. The list includes a complete and prioritised set of valid entities for the signal's "Time stamp – <...>" (A.1.5) value and not only a differential list between two sequential time stamps.		
Value type	[0...] integer value	**Unit**	1

A.1.12.8 Number of valid detections

Table A.40 — Signal: Number of valid detections

Name	Number of valid detections		
Description	The signal's "Number of valid detections" (A.1.12.8) value defines the number of valid entities in the detection list of a DLI. One measurement can generate more than one similar detection, if there are ambiguities. Additional information: The entries in the list may or may not be sorted. All those elements of the list which can be used by the fusion unit are regarded as valid. The list includes a complete and prioritised set of valid entities for the signal's "Time stamp – <...>" (A.1.5) value and not only a differential list between two sequential time stamps.		
Value type	[0...] integer value	**Unit**	1

A.1.12.9 Number of valid shapes

Table A.41 — Signal: Number of valid shapes

Name	Number of valid shapes		
Description	The signal's "Number of valid shapes" (A.1.12.9) value defines the number of valid entities in the detection list of the CDI or CFI. Additional information: The entries in the list may or may not be sorted. All those elements of the list which can be used by the fusion unit are regarded as valid. The list includes a complete and prioritised set of valid entities for the signal's "Time stamp – <...>" (A.1.5) value and not only a differential list between two sequential time stamps.		
Value type	[0...] integer value	**Unit**	1

A.1.12.10 Number of valid field-of-view segments

Table A.42 — Signal: Number of valid field-of-view segments

Name	Number of valid field-of-view segments

Table A.42 *(continued)*

Description	The signal's "Number of valid field-of-view segments" (A.1.12.10) value defines the number of FOVs sensor- or sensor cluster-segments of the SPI. Additional information: The entries in the list may or may not be sorted. All those elements of the list which can be used by the fusion unit are regarded as valid. The list includes a complete and prioritised set of valid entities for the signal's "Time stamp – <...>" (A.1.5) value and not only a differential list between two sequential time stamps.		
Value type	[0...] integer value	**Unit**	1

A.1.13 Tracking motion model

Table A.43 — Signal: Tracking motion model

Name	Tracking motion model		
Description	The object tracking uses a motion model for tracking moving items.		
Value type	enumeration	**Unit**	1

Table A.44 — Enumeration: Tracking motion model – Example enumerators

Name	**Description**	**RL enumerator**
TrackingMotionModel_ConstantVelocity	The motion model uses constant velocity.	"exemplary"
TrackingMotionModel_ConstantAcceleration	The motion model uses constant acceleration.	"exemplary"
TrackingMotionModel_ConstantTurnRate	The motion model uses constant turn rate.	"exemplary"

A.1.14 Motion type

Table A.45 — Signal: Motion type

Name	Motion type		
Description	The signal "Motion type" (A.1.14) provides the applied motion type of the interface's motion signals.		
Value type	enumeration	**Unit**	1

Table A.46 — Enumeration: Motion type – Example enumerators

Name	**Description**	**RL enumerator**
MotionType_Relative	The interface sends relative motion values of the object described in ego-vehicle axis system {x, y, z}.	"exemplary"
MotionType_Absolute	The interface sends absolute motion values of the object in world coordinate system described in ego-vehicle axis system {x, y, z}.	"exemplary"

A.1.15 Colour model type

Table A.47 — Signal: Colour model type

Name	Colour model type

Table A.47 *(continued)*

Description	The signal “Colour model type” (A.1.15) provides the applied colour model for the interface. Additional information: The signal “Colour model type” (A.1.15) defines also the number of elements for the signal “Colour values” (see A.2.69, A.3.12, A.4.28).		
Value type	enumeration	**Unit**	1

Table A.48 — Enumeration: Colour model type – Example enumerators

Name	Description	RL enumerator
ColourModelType_Grey	The colour type is grey scale and 1 colour value is provided.	“exemplary”
ColourModelType_RGB	The colour type is red, green, blue and 3 colour values are provided.	“exemplary”
ColourModelType_HSV	The colour type is hue, saturation, value and 3 colour values are provided.	“exemplary”
ColourModelType_LUV	The colour type is luminance and colour coordinates U, V and 3 colour values are provided.	“exemplary”
ColourModelType_ RGBIR	The colour type is red, green, blue, infrared and 4 colour values are provided.	“exemplary”
ColourModelType_ColourList	Each value references a predefined colour and 1 colour value for reference values is provided. Requirement: By using the enumerator “ColourModelType_ColourList” the signal’s “Colour values” (see A.2.69, A.3.12, A.4.28) values are defined during the system design phase.	“exemplary”

A.1.16 Radial velocity ambiguity domain {begin, end}

Table A.49 — Signal: Radial velocity ambiguity domain {begin, end}

Name	Radial velocity ambiguity domain {begin, end}		
Description	The signal “Radial velocity ambiguity domain {begin, end}” (A.1.16) provides a description of the Doppler ambiguity caused by under-sampling.		
Value type	2D vector real value	**Unit**	(m/s, m/s)

A.1.17 Range ambiguity domain {begin, end}

Table A.50 — Signal: Range ambiguity domain {begin, end}

Name	Range ambiguity domain {begin, end}		
Description	The range of the ambiguity domain is defined by {begin} and {end}. The defined range is only one possible solution for an ambiguous measurement method (for example, under-sampling).		
Value type	2D vector real value	**Unit**	(m, m)

A.1.18 Angle azimuth ambiguity domain {begin, end}

Table A.51 — Signal: Angle azimuth ambiguity domain {begin, end}

Name	Angle azimuth ambiguity domain {begin, end}

Table A.51 *(continued)*

Description	The azimuth angle of the ambiguity domain is defined by {begin} and {end}. The defined azimuth angle is only one possible solution for an ambiguous measurement method.		
Value type	2D vector real value	**Unit**	(rad, rad)

A.1.19 Angle elevation ambiguity domain {begin, end}

Table A.52 — Signal: Angle elevation ambiguity domain {begin, end}

Name	Angle elevation ambiguity domain {begin, end}		
Description	The elevation angle of the ambiguity domain is defined by {begin} and {end}. The defined elevation angle is only one possible solution for an ambiguous measurement method.		
Value type	2D vector real value	**Unit**	(rad, rad)

A.1.20 Interface applicability

Table A.53 — Signal: Interface applicability

Name	Interface applicability		
Description	The signal "Interface applicability" (A.1.20) defines the interface(s) for which the message is relevant for. The implementation of the interface may provide several interfaces with different signal's "Interface applicability" (A.1.20) enumerators. Additional information: The signal "Interface applicability" (A.1.20) is necessary, for example, for a service-oriented architecture.		
Value type	enumeration	**Unit**	1

Table A.54 — Enumeration: Interface applicability – Example enumerators

Name	**Description**	**RL enumerator**
InterfaceApplicability_OLI	The applicability of the interface is for all OLIs.	"exemplary"
InterfaceApplicability_FLI	The applicability of the interface is for the FLI.	"exemplary"
InterfaceApplicability_DLI	The applicability of the interface is for the DLI.	"exemplary"
InterfaceApplicability_FLIAndDLI	The applicability of the interface is for FLI and DLI.	"exemplary"
InterfaceApplicability_PMOI	The applicability of the interface is for the PMOI.	"exemplary"
InterfaceApplicability_RDOI	The applicability of the interface is for the RDOI.	"exemplary"
InterfaceApplicability_SOI	The applicability of the interface is for the SOI.	"exemplary"
InterfaceApplicability_All	The applicability of the interface is for all provided interfaces.	"exemplary"

A.1.21 Vehicle coordinate system type

Table A.55 — Signal: Vehicle coordinate system type

Name	Vehicle coordinate system type		
Description	The signal "Vehicle coordinate system type" (A.1.21) defines the reference vehicle coordinate system for the interfaces of the sensor cluster.		
Value type	enumeration	**Unit**	1

Table A.56 — Enumeration: Vehicle coordinate system type

Name	Description	RL enumerator
VehicleCoordinateSystemType_RearAxle	Use definition "vehicle rear-axle coordinate system" (3.7.12) for the ego-vehicle coordinate system of the sensor or the sensor cluster.	Mandatory
VehicleCoordinateSystemType_RoadLevel	Use definition "vehicle road-level coordinate system" (3.7.15) for the ego-vehicle coordinate system of the sensor or the sensor cluster.	Mandatory

A.1.22 Sensor origin point {x, y, z}

Table A.57 — Signal: Sensor origin point {x, y, z}

Name	Sensor origin point {x, y, z}		
Description	The signal "Sensor origin point {x, y, z}" (A.1.22) provides the position of the origin of the sensing element's sensor coordinate system in the ego-vehicle coordinate system [see signal "Vehicle coordinate system type" (A.1.21)]. Additional information: The signal "Sensor origin point – correction {x, y, z}" (A.5.43) is already applied.		
Value type	3D vector real value	**Unit**	(m, m, m)

A.1.23 Sensor origin point {x, y, z} – error

Table A.58 — Signal: Sensor origin point {x, y, z} – error

Name	Sensor origin point {x, y, z} – error		
Description	The uncertainty of the sensor origin point calibration should be included in the interface {x, y, z}. Additional information: If only the trueness error [see "Error model implementation" (B.4.1) and "Trueness and precision error values" (B.4.1.3.2)] in the overall system is known, the sensor origin point error shall not be sent. No high dynamic motion of the vehicle is compensated. The signal "Sensor origin point – correction {x, y, z} – error" (A.5.44) is already applied.		
Value type	scalar/vector/matrix real value [see subclause "Error model implementation" (B.4.1)]	**Unit**	see signal "Sensor origin point {x, y, z}" (A.1.22)

A.1.24 Sensor orientation {yaw, pitch, roll}

Table A.59 — Signal: Sensor orientation {yaw, pitch, roll}

Name	Sensor orientation {yaw, pitch, roll}

Table A.59 *(continued)*

Description	The signal "Sensor orientation {yaw, pitch, roll}" (A.1.24) provides the relative orientation of the sensor axis system with respect to the vehicle axis system. The sensor orientation represents the current orientation of the sensor origin point to the best knowledge. It includes the pose estimation given by the calibration. The uncertainty of this estimation is given with the corresponding error. The estimation does not include effects due to short-time dynamics, such as pitch angle changes during braking, but can include long-time effects resulting, for example, from luggage in the trunk. Additional information: Sensor orientation is given with {yaw, pitch, roll} angles, which have to be separated in a defined procedure. For example, the rotations are to be performed yaw first (around the *Z*-axis), pitch second (around the new *Y*-axis) and roll third (around the new *X*-axis) following the definition according to ISO 8855:2011. The signal "Sensor orientation – correction {yaw, pitch, roll}" (A.5.46) is already applied.		
Value type	3D vector real value	**Unit**	(rad, rad, rad)

A.1.25 Sensor orientation {yaw, pitch, roll} – error

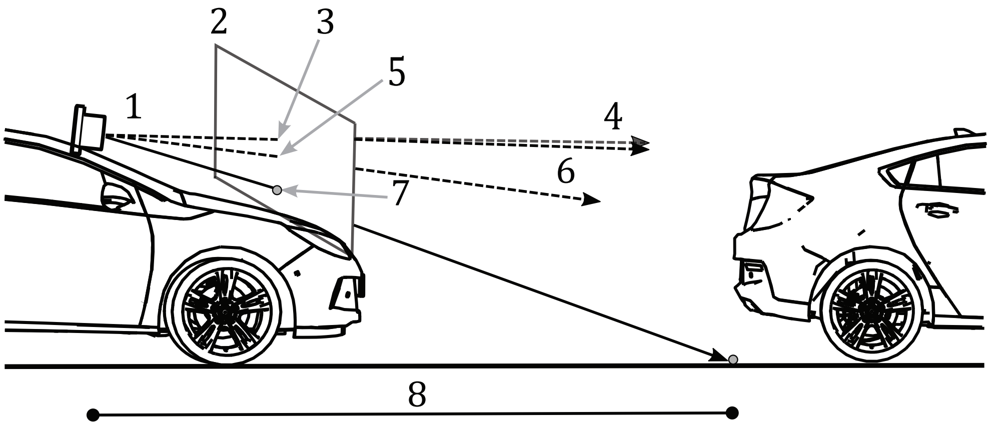
Table A.60 — Signal: Sensor orientation {yaw, pitch, roll} – error

Name	Sensor orientation {yaw, pitch, roll} – error		
Description	The uncertainty of the sensor orientation calibration should be included in the interface {yaw, pitch, roll}. Additional information: If only the trueness error in the overall system is known, the sensor orientation error shall not be sent. No high dynamic motion of the vehicle is compensated. The signal "Sensor orientation – correction {yaw, pitch, roll} – error" (A.5.47) is already applied.		
Value type	scalar/vector/matrix real value [see subclause "Error model implementation" (B.4.1)]	**Unit**	see signal "Sensor orientation {yaw, pitch, roll}" (A.1.24)

A.1.26 Vanishing point {azimuth, elevation}

Table A.61 — Signal: Vanishing point {azimuth, elevation}

Name	Vanishing point {azimuth, elevation}		
Description	A set of lines in the image plane that corresponds to a set of parallel surface lines in the 3D world space converges to a common point in the image space, known as the signal "Vanishing point {azimuth, elevation}" (A.1.26). Additional information: Figure A.1 shows the distance estimation using the vanishing point.		
Value type	2D vector real value	**Unit**	(rad, rad)



Key

1 camera sensor
2 image plane (enlarged)
3 vanishing point
4 ego-vehicle *X*-axis direction
5 image centre of the camera sensor
6 optical axis of the camera
7 image of a point on road level of the camera sensor
8 distance (in *X*-axis direction)

Figure A.1 — Example for distance estimation

A.1.27 Vanishing point {azimuth, elevation} – error

Table A.62 — Signal: Vanishing point {azimuth, elevation} – error

Name	Vanishing point {azimuth, elevation} – error		
Description	The signal "Vanishing point {azimuth, elevation} – error" (A.1.27) provides the uncertainty of the signal "Vanishing point {azimuth, elevation}" (A.1.26).		
Value type	scalar/vector/matrix real value [see subclause "Error model implementation" (B.4.1)]	**Unit**	see signal "Vanishing point {azimuth, elevation}" (A.1.26)

A.2 Object level entity signals

Signals are defined for the entities of the subclause "Potentially moving object interface" (7.3), the subclause "Road object interface" (7.4) and the subclause "Static object interface" (7.5) in this subclause (see Tables A.63 - A.228). These interfaces are located at OLI.

The OLI originates from several sensors. Therefore, at object level, the term sensor cluster is always used, even if a single sensor is serving the interface.

A.2.1 Existence probability – object level

Table A.63 — Signal: Existence probability – object level

Name	Existence probability – object level

Table A.63 *(continued)*

Description	The signal "Existence probability – object level" (A.2.1) represents the recognition quality and sensor's certainty that the object exists. A higher value indicates an object which is more certain to exist while a lower value indicates a potential false positive object. Additional information: Low values may be possible in the first cycles of the object's appearance.		
Value type	[0...100] real value	**Unit**	%

A.2.2 Object ID

Table A.64 — Signal: Object ID

Name	Object ID		
Description	The signal "Object ID" (A.2.2) provides the unique ID of the object within the object lists of the sensor cluster's interfaces. Additional information: The signal's "Object ID" (A.2.2) value is only unique for each sensor cluster's detected entities of OLIs. The signal's "Object ID" (A.2.2) value is unique for the same entity in multiple cycles. Reuse of the signal's "Object ID" (A.2.2) value is possible under defined conditions. Requirement: The reuse of the signal's "Object ID" (A.2.2) value shall be well-defined.		
Value type	integer value	**Unit**	1

A.2.3 Object grouping ID

Table A.65 — Signal: Object grouping ID

Name	Object grouping ID		
Description	A sensor cluster can group objects from the same object interface or different object interfaces of the sensor cluster which are linked together. All objects from the real-world entity have the same signal's "Object grouping ID" (A.2.3) value. The ID is unique considering all OLIs of a sensor cluster. In cases in which it is not clear, if it is a single traffic participant or multiple, the link can be used. EXAMPLE 1 Long vehicles (for example, truck with trailer) can be described by potentially moving objects. In this case the value logically connects or links one object with another object. EXAMPLE 2 A construction site vehicle with a traffic sign at the rear of the vehicle. Additional information: The signal "Object grouping ID" (A.2.3) := 0 if object is not associated with another object. Requirement: The reuse of the signal's "Object grouping ID" (A.2.3) values shall be well-defined.		
Value type	integer value	**Unit**	1

A.2.4 Age

Table A.66 — Signal: Age

Name	Age		
Description	The signal “Age” (A.2.4) provides the amount of time that this object has been currently tracked. Additional information: The signal “Age” (A.2.4) := current time stamp – time stamp when the track for this object has been initiated.		
Value type	[0…] real value	**Unit**	s

A.2.5 Number of valid observations – object level

Table A.67 — Signal: Number of valid observations – object level

Name	Number of valid observations – object level		
Description	The signal “Number of valid observations – object level” (A.2.5) provides the current number of valid tuples [specifically the signal “Time stamp reference – object level” (A.2.6) and “Observation status – object level” (A.2.7)].		
Value type	[0…] integer value	**Unit**	1

A.2.6 Time stamp reference – object level

Table A.68 — Signal: Time stamp reference – object level

Name	Time stamp reference – object level		
Description	The signal “Time stamp reference – object level” (A.2.6) provides a reference to a previously sent interface message with the referenced time stamp for signal “Time stamp – <…>” (A.1.5).		
Value type	real value	**Unit**	s

A.2.7 Observation status – object level

Table A.69 — Signal: Observation status – object level

Name	Observation status – object level		
Description	The signal “Observation status – object level” (A.2.7) provides the observation status of the object, which was recognised in a previous cycle [see signal “Time stamp reference – object level” (A.2.6)].		
Value type	enumeration	**Unit**	1

Table A.70 — Enumeration: Observation status – object level – Example enumerators

Name	Description	RL enumerator
Observation_True	The object was observed in the current cycle.	“exemplary”
Observation_False	The object was not observed in the current cycle. It may be predicted in the current cycle.	“exemplary”

A.2.8 Track quality

Table A.71 — Signal: Track quality

Name	Track quality

Table A.71 *(continued)*

Description	The signal "Track quality" (A.2.8) value describes how well the track was predicted of the current observation. Usually there is a variety of signals which are used to set up a track. The quality of the track is changing gradually from a precise prediction where all values met the prediction very close down to a prediction which is based on less signals and where the deviation gets large. A value close to 100 % indicates that both the prediction is very precise and most of available signals are used to build the track. A very low value (near 0 %) means a very high deviation of the observation from the predicted track and that the availability of the signals is lower.		
Value type	[0...100] real value	**Unit**	%

A.2.9 Measurement status – object level

Table A.72 — Signal: Measurement status – object level

Name	Measurement status – object level		
Description	The signal "Measurement status – object level" (A.2.9) provides the current measurement status of the object. The objects should be sent out as fast as possible, even if not all statuses are observed and the object's signals might not be fully filled.		
Value type	enumeration	**Unit**	1

Table A.73 — Enumeration: Measurement status – object level – Example enumerators

Name	Description	RL enumerator
MeasurementStatus_Measured	The object was updated with new information from a sensor in the last update cycle of this object list.	"exemplary"
MeasurementStatus_New	This indicates the first occurrence of this particular object in the object list.	"exemplary"
MeasurementStatus_Predicted	The object was predicted without any new information with respect to the previous update cycle.	"exemplary"
MeasurementStatus_PartlyMeasured	The object was partly updated with new information from a sensor in the last update cycle of this object list. The measurement was incomplete and could, therefore, only partially update the signals of this object. The remaining signals of this object are unchanged or predicted for this measurement cycle.	"exemplary"
MeasurementStatus_Invalid	The object is invalid in this cycle.	"exemplary"
MeasurementStatus_PredictedOccluded	Tracked object is temporarily occluded by another entity.	"exemplary"

A.2.10 Number of valid potentially moving object classifications

Table A.74 — Signal: Number of valid potentially moving object classifications

Name	Number of valid potentially moving object classifications		
Description	The signal "Number of valid potentially moving object classifications" (A.2.10) provides the current number of valid tuples [specifically the signal "Potentially moving object classification type" (A.2.11) and the signal "Potentially moving object classification type – confidence" (A.2.12)] for a potentially moving object.		
Value type	[0...] integer value	**Unit**	1

A.2.11 Potentially moving object classification type

Table A.75 — Signal: Potentially moving object classification type

Name	Potentially moving object classification type

Table A.75 *(continued)*

Description	The signal "Potentially moving object classification type" (A.2.11) provides a list of classification types for potentially moving objects.		
Value type	enumeration	**Unit**	1

Table A.76 — Enumeration: Potentially moving object classification type – Example enumerators

Name	Description	RL enumerator
PMOType_Car	The classification type represents an object car.	"exemplary"
PMOType_HeavyTruck	The classification type represents an object heavy truck.	"exemplary"
PMOType_Van	The classification type represents an object van.	"exemplary"
PMOType_Bus	The classification type represents an object bus.	"exemplary"
PMOType_Trailer	The classification type represents an object trailer.	"exemplary"
PMOType_Train	The classification type represents an object train.	"exemplary"
PMOType_OtherVehicle	The classification type represents an object unidentified vehicle.	"exemplary"
PMOType_Motorbike	The classification type represents an object motorbike.	"exemplary"
PMOType_Animal	The classification type represents an object animal.	"exemplary"
PMOType_Bicycle	The classification type represents an object bicycle.	"exemplary"
PMOType_Tricycle	The classification type represents an object tricycle.	"exemplary"
PMOType_Pedestrian	The classification type represents an object pedestrian.	"exemplary"
PMOType_Wheelchair	The classification type represents an object wheelchair.	"exemplary"

A.2.12 Potentially moving object classification type – confidence

Table A.77 — Signal: Potentially moving object classification type – confidence

Name	Potentially moving object classification type – confidence		
Description	The signal "Potentially moving object classification type – confidence" (A.2.12) provides the certainty for the corresponding signal "Potentially moving object classification type" (A.2.11).		
Value type	[0...100] real value	**Unit**	%

A.2.13 Position – object level {x, y, z}

Table A.78 — Signal: Position – object level {x, y, z}

Name	Position – object level {x, y, z}		
Description	The signal "Position – object level {x, y, z}" (A.2.13) provides the position, which the signal "Reference point" (A.2.17) refers to. It is defined in the ego-vehicle coordinate system [see signal "Vehicle coordinate system type" (A.1.21)]. Additional information: The origin point of the object coordinate system is the ego-vehicle coordinate system (see Figure A.3). If no signal "Reference point" (A.2.17) is defined, the reference point is located at the geometric centre (specifically the enumerator "RP_MidsideMidwidthMidheight") of the bounding box.		
Value type	3D vector real value	**Unit**	(m, m, m)

A.2.14 Position – object level {x, y, z} – error

Table A.79 — Signal: Position – object level {x, y, z} – error

Name	Position – object level {x, y, z} – error		
Description	The signal "Position – object level {x, y, z} – error" (A.2.14) provides the error which represents the uncertainty of the state estimation of the signal's "Position – object level {x, y, z}" (A.2.13) position (see Figure A.3).		
Value type	scalar/vector/matrix real value [see subclause "Error model implementation" (B.4.1)]	**Unit**	see signal "Position – object level {x, y, z}" (A.2.13)

A.2.15 Orientation {yaw, pitch, roll}

Table A.80 — Signal: Orientation {yaw, pitch, roll}

Name	Orientation {yaw, pitch, roll}		
Description	The signal "Orientation {yaw, pitch, roll}" (A.2.15) provides the orientation of the object's bounding box defined in the ego-vehicle coordinate system [see signal "Vehicle coordinate system type" (A.1.21)]. Additional information: A potentially moving object may move in another direction as the *X*-axis of the bounding box [see signal "Bounding box extent {length, width, height}" (A.2.19)]. The axis system of the bounding box and the object's axis system may be different.		
Value type	3D vector real value	**Unit**	(rad, rad, rad)

A.2.16 Orientation {yaw, pitch, roll} – error

Table A.81 — Signal: Orientation {yaw, pitch, roll} – error

Name	Orientation {yaw, pitch, roll} – error		
Description	The signal "Orientation {yaw, pitch, roll} – error" (A.2.16) provides the error which represents the uncertainty of the state estimation of the signal's "Orientation {yaw, pitch, roll}" (A.2.15) values for orientation (see Figure A.2).		
Value type	scalar/vector/matrix real value [see subclause "Error model implementation" (B.4.1)]	**Unit**	see signal "Orientation {yaw, pitch, roll}" (A.2.15)

A.2.17 Reference point

Table A.82 — Signal: Reference point

Name	Reference point

Table A.82 *(continued)*

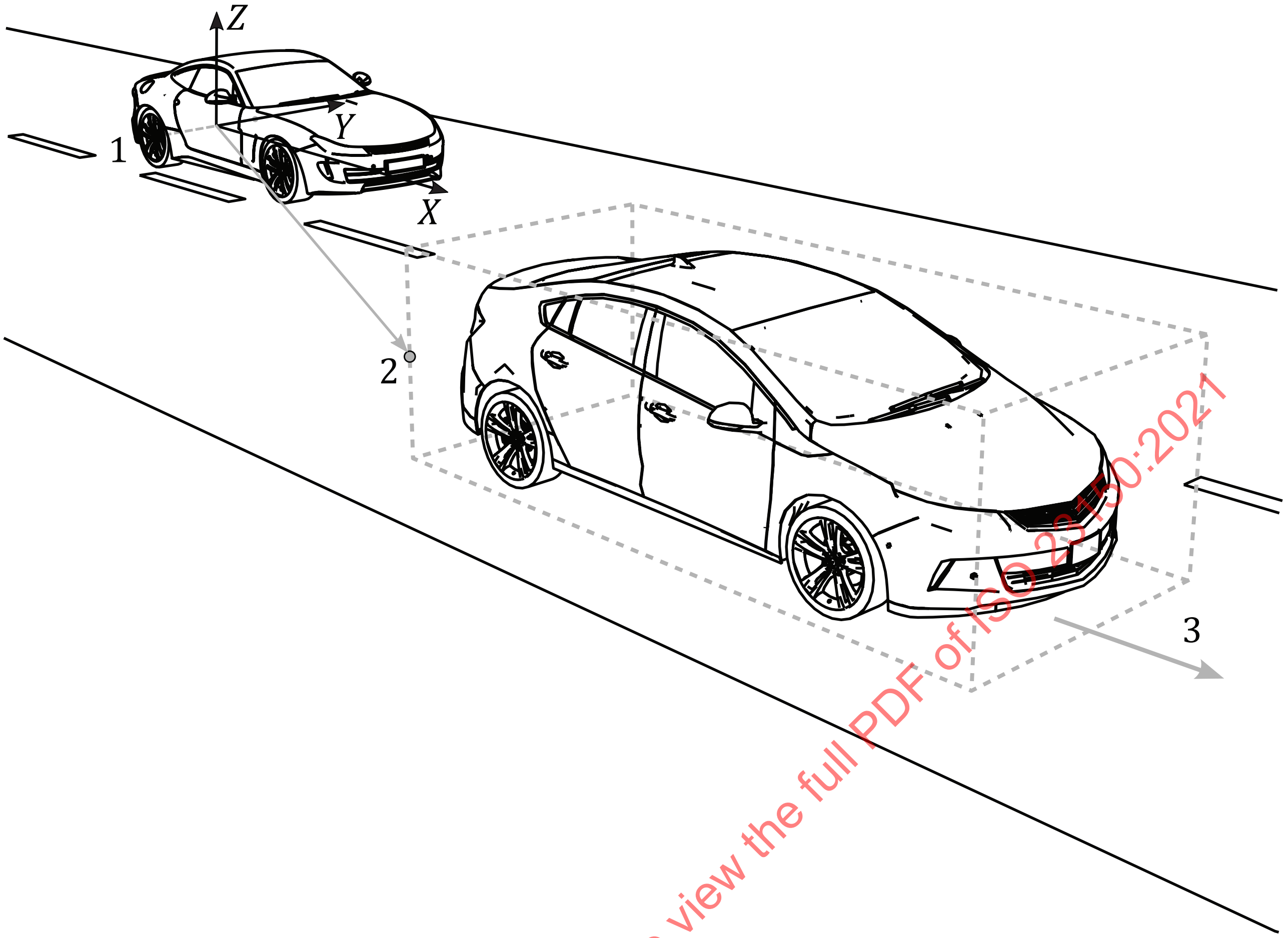
Description	The signal “Reference point” (A.2.17) provides the outer edges of the recognised object’s bounding box (see Figure A.2). During tracking, this point can change in each cycle. Additional information: Only valid for cuboid model. The reference point is always on the outer edge with z value "middle" (middle height), if it is not defined in the enumeration. “middle” is the geometrical middle of the bounding box for each axis. If the signal is not part of the implemented interface, the reference point is located in the midside, midwidth, midheight of the object's bounding box (see enumerator “RP_MidsideMidwidthMidheight”). Front/rear, left/right, top/bottom are defined for the bounding box with respect to the object’s orientation [see signal “Orientation {yaw, pitch, roll}” (A.2.15)]. Naming for enumerators: - *X*-axis of the bounding box: Rear – Midside – Front; - *Y*-axis of the bounding box: Right – Midwidth – Left; - *Z*-axis of the bounding box: Bottom – Midheight – Top.		
Value type	enumeration	**Unit**	1

Table A.83 — Enumeration: Reference point – Example enumerators

Name	**Description**	**RL enumerator**
RP_FrontLeftTop	The reference point is located at the front, left, top corner of the bounding box.	“exemplary”
RP_FrontMidwidthTop	The reference point is located in the middle of the front, top edge of the bounding box.	“exemplary”
RP_FrontRightTop	The reference point is located at the front, right, top corner of the bounding box.	“exemplary”
RP_MidsideLeftTop	The reference point is located in the middle of the left, top edge of the bounding box.	“exemplary”
RP_MidsideMidwidthTop	The reference point is located at the geometric centre of the bounding box’s top plane.	“exemplary”
RP_MidsideRightTop	The reference point is located in the middle of the right, top edge of the bounding box.	“exemplary”
RP_RearLeftTop	The reference point is located at the rear, left, top corner of the bounding box.	“exemplary”
RP_RearMidwidthTop	The reference point is located in the middle of the rear, top edge of the bounding box.	“exemplary”
RP_RearRightTop	The reference point is located at the rear, right, top corner of the bounding box.	“exemplary”
RP_FrontLeftMidheight	The reference point is located in the middle of the front, left edge of the bounding box.	“exemplary”
RP_FrontMidwidthMidheight	The reference point is located at the geometric centre of the bounding box’s front plane.	“exemplary”
RP_FrontRightMidheight	The reference point is located in the middle of the front, right edge of the bounding box.	“exemplary”
RP_MidsideLeftMidheight	The reference point is located at the geometric centre of the bounding box’s left-side plane.	“exemplary”
RP_MidsideMidwidthMidheight	The reference point is located at the geometric centre of the bounding box.	“exemplary”

Table A.83 *(continued)*

Name	Description	RL enumerator
RP_MidsideRightMidheight	The reference point is located at the geometric centre of the bounding box's right-side plane.	"exemplary"
RP_RearLeftMidheight	The reference point is located in the middle of the rear, left edge of the bounding box.	"exemplary"
RP_RearMidwidthMid-height	The reference point is located at the geometric centre of the bounding box's rear plane.	"exemplary"
RP_RearRightMidheight	The reference point is located in the middle of the rear, right edge of the bounding box.	"exemplary"
RP_FrontLeftBottom	The reference point is located at the front, left, bottom corner of the bounding box.	"exemplary"
RP_FrontMidwidthBottom	The reference point is located in the middle of the front, bottom edge of the bounding box.	"exemplary"
RP_FrontRightBottom	The reference point is located at the front, right, bottom corner of the bounding box.	"exemplary"
RP_MidsideLeftBottom	The reference point is located in the middle of the left, bottom edge of the bounding box.	"exemplary"
RP_MidsideMidwidthBot-tom	The reference point is located at the geometric centre of the bounding box's bottom plane.	"exemplary"
RP_MidsideRightBottom	The reference point is located in the middle of the right, bottom edge of the bounding box.	"exemplary"
RP_RearLeftBottom	The reference point is located at the rear, left, bottom corner of the bounding box.	"exemplary"
RP_RearMidwidthBottom	The reference point is located in the middle of the rear, bottom edge of the bounding box.	"exemplary"
RP_RearRightBottom	The reference point is located at the rear, right, bottom corner of the bounding box.	"exemplary"



Key

1 vehicle's rear axle with ego-vehicle coordinate system

2 point of signal "Reference point" (A.2.17) with enumerator "RP_RearRightMidheight" refers to rear right midheight of the potentially moving object's bounding box and orientation at this point [see signal "Orientation {yaw, pitch, roll}" (A.2.15)]

3 potentially moving object's moving direction

Figure A.2 — Example for reference point of a potentially moving object's bounding box

A.2.18 Road level

Table A.84 — Signal: Road level

Name	Road level		
Description	A road level indication provides information about relevance to ego-vehicle road level, for example, in case no {z} value for position can be given. Additional information: In situations it happens that a target vehicle is detected by the sensor, but the vehicle is on different road levels compared to the ego-vehicle (for example, motorway bridge ahead on which a vehicle is crossing the motorway). This signal indicates the relevance to the ego-vehicle road level.		
Value type	enumeration	**Unit**	1

Table A.85 — Enumeration: Road level – Example enumerators

Name	Description	RL enumerator
RoadLevel_EgoRoadLevel	The object is on the same road level as the ego-vehicle.	“exemplary”
RoadLevel_RoadLevelAbove	The object is on a different road level as the ego-vehicle. It is on a road level above.	“exemplary”
RoadLevel_RoadLevelBelow	The object is on a different road level as the ego-vehicle. It is on a road level below.	“exemplary”

A.2.19 Bounding box extent {length, width, height}

Table A.86 — Signal: Bounding box extent {length, width, height}

Name	Bounding box extent {length, width, height}		
Description	{length}: distance between the rear plane and the front plane of the bounding box. It is the object dimension along the *X*-axis of the bounding box axis system (see Table A.83). {width}: distance between the right-side plane and the left-side plane of the bounding box. It is the object dimension along the *Y*-axis of the bounding box axis system (see Table A.83). {height}: distance between the bottom plane and the top plane of the bounding box. It is the object dimension along the *Z*-axis of the bounding box axis system (see Table A.83). Additional information: The orientation of the bounding box is defined by the signal “Orientation {yaw, pitch, roll}” (A.2.15). Requirement: The degree to which small structures are part of the extent of the bounding box shall be defined by the sensor manufacturer during the system design phase [see signal “Included geometric structures” (A.2.22)].		
Value type	3D vector real value	**Unit**	(m, m, m)

A.2.20 Bounding box extent {length, width, height} – error

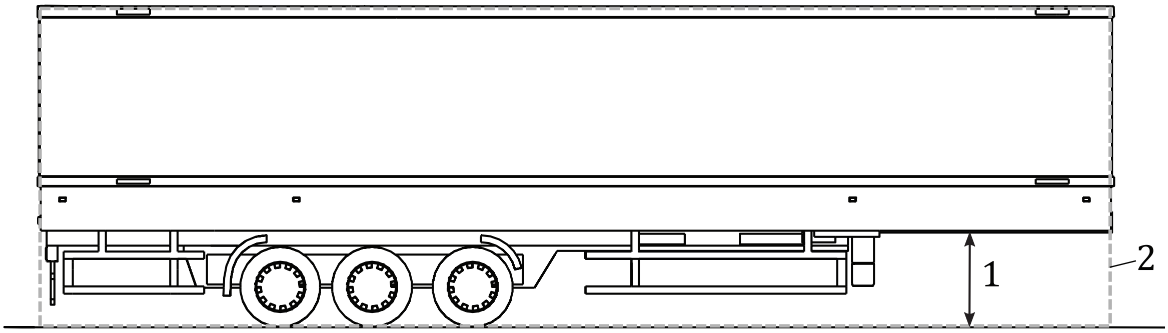
Table A.87 — Signal: Bounding box extent {length, width, height} – error

Name	Bounding box extent {length, width, height} – error		
Description	The signal “Bounding box extent {length, width, height} – error” (A.2.20) provides the error of the signal “Bounding box extent {length, width, height}” (A.2.19).		
Value type	scalar/vector/matrix real value [see subclause “Error model implementation” (B.4.1)]	**Unit**	see signal “Bounding box extent {length, width, height}” (A.2.19)

A.2.21 Bounding box ground clearance

Table A.88 — Signal: Bounding box ground clearance

Name	Bounding box ground clearance		
Description	The signal “Bounding box ground clearance” (A.2.21) provides the maximum height of the clearance area between ground and object lower shape. The bounding box is indicated with the dashed box (see Figure A.3).		
Value type	[0...] real value	**Unit**	m



Key

1 bounding box ground clearance

2 bounding box

Figure A.3 — Example for the signal "Bounding box ground clearance" (A.2.21)

A.2.22 Included geometric structures

Table A.89 — Signal: Included geometric structures

Name	Included geometric structures		
Description	The signal "Included geometric structures" (A.2.22) provides the geometrical structures that are taken into account in the bounding boxes.		
Value type	enumeration	**Unit**	1

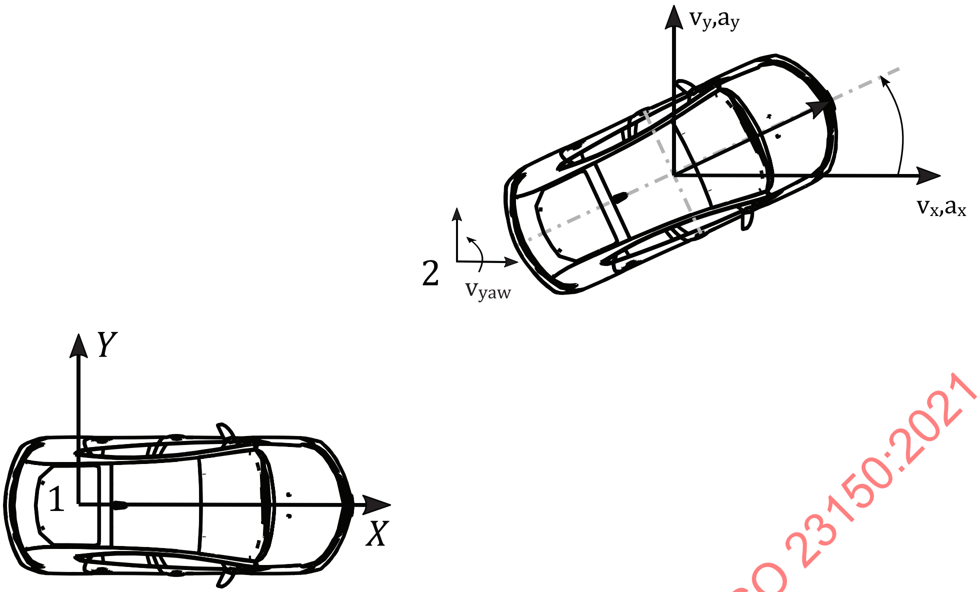
Table A.90 — Enumeration: Included geometric structures – Example enumerators

Name	**Description**	**RL enumerator**
IncludedGeometricStructures_WithoutMirrors	The bounding box dimensions exclude vehicle mirrors [see signal "Bounding box extent {length, width, height}" (A.2.19)].	"exemplary"
IncludedGeometricStructures_WithMirrors	The bounding box dimensions include vehicle mirrors [see signal "Bounding box extent {length, width, height}" (A.2.19)].	"exemplary"

A.2.23 Velocity {x, y, z} – object level

Table A.91 — Signal: Velocity {x, y, z} – object level

Name	Velocity {x, y, z} – object level		
Description	The signal "Velocity {x, y, z} – object level" (A.2.23) provides the velocity of the object with respect to the signal "Motion type" (A.1.14) (see Figure A.4) in the ego-vehicle coordinate system [see signal "Vehicle coordinate system type" (A.1.21)].		
Value type	3D vector real value	**Unit**	(m/s, m/s, m/s)



Key

1 ego-vehicle coordinate system

2 instantaneous centre of rotation of the potentially moving object

Figure A.4 — Example for an ego-vehicle and object dynamics of a potentially moving object

A.2.24 Velocity {x, y, z} – object level – error

Table A.92 — Signal: Velocity {x, y, z} – object level – error

Name	Velocity {x, y, z} – object level – error		
Description	The signal “Velocity {x, y, z} – object level – error” (A.2.24) provides the error which represents the uncertainty of the state estimation of {x, y, z} with respect to the signal “Motion type” (A.1.14) (see Figure A.4) in the ego-vehicle coordinate system [see signal “Vehicle coordinate system type” (A.1.21)].		
Value type	scalar/vector/matrix real value [see subclause “Error model implementation” (B.4.1)]	**Unit**	see signal “Velocity {x, y, z} – object level” (A.2.23)

A.2.25 Acceleration {x, y, z}

Table A.93 — Signal: Acceleration {x, y, z}

Name	Acceleration {x, y, z}		
Description	The signal “Acceleration {x, y, z}” (A.2.25) provides the acceleration of the object with respect to the signal “Motion type” (A.1.14) (see Figure A.4) in the ego-vehicle coordinate system [see signal “Vehicle coordinate system type” (A.1.21)].		
Value type	3D vector real value	**Unit**	(m/s^2, m/s^2, m/s^2)

A.2.26 Acceleration {x, y, z} – error

Table A.94 — Signal: Acceleration {x, y, z} – error

Name	Acceleration {x, y, z} – error

Table A.94 *(continued)*

Description	The signal "Acceleration {x, y, z} – error" (A.2.26) provides the error which represents the uncertainty of the state estimation of {x, y, z} acceleration with respect to the signal "Motion type" (A.1.14) (see Figure A.4) in the ego-vehicle coordinate system [see signal "Vehicle coordinate system type" (A.1.21)].		
Value type	scalar/vector/matrix real value [see subclause "Error model implementation" (B.4.1)]	**Unit**	see signal "Acceleration {x, y, z}" (A.2.25)

A.2.27 Instantaneous centre of rotation {x, y}

Table A.95 — Signal: Instantaneous centre of rotation {x, y}

Name	Instantaneous centre of rotation {x, y}		
Description	The signal "Instantaneous centre of rotation {x, y}" (A.2.27) provides the centre-point of the instantaneous rotation in the ego-vehicle coordinate system [see signal "Vehicle coordinate system type" (A.1.21)]. The centre can be located outside the object's bounding box (see Figure A.4). Additional information: The centre-point can be arbitrary for the object. The centre-point is the origin of the rotation rates [see signal "Rotation rate at instantaneous centre of rotation {yaw}" (A.2.29)] and it is defined in the ego-vehicle coordinate system [see signal "Vehicle coordinate system type" (A.1.21)].		
Value type	2D vector real value	**Unit**	(m, m)

A.2.28 Instantaneous centre of rotation {x, y} – error

Table A.96 — Signal: Instantaneous centre of rotation {x, y} – error

Name	Instantaneous centre of rotation {x, y} – error		
Description	The signal "Instantaneous centre of rotation {x, y} – error" (A.2.28) provides the error of the state estimation of the signal "Instantaneous centre of rotation {x, y}" (A.2.27) (see Figure A.4) in the ego-vehicle coordinate system [see signal "Vehicle coordinate system type" (A.1.21)].		
Value type	scalar/vector/matrix real value [see subclause "Error model implementation" (B.4.1)]	**Unit**	see signal "Instantaneous centre of rotation {x, y}" (A.2.27)

A.2.29 Rotation rate at instantaneous centre of rotation {yaw}

Table A.97 — Signal: Rotation rate at instantaneous centre of rotation {yaw}

Name	Rotation rate at instantaneous centre of rotation {yaw}		
Description	The signal "Rotation rate at instantaneous centre of rotation {yaw}" (A.2.29) provides the {yaw}-rate of the object at the signal "Instantaneous centre of rotation {x, y}" (A.2.27) with respect to the signal "Motion type" (A.1.14) (see Figure A.4) and with respect to the ego-vehicle coordinate system [see signal "Vehicle coordinate system type" (A.1.21)].		
Value type	1D vector real value	**Unit**	rad/s

A.2.30 Rotation rate at instantaneous centre of rotation {yaw} – error

Table A.98 — Signal: Rotation rate at instantaneous centre of rotation {yaw} – error

Name	Rotation rate at instantaneous centre of rotation {yaw} – error

Table A.98 *(continued)*

Description	The signal "Rotation rate at instantaneous centre of rotation {yaw} – error" (A.2.30) provides the error of the {yaw}-rate at the object's signal "Instantaneous centre of rotation {x, y}" (A.2.27) with respect to the signal "Motion type" (A.1.14) (see Figure A.4) and with respect to the ego-vehicle coordinate system [see signal "Vehicle coordinate system type" (A.1.21)].		
Value type	scalar/vector/matrix real value [see subclause "Error model implementation" (B.4.1)]	**Unit**	see signal "Rotation rate at instantaneous centre of rotation {yaw}" (A.2.29)

A.2.31 Movement status

Table A.99 — Signal: Movement status

Name	Movement status		
Description	The signal "Movement status" (A.2.31) provides the information about a possible movement of the object during tracking.		
Value type	enumeration	**Unit**	1

Table A.100 — Enumeration: Movement status – Example enumerators

Name	**Description**	**RL enumerator**
MovementStatus_Moving	The object is currently moving.	"exemplary"
MovementStatus_StoppedMoving	The object movement was detected in the tracking history of the object.	"exemplary"
MovementStatus_Stationary	Until now no object movement was detected in the tracking history of the object.	"exemplary"

A.2.32 Number of valid lights

Table A.101 — Signal: Number of valid lights

Name	Number of valid lights		
Description	The signal "Number of valid lights" (A.2.32) provides the current number of valid tuples [specifically the signal "Light type" (A.2.33) and the signal "Light status" (A.2.34) for a potentially moving object.		
Value type	[0..] integer value	**Unit**	1

A.2.33 Light type

Table A.102 — Signal: Light type

Name	Light type		
Description	The signal "Light type" (A.2.33) provides a list of electric light classifications for a potentially moving object.		
Value type	Enumeration	**Unit**	1

Table A.103 — Enumeration: Light type – Example enumerators

Name	**Description**	**RL enumerator**
LightType_LeftTurnIndicator	The light is the vehicle's left turn indicator.	"exemplary"
LightType_RightTurnIndicator	The light is the vehicle's right turn indicator.	"exemplary"
LightType_HazardFlashLight	The light is the vehicle's hazard flash light.	"exemplary"

Table A.103 *(continued)*

Name	Description	RL enumerator
LightType_LeftBrakeLight	The light is the vehicle's left brake light.	"exemplary"
LightType_RightBrakeLight	The light is the vehicle's right brake light.	"exemplary"
LightType_CentreBrakeLight	The light is the vehicle's centre brake light.	"exemplary"
LightType_LeftOtherLight	The light is the vehicle's left light (no turn indicator, flash- or brake light).	"exemplary"
LightType_RightOtherLight	The light is the vehicle's right light (no turn indicator, flash- or brake light).	"exemplary"
LightType_CentreOtherLight	The light is a vehicle's light which is not on the left or right side (no turn indicator, flash- or brake light).	"exemplary"

A.2.34 Light status

Table A.104 — Signal: Light status

Name	Light status		
Description	The signal "Light status" (A.2.34) provides the status of a potentially moving object's electric light [see signal "Light type" (A.2.33)]. Additional information: It is possible to have more than one active light signals (for example, low beam and blinker).		
Value type	enumeration	**Unit**	1

Table A.105 — Enumeration: Light status – Example enumerators

Name	Description	RL enumerator
LightStatus_Off	The light status is off.	"exemplary"
LightStatus_On	The light status is on.	"exemplary"
LightStatus_Flash	The light status is cyclic flashing.	"exemplary"
LightStatus_Brake	The light status indicates braking.	"exemplary"
LightStatus_Warning	The light status indicates warning.	"exemplary"
LightStatus_Other	The light status could not be determined.	"exemplary"

A.2.35 Number of valid person's poses

Table A.106 — Signal: Number of valid person's poses

Name	Number of valid person's poses		
Description	The signal "Number of valid person's poses" (A.2.35) provides the current number of valid tuples [specifically the signal "Person pose type" (A.2.36) and the signal "Person pose {yaw, pitch, roll}" (A.2.37)] for a potentially moving object. All poses are related to one person (for example, the object is a pedestrian).		
Value type	[0...] integer value	**Unit**	1

A.2.36 Person pose type

Table A.107 — Signal: Person pose type

Name	Person pose type
Description	The signal "Person pose type" (A.2.36) provides the pose type which is defined by the signal "Person pose {yaw, pitch, roll}" (A.2.37).

Table A.107 *(continued)*

Value type	enumeration	**Unit**	1

Table A.108 — Enumeration: Person pose type – Example enumerators

Name	Description	RL enumerator
PersonPoseType_Head	The person's head pose is defined by the corresponding signal “Person pose {yaw, pitch, roll}” (A.2.37).	“exemplary”
PersonPoseType_UpperBody	The person's upper-body pose is defined by the corresponding signal “Person pose {yaw, pitch, roll}” (A.2.37).	“exemplary”
PersonPoseType_LeftHand	The person's left-hand pose is defined by the corresponding signal “Person pose {yaw, pitch, roll}” (A.2.37). The normal vector of the hand palm is used to determine the pose relative to the ego-vehicle axis system [see signal “Vehicle coordinate system type” (A.1.21)].	“exemplary”
PersonPoseType_RightHand	The person's right-hand pose is defined by the corresponding signal “Person pose {yaw, pitch, roll}” (A.2.37). The normal vector of the hand palm is used to determine the pose relative to the ego-vehicle axis system [see signal “Vehicle coordinate system type” (A.1.21)].	“exemplary”
PersonPoseType_LeftLowerArm	The person's left-lower arm pose is defined by the corresponding signal “Person pose {yaw, pitch, roll}” (A.2.37). The long axis of the body extremities (orientation towards the end of the extremity) is used to determine the pose relative to the ego-vehicle axis system [see signal “Vehicle coordinate system type” (A.1.21)].	“exemplary”
PersonPoseType_RightLowerArm	The person's right-lower arm pose is defined by the corresponding signal “Person pose {yaw, pitch, roll}” (A.2.37). The long axis of the body extremities (orientation towards the end of the extremity) is used to determine the pose relative to the ego-vehicle axis system [see signal “Vehicle coordinate system type” (A.1.21)].	“exemplary”
PersonPoseType_LeftUpperArm	The person's left-upper arm pose is defined by the corresponding signal “Person pose {yaw, pitch, roll}” (A.2.37). The long axis of the body extremities (orientation towards the end of the extremity) is used to determine the pose relative to the ego-vehicle axis system [see signal “Vehicle coordinate system type” (A.1.21)].	“exemplary”
PersonPoseType_RightUpperArm	The person's right-upper arm pose is defined by the corresponding signal “Person pose {yaw, pitch, roll}” (A.2.37). The long axis of the body extremities (orientation towards the end of the extremity) is used to determine the pose relative to the ego-vehicle axis system [see signal “Vehicle coordinate system type” (A.1.21)].	“exemplary”
PersonPoseType_LeftUpperLeg	The person's left-upper leg pose is defined by the corresponding signal “Person pose {yaw, pitch, roll}” (A.2.37). The long axis of the body extremities (orientation towards the end of the extremity) is used to determine the pose relative to the ego-vehicle axis system [see signal “Vehicle coordinate system type” (A.1.21)].	“exemplary”

Table A.108 *(continued)*

Name	Description	RL enumerator
PersonPoseType_RightUpperLeg	The person's right-upper leg pose is defined by the corresponding signal "Person pose {yaw, pitch, roll}" (A.2.37). The long axis of the body extremities (orientation towards the end of the extremity) is used to determine the pose relative to the ego-vehicle axis system [see signal "Vehicle coordinate system type" (A.1.21)].	"exemplary"
PersonPoseType_LeftLowerLeg	The person's left-lower leg pose is defined by the corresponding signal "Person pose {yaw, pitch, roll}" (A.2.37). The long axis of the body extremities (orientation towards the end of the extremity) is used to determine the pose relative to the ego-vehicle axis system [see signal "Vehicle coordinate system type" (A.1.21)].	"exemplary"
PersonPoseType_RightLowerLeg	The person's right-lower leg pose is defined by the corresponding signal "Person pose {yaw, pitch, roll}" (A.2.37). The long axis of the body extremities (orientation towards the end of the extremity) is used to determine the pose relative to the ego-vehicle axis system [see signal "Vehicle coordinate system type" (A.1.21)].	"exemplary"
PersonPoseType_LeftFoot	The person's left-foot pose is defined by the corresponding signal "Person pose {yaw, pitch, roll}" (A.2.37). The foot sole direction towards the toe is used to determine the pose relative to the ego-vehicle axis system [see signal "Vehicle coordinate system type" (A.1.21)].	"exemplary"
PersonPoseType_RightFoot	The person's right-foot pose is defined by the corresponding signal "Person pose {yaw, pitch, roll}" (A.2.37). The foot sole direction towards the toe is used to determine the pose relative to the ego-vehicle axis system [see signal "Vehicle coordinate system type" (A.1.21)].	"exemplary"

A.2.37 Person pose {yaw, pitch, roll}

Table A.109 — Signal: Person pose {yaw, pitch, roll}

Name	Person pose {yaw, pitch, roll}		
Description	The signal "Person pose {yaw, pitch, roll}" (A.2.37) provides a defined pedestrian-pose for behaviour prediction. Additional information: The signal "Person pose {yaw, pitch, roll}" (A.2.37) describes the angles of the pose [defined by signal "Person pose type" (A.2.36)] with respect to the ego-vehicle orientation, specifically the ego-vehicle axis system [see signal "Vehicle coordinate system type" (A.1.21)].		
Value type	3D vector real value	**Unit**	(rad, rad, rad)

A.2.38 Person pose {yaw, pitch, roll} – error

Table A.110 — Signal: Person pose {yaw, pitch, roll} – error

Name	Person pose {yaw, pitch, roll} – error
Description	The signal "Person pose {yaw, pitch, roll} – error" (A.2.38) provides the error of a defined pedestrian-pose [see signal "Person pose {yaw, pitch, roll} – error" (A.2.38)].

Table A.110 *(continued)*

Value type	scalar/vector/matrix real value [see subclause “Error model implementation” (B.4.1)]	**Unit**	see signal “Person pose {yaw, pitch, roll}– error” (A.2.38)

A.2.39 Object lane association

Table A.111 — Signal: Object lane association

Name	Object lane association		
Description	The signal “Object lane association” (A.2.39) provides an association of the object to neighbouring lanes with respect to the ego-vehicle’s lane.		
Value type	enumeration	**Unit**	1

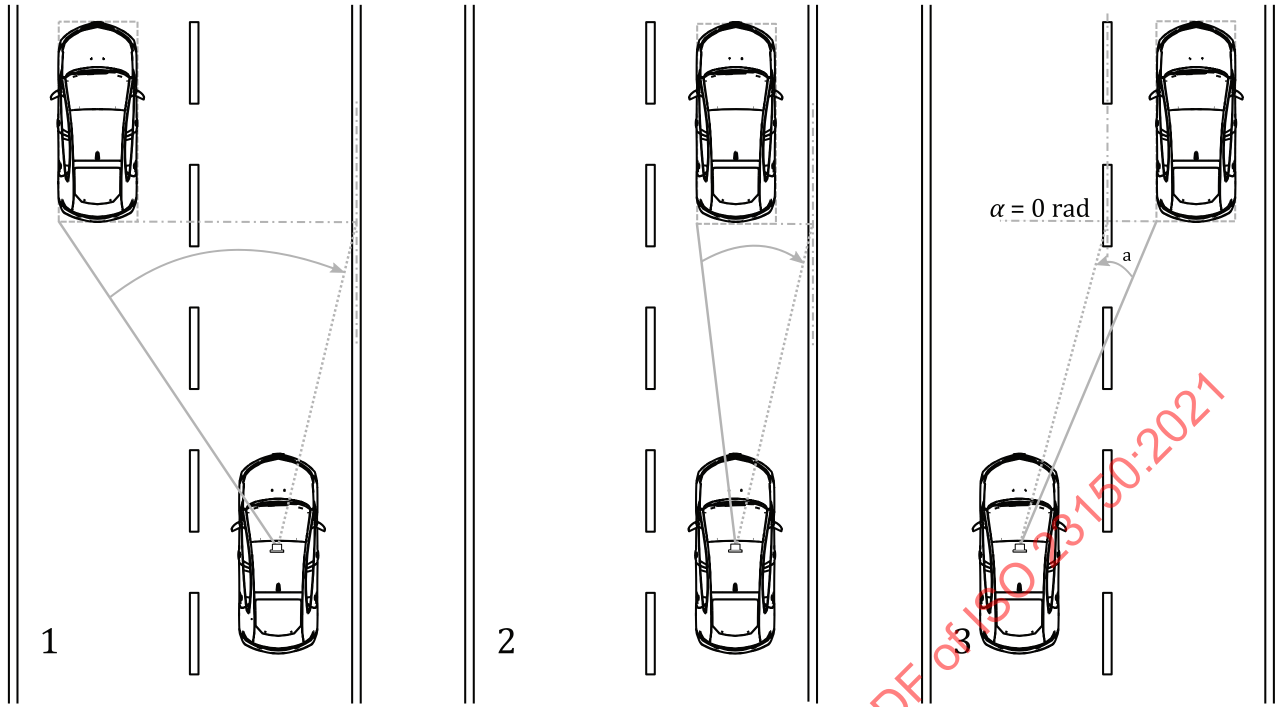
Table A.112 — Enumeration: Object lane association – Example enumerators

Name	**Description**	**RL enumerator**
ObjectLaneAssociation_EgoLane	The object is in the ego lane.	“exemplary”
ObjectLaneAssociation_Left1Lane	The object is in the first-left neighbouring lane.	“exemplary”
ObjectLaneAssociation_Right1Lane	The object is in the first-right neighbouring lane.	“exemplary”
ObjectLaneAssociation_EgoRight1Lane	The object is located between the ego and right neighbouring lane.	“exemplary”
ObjectLaneAssociation_EgoLeft1Lane	The object is located between the ego and left neighbouring lane.	“exemplary”
ObjectLaneAssociation_Left2Lane	The object is in the second-left neighbouring lane.	“exemplary”
ObjectLaneAssociation_Right2Lane	The object is in the second-right neighbouring lane.	“exemplary”

A.2.40 Angle between object edge and lane {left edge right lane, right edge left lane}

Table A.113 — Signal: Angle between object edge and lane {left edge right lane, right edge left lane}

Name	Angle between object edge and lane {left edge right lane, right edge left lane}		
Description	{left edge right lane}: angle between the left edge of the object and the right centre of the ego-vehicle lane’s road marking. {right edge left lane}: angle between the right edge of the object and the left centre of the ego-vehicle lane’s road marking. Additional information: Signal, definition of 0 rad and positive angle is defined in Figure A.5. The edge is always the nearest edge of the object’s bounding box to the ego-vehicle. The centre of the road marking is the basis for the calculation.		
Value type	2D vector real value	**Unit**	(rad, rad)



Key

1 object is in the first-left neighbouring lane of the ego-vehicle

2 object is in the ego-vehicle lane

3 object is in the first-right neighbouring lane of the ego-vehicle

a Position angle.

Figure A.5 — Example for the signal: "Angle between object edge and lane {left edge right lane, right edge left lane}" (A.2.40) with left edge of the object and the right centre of the ego-vehicle lane's road marking

A.2.41 Angle between object edge and lane {left edge right lane, right edge left lane} – error

Table A.114 — Signal: Angle between object edge and lane {left edge right lane, right edge left lane} – error

Name	Angle between object edge and lane {left edge right lane, right edge left lane} – error		
Description	The signal "Angle between object edge and lane {left edge right lane, right edge left lane} – error" (A.2.41) provides the error of the signal "Angle between object edge and lane {left edge right lane, right edge left lane}" (A.2.40).		
Value type	scalar/vector/matrix real value [see subclause "Error model implementation" (B.4.1)]	**Unit**	see signal "Angle between object edge and lane {left edge right lane, right edge left lane}" (A.2.40)

A.2.42 Percentage side lane {left, right}

Table A.115 — Signal: Percentage side lane {left, right}

Name	Percentage side lane {left, right}

Table A.115 *(continued)*

Description	The signal "Percentage side lane {left, right}" (A.2.42) provides the {left, right} assignment of the object to specified lane in percent. Additional information: The signal "Percentage side lane {left, right}" (A.2.42) provides the percentage value of the object width in the corresponding {left, right} lane. The centre of the road marking is the basis for the calculation.		
Value type	[0...100] 2D vector real value	**Unit**	(%, %)

A.2.43 Angular position {azimuth}

Table A.116 — Signal: Angular position {azimuth}

Name	Angular position {azimuth}		
Description	The signal "Angular position {azimuth}" (A.2.43) provides the lateral position of the bounding box centre (midside, midwidth, midheight) measured in the 2D image as {azimuth} angle in the sensor's coordinate system.		
Value type	1D vector real value	**Unit**	rad

A.2.44 Angular velocity {azimuth}

Table A.117 — Signal: Angular velocity {azimuth}

Name	Angular velocity {azimuth}		
Description	The signal "Angular velocity {azimuth}" (A.2.44) provides the angular {azimuth} velocity of the bounding box centre (midside, midwidth, midheight) measured in the 2D image in the sensor's coordinate system. Additional information: Requires calibrated sensor.		
Value type	1D vector real value	**Unit**	rad/s

A.2.45 Scale change – object level

Table A.118 — Signal: Scale change – object level

Name	Scale change – object level		
Description	The signal "Scale change – object level" (A.2.45) provides the change in scale of the object in sensor's projection plane. It indicates whether the object gets closer to the ego-vehicle (possible collision) or if it is departing from the ego-vehicle.		
Value type	real value	**Unit**	%/s

A.2.46 Entity radar cross section

Table A.119 — Signal: Entity radar cross section

Name	Entity radar cross section

Table A.119 *(continued)*

Description	The signal "Entity radar cross section" (A.2.46) provides the radar cross section (RCS) of the recognised object. Additional information: It is the representative RCS of the complete recognised object.		
Value type	real value	**Unit**	dB m^2

A.2.47 Entity lidar reflectivity

Table A.120 — Signal: Entity lidar reflectivity

Name	Entity lidar reflectivity		
Description	The signal "Entity lidar reflectivity" (A.2.47) provides the lidar reflectivity of the recognised object, assuming homogeneous Lambertian reflectivity for the used laser wave length. Additional information: It is the representative lidar reflectivity of the complete recognised object.		
Value type	real value	**Unit**	%

A.2.48 Road type

Table A.121 — Signal: Road type

Name	Road type		
Description	The signal "Road type" (A.2.48) provides the ego-vehicle relevant type of the road.		
Value type	enumeration	**Unit**	1

Table A.122 — Enumeration: Road type – Example enumerators

Name	**Description**	**RL enumerator**
RoadType_Highway	The road is a highway.	"exemplary"
RoadType_Rural	The road is a rural road.	"exemplary"
RoadType_City	The road is a city road.	"exemplary"
RoadType_OffRoad	The road is unpaved.	"exemplary"

A.2.49 Number of valid road surface classifications

Table A.123 — Signal: Number of valid road surface classifications

Name	Number of valid road surface classifications		
Description	The signal "Number of valid road surface classifications" (A.2.49) provides the current number of valid tuples for road surface classifications. The tuples are defined by the signals "Road surface classification type" (A.2.50) and "Road surface classification type – confidence" (A.2.51).		
Value type	[0...] integer value	**Unit**	1

A.2.50 Road surface classification type

Table A.124 — Signal: Road surface classification type

Name	Road surface classification type

Table A.124 *(continued)*

Description	The signal "Road surface classification type" (A.2.50) provides the ego-vehicle relevant type of the road surface.		
Value type	enumeration	**Unit**	1

Table A.125 — Enumeration: Road surface classification type – Example enumerators

Name	Description	RL enumerator
RoadSurfaceType_RomanRoad	The road is a cobblestone street.	"exemplary"
RoadSurfaceType_OffRoad	The road is an off-road road.	"exemplary"
RoadSurfaceType_Flat	The road is a smooth surface road.	"exemplary"
RoadSurfaceType_Bumpy	The road is bumpy.	"exemplary"

A.2.51 Road surface classification type – confidence

Table A.126 — Signal: Road surface classification type – confidence

Name	Road surface classification type – confidence		
Description	The signal "Road surface classification type – confidence" (A.2.51) provides the certainty for the corresponding signal "Road surface classification type" (A.2.50).		
Value type	[0...100] real value	**Unit**	%

A.2.52 Road surface roughness

Table A.127 — Signal: Road surface roughness

Name	Road surface roughness		
Description	The signal "Road surface roughness" (A.2.52) provides the ego-vehicle relevant roughness or unevenness of the road surface. Additional information: The international roughness index (IRI) has a value range from 0 = mm/m (smooth surface which is equivalent to driving on a plate of glass) up to 20 mm/m (a very rough road).		
Value type	[0...] real value	**Unit**	mm/m

A.2.53 Number of valid road surface condition classifications

Table A.128 — Signal: Number of valid road surface condition classifications

Name	Number of valid road surface condition classifications		
Description	The signal "Number of valid road surface condition classifications" (A.2.53) provides the current number of valid tuples for road surface condition classifications. The tuples are defined by the signals "Road surface condition classification type" (A.2.54) and "Road surface condition classification type – confidence" (A.2.55).		
Value type	[0...] integer value	**Unit**	1

A.2.54 Road surface condition classification type

Table A.129 — Signal: Road surface condition classification type

Name	Road surface condition classification type
Description	The signal "Road surface condition classification type" (A.2.54) provides the ego-vehicle relevant condition of the road surface.

Table A.129 *(continued)*

Value type	enumeration	**Unit**	1

Table A.130 — Enumeration: Road surface condition classification type – Example enumerators

Name	Description	RL enumerator
RoadSurfaceCondition_Dry	The road surface is dry.	“exemplary”
RoadSurfaceCondition_Wet	The road surface is wet.	“exemplary”
RoadSurfaceCondition_Snow	The road surface is covered with snow.	“exemplary”
RoadSurfaceCondition_Ice	The road surface is covered with ice.	“exemplary”

A.2.55 Road surface condition classification type – confidence

Table A.131 — Signal: Road surface condition classification type – confidence

Name	Road surface condition classification type – confidence		
Description	The signal “Road surface condition classification type – confidence” (A.2.55) provides the certainty for the corresponding signal “Road surface condition classification type” (A.2.54).		
Value type	[0...100] real value	**Unit**	%

A.2.56 Number of valid road marking classifications

Table A.132 — Signal: Number of valid road marking classifications

Name	Number of valid road marking classifications		
Description	The signal “Number of valid road marking classifications” (A.2.56) provides the current number of valid tuples for road marking types. The tuples are defined by the signals “Road marking type” (A.2.57) and “Road marking type – confidence” (A.2.58).		
Value type	[0...] integer value	**Unit**	1

A.2.57 Road marking type

Table A.133 — Signal: Road marking type

Name	Road marking type		
Description	The signal “Road marking type” (A.2.57) indicates the type of the road markings. Additional information: All road markings can be represented as polyline with further information of heading and width (same information as a bounding box can deliver).		
Value type	enumeration	**Unit**	1

Table A.134 — Enumeration: Road marking type – Example enumerators

Name	Description	RL enumerator
RoadMarkingType_Solid	The road marking is solid. It could also be a stop line.	“exemplary”
RoadMarkingType_CentreLineDashedMarking	The centre line road marking is dashed. See ISO 11270:2014, Annex B.	“exemplary”
RoadMarkingType_EdgeLineDashedMarking	The edge line road marking is dashed. See ISO 11270:2014, Annex B.	“exemplary”

Table A.134 *(continued)*

Name	Description	RL enumerator
RoadMarkingType_Triangular	The road marking is a line of triangles.	“exemplary”
RoadMarkingType_DoubleLineSolid	The road marking has two lines and the most inner line (with respect to the ego-vehicle) is solid.	“exemplary”
RoadMarkingType_CentreLineDoubleLineDashed	The centre line road marking has two lines and the most inner line (with respect to the ego-vehicle) is dashed. See ISO 11270:2014, Annex B.	“exemplary”
RoadMarkingType_EdgeLineDoubleLineDashed	The edge line road marking has two lines and the most inner line (with respect to the ego-vehicle) is dashed. See ISO 11270:2014, Annex B.	“exemplary”
RoadMarkingType_MultipleLineSolid	The road marking has more than two lines and the most inner line (with respect to the ego-vehicle) is solid.	“exemplary”
RoadMarkingType_CentreLineMultipleLineDashed	The centre line road marking has more than two lines and the most inner line (with respect to the ego-vehicle) is dashed. See ISO 11270:2014, Annex B.	“exemplary”
RoadMarkingType_EdgeLineMultipleLineDashed	The edge line road marking has more than two lines and the most inner line (with respect to the ego-vehicle) is dashed. See ISO 11270:2014, Annex B.	“exemplary”
RoadMarkingType_BottsDotsCatsEyes	The road marking consists of Botts' dots or cats' eyes.	“exemplary”
RoadMarkingType_AttentionMarker	The road marking is an attention marker, for example, US, China and Japan.	“exemplary”
RoadMarkingType_Hatched	The road marking is hatched/chevron.	“exemplary”
RoadMarkingType_Box	The road marking of a junction.	“exemplary”
RoadMarkingType_ColouredArea	The road marking is a coloured area.	“exemplary”
RoadMarkingType_Arrow	The road marking is an arrow.	“exemplary”
RoadMarkingType_ZebraCrossing	The road marking is a zebra crossing/continental/ladder.	“exemplary”
RoadMarkingType_GenericSymbol	The road marking is a generic symbol.	“exemplary”
RoadMarkingType_TrafficSignOnLane	The road marking is a traffic sign.	“exemplary”
RoadMarkingType_GenericLine	The road marking is a generic line.	“exemplary”
RoadMarkingType_ParkingArea	The road marking is a parking area.	“exemplary”
RoadMarkingType_TShapeMarkingBegin	The road marking is a parking T-shape beginning parking line.	“exemplary”
RoadMarkingType_TShapeMarkingEnd	The road marking is a parking T-shape ending parking line.	“exemplary”
RoadMarkingType_IShapeMarkingBegin	The road marking is a parking I-shape beginning parking line.	“exemplary”
RoadMarkingType_IShapeMarkingEnd	The road marking is a parking I-shape ending parking line.	“exemplary”
RoadMarkingType_LShapeMarkingBegin	The road marking is a parking L-shape beginning parking line.	“exemplary”
RoadMarkingType_LShapeMarkingEnd	The road marking is a parking L-shape ending parking line.	“exemplary”

Table A.134 *(continued)*

Name	Description	RL enumerator
RoadMarkingType_Nets	The road marking is a net, specifically a non-stopping area.	"exemplary"

A.2.58 Road marking type – confidence

Table A.135 — Signal: Road marking type – confidence

Name	Road marking type – confidence		
Description	The signal "Road marking type – confidence" (A.2.58) provides the confidence of the classified signal "Road marking type" (A.2.57) information.		
Value type	[0...100] real value	**Unit**	%

A.2.59 Road object lane association

Table A.136 — Signal: Road object lane association

Name	Road object lane association		
Description	The signal "Road object lane association" (A.2.59) provides the association of a road marking or a road boundary to a lane with respect to the ego-vehicle lane.		
Value type	enumeration	**Unit**	1

Table A.137 — Enumeration: Road object lane association – Example enumerators

Name	Description	RL enumerator
RoadObjectLaneAssociation_EgoLane	The road marking is on the ego lane.	"exemplary"
RoadObjectLaneAssociation_EgoLeft1Lane	The road boundary separates the ego lane from 1st left neighbouring lane. The road marking is associated to the ego lane and the 1st left neighbouring lane.	"exemplary"
RoadObjectLaneAssociation_EgoRight1Lane	The road boundary separates the ego lane from 1st right neighbouring lane. The road marking is associated to the ego lane and the 1st right neighbouring lane.	"exemplary"
RoadObjectLaneAssociation_Left1Lane	The road marking is on the 1st left neighbouring lane.	"exemplary"
RoadObjectLaneAssociation_Right1Lane	The road marking is on the 1st right neighbouring lane.	"exemplary"
RoadObjectLaneAssociation_Left1Left2Lane	The road boundary separates the 1st left lane from the 2nd left neighbouring lane. The road marking is associated to the 1st left and the 2nd left neighbouring lanes.	"exemplary"
RoadObjectLaneAssociation_Right1Right2Lane	The road boundary separates the 1st right lane from the 2nd right neighbouring lane. The road marking is associated to the 1st right and the 2nd right neighbouring lanes.	"exemplary"
RoadObjectLaneAssociation_Left2Lane	The road marking is on the 2nd left neighbouring lane.	"exemplary"
RoadObjectLaneAssociation_Right2Lane	The road marking is on the 2nd right neighbouring lane.	"exemplary"
RoadObjectLaneAssociation_LeftRoadEdge	The road boundary limits at the outer edge of the leftmost lane.	"exemplary"

Table A.137 *(continued)*

Name	Description	RL enumerator
RoadObjectLaneAssociation_RightRoadEdge	The road boundary limits at the outer edge of the rightmost lane.	"exemplary"

A.2.60 Road object lane association – confidence

Table A.138 — Signal: Road object lane association – confidence

Name	Road object lane association – confidence		
Description	The signal "Road object lane association – confidence" (A.2.60) provides the confidence of the signal's "Road object lane association" (A.2.59) information.		
Value type	[0...100] real value	**Unit**	%

A.2.61 Arrow orientation {yaw}

Table A.139 — Signal: Arrow orientation

Name	Arrow orientation		
Description	The signal "Arrow orientation" (A.2.61) provides the orientation of the arrow symbol on the ground plane with respect to the driving direction of the ego-vehicle. Additional information: The signal "Arrow orientation" (A.2.61) := 0 rad, if the signal "Road marking type" (A.2.57) does not have the enumerator "RoadMarkingType_Arrow".		
Value type	1D vector real value	**Unit**	rad

A.2.62 Arrow direction

Table A.140 — Signal: Arrow direction

Name	Arrow direction		
Description	The signal "Arrow direction" (A.2.62) provides the estimated direction of the displayed arrow. Additional information: The enumerator "ArrowDirection_Foreward", which essentially means 0 rad, points in the direction of driving straight ahead. The signal "Arrow direction" (A.2.62) := "ArrowDirection_Foreward", if the signal "Road marking type" (A.2.57) does not have the enumerator "RoadMarkingType_Arrow".		
Value type	enumeration	**Unit**	1

Table A.141 — Enumeration: Arrow direction – Example enumerators

Name	Description	RL enumerator
ArrowDirection_Foreward	The arrow has an estimated direction of 0 rad (0°).	"exemplary"
ArrowDirection_Left	The arrow has an estimated direction of $+\frac{\pi}{2}$ rad (+90 °).	"exemplary"
ArrowDirection_Right	The arrow has an estimated direction of $-\frac{\pi}{2}$ rad (−90 °).	"exemplary"

Table A.141 *(continued)*

Name	Description	RL enumerator
ArrowDirection_StraightLeft	The arrow is straight left and has an estimated direction of $+\frac{\pi}{2}$ rad (+90 °).	"exemplary"
ArrowDirection_StraightRight	The arrow is straight right and has an estimated direction of $-\frac{\pi}{2}$ rad (−90 °).	"exemplary"
ArrowDirection_TurningPointLeft	The arrow has an estimated direction of $+\pi$ rad (+180 °).	"exemplary"
ArrowDirection_TurningPointRight	The arrow has an estimated direction of $-\pi$ rad (−180 °).	"exemplary"
ArrowDirection_45DegLeft	The arrow has an estimated direction of $+\frac{\pi}{4}$ rad (+45 °).	"exemplary"
ArrowDirection_45DegRight	The arrow has an estimated direction of $-\frac{\pi}{4}$ rad (−45 °).	"exemplary"
ArrowDirection_NoArrow	No arrow is present.	"exemplary"

A.2.63 Number of valid sign classifications

Table A.142 — Signal: Number of valid sign classifications

Name	Number of valid sign classifications		
Description	The signal "Number of valid sign classifications" (A.2.63) provides the current number of valid tuples for general landmark classifications. The tuples are defined by the signals "Sign classification type" (A.2.64) and "Sign classification type – confidence" (A.2.65).		
Value type	[0...] integer value	**Unit**	1

A.2.64 Sign classification type

Table A.143 — Signal: Sign classification type

Name	Sign classification type		
Description	The signal "Sign classification type" (A.2.64) provides the type of the sign as main traffic sign or road marking.		
Value type	enumeration	**Unit**	1

Table A.144 — Enumeration: Sign classification type – Example enumerators

Name	Description	RL enumerator
SignType_StopSign	The sign is a stop sign.	"exemplary"
SignType_YieldSign	The sign is a yield sign.	"exemplary"
SignType_SpeedLimitSign	The sign is a speed limit sign.	"exemplary"
SignType_NoMainSign	The sign is no main sign: it is only one or multiple supplementary signs.	"exemplary"
SignType_GreenArrowSign	The sign is a green arrow sign at traffic lights.	"exemplary"
SignType_HeightLimitSign	The sign is a height limit sign.	"exemplary"
SignType_EmptySign	The sign may be a changeable traffic sign without displaying a traffic sign symbol.	"exemplary"

A.2.65 Sign classification type – confidence

Table A.145 — Signal: Sign classification type – confidence

Name	Sign classification type – confidence		
Description	The signal "Sign classification type – confidence" (A.2.65) provides the confidence for the signal's "Sign classification type" (A.2.64) classification information, for example, the degree of certainty that the main traffic sign is of the reported type, for example, stop sign.		
Value type	[0...100] real value	**Unit**	%

A.2.66 Sign value

Table A.146 — Signal: Sign value

Name	Sign value		
Description	The signal "Sign value" (A.2.66) provides the depicted numerical value on the sign.		
Value type	real value	**Unit**	The unit is defined by the enumerator of the signal "Sign value unit" (A.2.67).

A.2.67 Sign value unit

Table A.147 — Signal: Sign value unit

Name	Sign value unit		
Description	The signal "Sign value unit" (A.2.67) provides the unit which the numerical value of the signal "Sign value" (A.2.66) is referring to.		
Value type	enumeration	**Unit**	1

Table A.148 — Enumeration: Sign value unit – Example enumerators

Name	**Description**	**RL enumerator**
SignValueUnit_KilometrePerHour	The unit for the sign value is kilometre per hour.	"exemplary"
SignValueUnit_MilePerHour	The unit for the sign value is mile per hour.	"exemplary"
SignValueUnit_Metre	The unit for the sign value is metre.	"exemplary"
SignValueUnit_Kilometre	The unit for the sign value is kilometre.	"exemplary"
SignValueUnit_Feet	The unit for the sign value is feet.	"exemplary"
SignValueUnit_Mile	The unit for the sign value is mile.	"exemplary"
SignValueUnit_MetricTon	The unit for the sign value is metric ton.	"exemplary"
SignValueUnit_LongTon	The unit for the sign value is long ton.	"exemplary"
SignValueUnit_ShortTon	The unit for the sign value is short ton.	"exemplary"
SignValueUnit_Minute	The unit for the sign value is minute.	"exemplary"
SignValueUnit_Hour	The unit for the sign value is hour.	"exemplary"
SignValueUnit_Day	The unit for the sign value is the day of the month.	"exemplary"
SignValueUnit_Weekday	The unit for the sign value is the weekday. The numeric value of the weekday is defined as: - Sunday (Sign value (A.2.66) := 0) - ... - Saturday (Sign value (A.2.66) := 6)	"exemplary"
SignValueUnit_Percentage	The unit for the sign value is percentage.	"exemplary"

A.2.68 Sign state

Table A.149 — Signal: Sign state

Name	Sign state		
Description	The signal "Sign state" (A.2.68) describes the state of the sign. It describes, for example, whether or not the message of the sign is variable/changeable, static or if that information is not available. It describes also, if the traffic sign is out of service, for example, the symbol is crossed out.		
Value type	enumeration	**Unit**	1

Table A.150 — Enumeration: Sign state – Example enumerators

Name	Description	RL enumerator
SignState_Static	The message sign is not a variable message sign.	"exemplary"
SignState_Variable	The message sign is a variable message sign.	"exemplary"
SignState_SwitchedOff	The message sign is a variable message sign which is switched off.	"exemplary"
SignState_FullOutOfService	The message sign is full out of service.	"exemplary"
SignState_PartlyOutOfService	Part of the message sign is out of service.	"exemplary"
SignState_OutOfView	The message sign has rotated.	"exemplary"

A.2.69 Colour value – object level

Table A.151 — Signal: Colour value – object level

Name	Colour value – object level		
Description	The signal "Colour value – object level" (A.2.69) provides the colour value(s). The number of values depends on the signal "Colour model type" (A.1.15). Additional information: For example, enumerator "ColourType_RGB" indicates 3 colour values, specifically first value red, second value green, third value blue.		
Value type	[0...100] real value	**Unit**	%

A.2.70 Colour tone – confidence – object level

Table A.152 — Signal: Colour tone – confidence – object level

Name	Colour tone – confidence – object level		
Description	The signal "Colour tone – confidence – object level" (A.2.70) provides the confidence of the colour tone. The colour tone is defined by the signal "Colour model type" (A.1.15) and the list of signals "Colour value – object level" (A.2.69).		
Value type	[0...100] real value	**Unit**	%

A.2.71 Number of valid connections

Table A.153 — Signal: Number of valid connections

Name	Number of valid connections		
Description	The signal "Number of valid connections" (A.2.71) provides the current number of valid tuples for connections. The tuples are defined by the signal "Connection type" (A.2.72) and unique signal's "Connection grouping ID" (A.2.73) value for each connection.		
Value type	[0...] integer value	**Unit**	1

A.2.72 Connection type

Table A.154 — Signal: Connection type

Name	Connection type		
Description	The signal “Connection type” (A.2.72) provides the type of connection of at least two road marking’s polylines or polynomials.		
Value type	enumeration	**Unit**	1

Table A.155 — Enumeration: Connection type – Example enumerators

Name	Description	RL enumerator
ConnectionType_Merge	The connection of road markings is a merge of road markings.	“exemplary”
ConnectionType_Split	The connection of road markings is a split of road markings.	“exemplary”
ConnectionType_Interconnection	The connection of road markings is an interconnection of road markings.	“exemplary”
ConnectionType_Extension	The connection of road markings is an extension of two road markings.	“exemplary”

A.2.73 Connection grouping ID

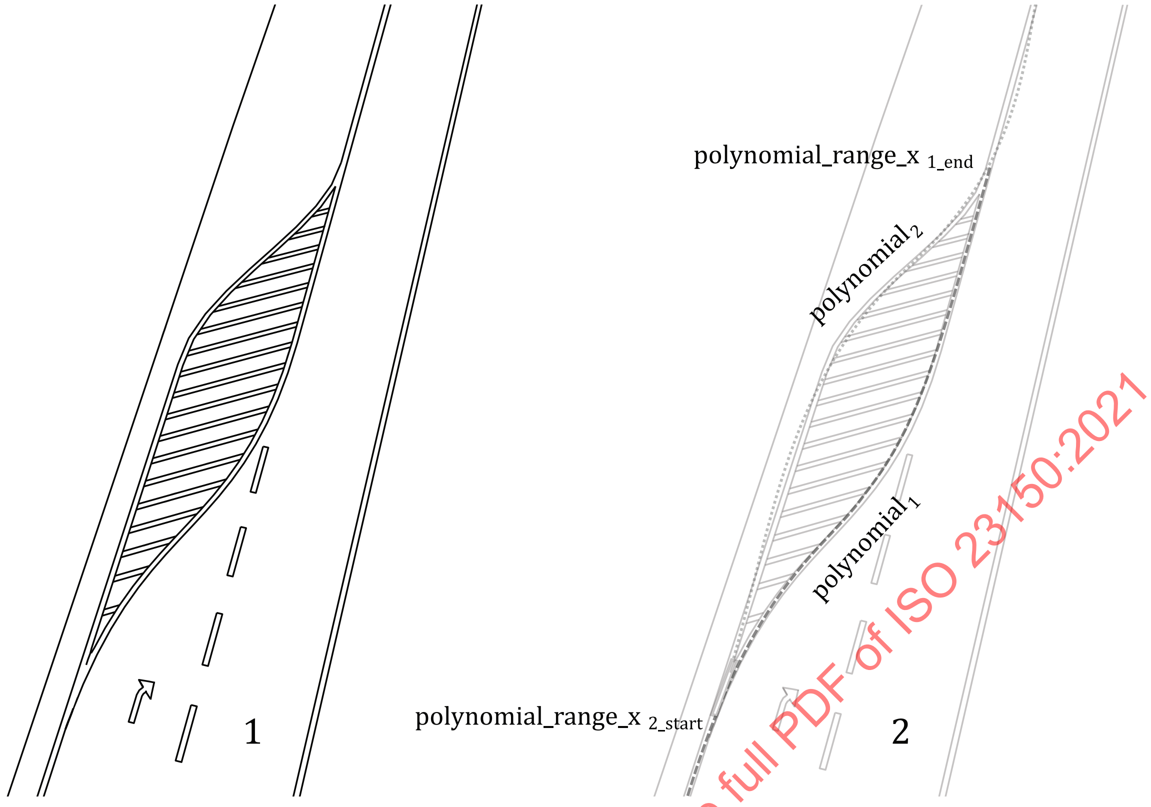
Table A.156 — Signal: Connection grouping ID

Name	Connection grouping ID		
Description	The signal “Connection grouping ID” (A.2.73) provides a unique grouping ID linking all connected road boundaries together. Additional information: The signal “Connection grouping ID” (A.2.73) shall be unique in each cycle of the sensor cluster.		
Value type	[...] integer value	**Unit**	1

A.2.74 Number of valid polynomials

Table A.157 — Signal: Number of valid polynomials

Name	Number of valid polynomials		
Description	The signal “Number of valid polynomials” (A.2.74) provides the current number of valid polynomials to describe the shape (see Figure A.6).		
Value type	[0...] integer value	**Unit**	1



Key

1 real-world scene

2 polynomial contour representation with 2 polynomials

Figure A.6 — Example for a polynomial contour representation

A.2.75 Polynomial coefficient y $\{c_0, c_1, c_2, c_3\}$

Table A.158 — Signal: Polynomial coefficient y $\{c_0, c_1, c_2, c_3\}$

Name	Polynomial coefficient y $\{c_0, c_1, c_2, c_3\}$		
Description	The signal "Polynomial coefficient y $\{c_0, c_1, c_2, c_3\}$" (A.2.75) provides the calculated coefficients for the *Y*-axis value of the polynomial line. Calculation method: $y = c_0 + c_1 \cdot x + c_2 \cdot x^2 + c_3 \cdot x^3$ where c_0: is the coefficient for degree 0, c_1: is the coefficient for degree 1, c_2: is the coefficient for degree 2, c_3: is the coefficient for degree 3 and x: is the value of the *X*-axis [see signal "Polynomial range x {begin, end}" (A.2.79)].		
Value type	4D vector real value	**Unit**	(m, 1, 1/m, 1/m^2)

A.2.76 Polynomial coefficient z {c_0, c_1, c_2, c_3}

Table A.159 — Signal: Polynomial coefficient z {c_0, c_1, c_2, c_3}

Name	Polynomial coefficient z {c_0, c_1, c_2, c_3}		
Description	The signal "Polynomial coefficient z {c_0, c_1, c_2, c_3}" (A.2.76) provides the calculated coefficients for the *Z*-axis value of the polynomial line. Calculation method: $z = c_0 + c_1 \cdot x + c_2 \cdot x^2 + c_3 \cdot x^3$ where c_0: is the coefficient for degree 0, c_1: is the coefficient for degree 1, c_2: is the coefficient for degree 2, c_3: is the coefficient for degree 3 and x: is the value of the *X*-axis [see signal "Polynomial range x {begin, end}" (A.2.79)].		
Value type	4D vector real value	**Unit**	(m, 1, 1/m, 1/m²)

A.2.77 Polynomial y – error

Table A.160 — Signal: Polynomial y – error

Name	Polynomial y – error		
Description	The signal "Polynomial y – error" (A.2.77) provides the error of the *Y*-axis value of the polynomial line. The unit and the error definition for the signal "Polynomial y – error" (A.2.77) is defined during the system design phase.		
Value type	scalar/vector/matrix real value [see subclause "Error model implementation" (B.4.1)]	**Unit**	see signal "Polynomial coefficient y {c_0, c_1, c_2, c_3}" (A.2.75)

A.2.78 Polynomial z – error

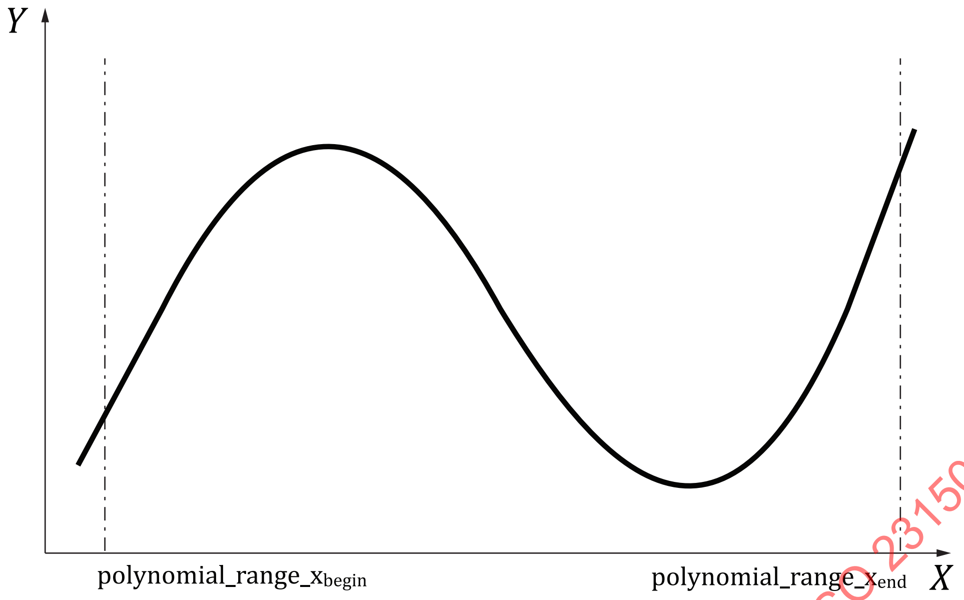
Table A.161 — Signal: Polynomial z – error

Name	Polynomial z – error		
Description	The signal "Polynomial z – error" (A.2.78) provides the error of the *Z*-axis value of the polynomial line. The unit and the error definition for the signal "Polynomial z – error" (A.2.78) is defined during the system design phase.		
Value type	scalar/vector/matrix real value [see subclause "Error model implementation" (B.4.1)]	**Unit**	see signal "Polynomial coefficient z {c_0, c_1, c_2, c_3}" (A.2.76)

A.2.79 Polynomial range x {begin, end}

Table A.162 — Signal: Polynomial range x {begin, end}

Name	Polynomial range x {begin, end}		
Description	The signal "Polynomial range x {begin, end}" (A.2.79) provides the valid range of the polynomials [x_{Begin}, x_{End}] (see Figure A.7).		
Value type	2D vector real value	**Unit**	(m, m)



Key

X-axis and *Y*-axis: vehicle axis system

Figure A.7 — Example for the signal "Polynomial range x {begin, end}" (A.2.79) for signal "Polynomial coefficient y {c_0, c_1, c_2, c_3}" (A.2.75)

A.2.80 Width – polynomial

Table A.163 — Signal: Width – polynomial

Name	Width – polynomial		
Description	The signal "Width – polynomial" (A.2.80) provides the measured width at the *X*-axis value (x_{Begin} + x_{End})/2. The width information is relative to the polynomial line which defines the centre of the width range.		
Value type	real value	**Unit**	m

A.2.81 Width – polynomial – error

Table A.164 — Signal: Width – polynomial – error

Name	Width – polynomial – error		
Description	The signal "Width – polynomial – error" (A.2.81) provides the error of the signal "Width – polynomial" (A.2.80).		
Value type	scalar/vector/matrix real value [see subclause "Error model implementation" (B.4.1)]	**Unit**	see signal "Width – polynomial" (A.2.80)

A.2.82 Width – polynomial – confidence

Table A.165 — Signal: Width – polynomial – confidence

Name	Width – polynomial – confidence		
Description	The signal "Width – polynomial – confidence" (A.2.82) provides the confidence of the width for the polynomial segment.		
Value type	[0...100] real value	**Unit**	%

A.2.83 Height – polynomial

Table A.166 — Signal: Height – polynomial

Name	Height – polynomial		
Description	The signal "Height – polynomial" (A.2.83) provides the measured height at the *X*-axis value (x_{Begin} + x_{End})/2. The height of the road object is added to the polynomial z value.		
Value type	real value	**Unit**	m

A.2.84 Height – polynomial – error

Table A.167 — Signal: Height – polynomial – error

Name	Height – polynomial – error		
Description	The signal "Height – polynomial – error" (A.2.84) provides the error of the signal "Height – polynomial" (A.2.83).		
Value type	scalar/vector/matrix real value [see subclause "Error model implementation" (B.4.1)]	**Unit**	see signal "Height – polynomial" (A.2.83)

A.2.85 Height – polynomial – confidence

Table A.168 — Signal: Height – polynomial – confidence

Name	Height – polynomial – confidence		
Description	The signal "Height – polynomial – confidence" (A.2.85) provides the confidence of the height for the polynomial segment.		
Value type	[0...100] real value	**Unit**	%

A.2.86 Number of valid data ranges

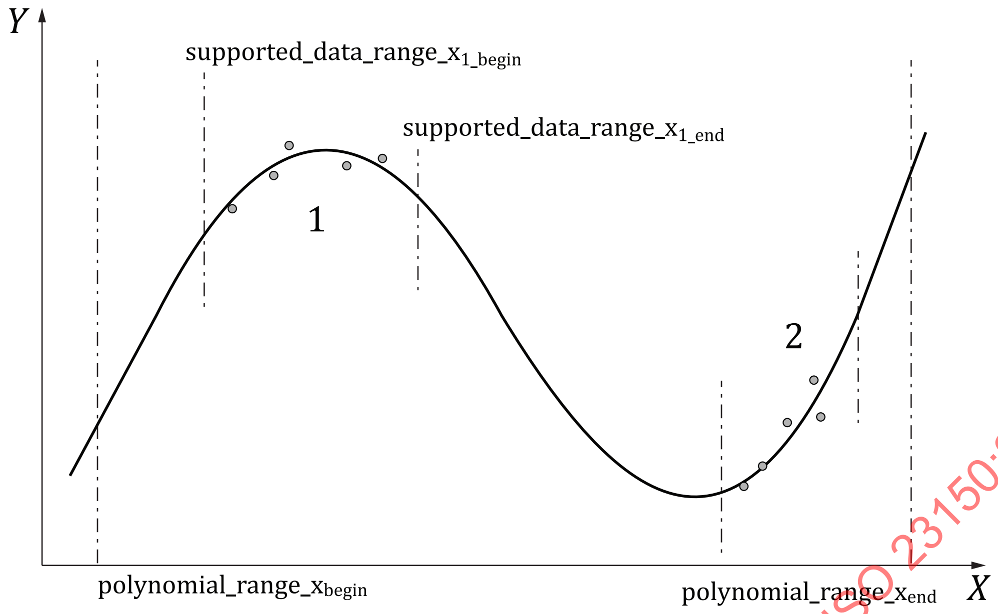
Table A.169 — Signal: Number of valid data ranges

Name	Number of valid data ranges		
Description	The signal "Number of valid data ranges" (A.2.86) provides the current number of valid *X*-axis data ranges for the polynomial lines.		
Value type	[0...] integer value	**Unit**	1

A.2.87 Supported data range x {begin, end}

Table A.170 — Signal: Supported data range x {begin, end}

Name	Supported data range x {begin, end}		
Description	The signal "Supported data range x {begin, end}" (A.2.87) provides the supported range of the polynomial [x_{Begin}, x_{End}] which is covered with measured points (see Figure A.8).		
Value type	2D vector real value	**Unit**	(m, m)



Key

X-axis and *Y*-axis: vehicle axis system

1 supported data range 1 with measured points

2 supported data range 2 with measured points

Figure A.8 — Example for the signal "Supported data range x {begin, end}" (A.2.87) for signal "Polynomial coefficient y {c_0, c_1, c_2, c_3}" (A.2.75)

A.2.88 Supported axis

Table A.171 — Signal: Supported axis

Name	Supported axis		
Description	The signal "Supported axis" (A.2.88) provides the information of the polynomial's supported axis for the signal "Supported data range x {begin, end}" (A.2.87).		
Value type	enumeration	**Unit**	1

Table A.172 — Enumeration: Polyline interpolation method – Example enumerators

Name	Description	RL enumerator
SupportedAxis_Y	The signal "Supported data range x {begin, end}" (A.2.87) corresponds to *Y*-axis polynomial line of signal "Polynomial coefficient y {c_0, c_1, c_2, c_3}" (A.2.75).	"exemplary"
SupportedAxis_Z	The signal "Supported data range x {begin, end}" (A.2.87) corresponds to *Z*-axis polynomial line of signal "Polynomial coefficient z {c_0, c_1, c_2, c_3}" (A.2.76).	"exemplary"
SupportedAxis_YAndZ	The signal "Supported data range x {begin, end}" (A.2.87) corresponds to both polynomial lines of signal "Polynomial coefficient y {c_0, c_1, c_2, c_3}" (A.2.75) and signal "Polynomial coefficient z {c_0, c_1, c_2, c_3}" (A.2.76).	"exemplary"

A.2.89 Polyline interpolation method

Table A.173 — Signal: Polyline interpolation method

Name	Polyline interpolation method

Table A.173 *(continued)*

Description	The signal "Polyline interpolation method" (A.2.89) provides the method for interpolating between the points of a polyline. Additional information: Interpolation method may be different for *Y*-axis and *Z*-axis values.		
Value type	enumeration	**Unit**	1

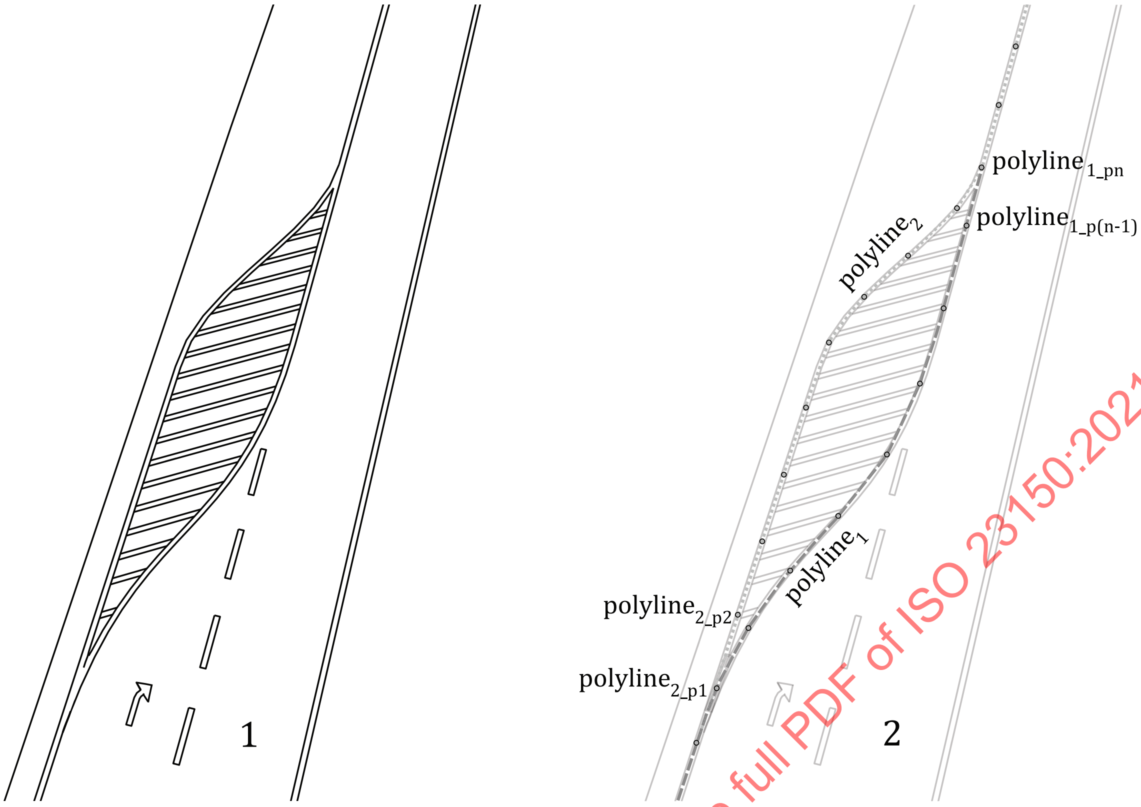
Table A.174 — Enumeration: Polyline interpolation method – Example enumerators

Name	Description	RL enumerator
PolylineInterpolationMethod_Linear	The polyline is interpolated with a linear interpolation between two sequential points.	"exemplary"
PolylineInterpolationMethod_Spline	The polyline is interpolated with a spline interpolation between two sequential points.	"exemplary"
PolylineInterpolationMethod_Cubic	The polyline is interpolated with a cubic interpolation between two sequential points.	"exemplary"

A.2.90 Number of valid polylines

Table A.175 — Signal: Number of valid polylines

Name	Number of valid polylines		
Description	The signal "Number of valid polylines" (A.2.90) provides the current number of valid polylines to describe a shape with multiple polylines (see Figure A.9).		
Value type	[0...] integer value	**Unit**	1



Key

1 real-world scene

2 polyline contour representation with 2 polylines and multiple vertices

Figure A.9 — Example for a polyline contour representation

A.2.91 Number of valid vertices

Table A.176 — Signal: Number of valid vertices

Name	Number of valid vertices		
Description	The signal "Number of valid vertices" (A.2.91) provides the current number of valid vertices of a polyline. A polyline describes a part of the shape border with one continuous segmented line (see Figure A.9) with at least 2 vertex points. Additional information: The signals "Width – vertex" (A.2.95) and "Height – vertex" (A.2.98) are attributes of the vertex at the point of the signal "Vertex point {x, y, z}" (A.2.92). The error and confidence signals apply to this point.		
Value type	[2...] integer value	**Unit**	1

A.2.92 Vertex point {x, y, z}

Table A.177 — Signal: Vertex point {x, y, z}

Name	Vertex point {x, y, z}		
Description	The signal "Vertex point {x, y, z}" (A.2.92) provides the measured longitudinal, lateral and vertical distance of the vertex point.		
Value type	3D vector real value	**Unit**	(m, m, m)

A.2.93 Vertex point {x, y, z} – error

Table A.178 — Signal: Vertex point {x, y, z} – error

Name	Vertex point {x, y, z} – error		
Description	The signal “Vertex point {x, y, z} – error” (A.2.93) provides the measured error of the signal “Vertex point {x, y, z}” (A.2.92).		
Value type	scalar/vector/matrix real value [see subclause “Error model implementation” (B.4.1)]	**Unit**	see signal “Vertex point {x, y, z}” (A.2.92)

A.2.94 Vertex point – confidence {x, y, z}

Table A.179 — Signal: Vertex point – confidence {x, y, z}

Name	Vertex point – confidence {x, y, z}		
Description	The signal “Vertex point – confidence {x, y, z}” (A.2.94) provides the confidence of the measured longitudinal, lateral and vertical distance of the vertex point.		
Value type	[0...100] 3D vector real value	**Unit**	(%, %, %)

A.2.95 Width – vertex

Table A.180 — Signal: Width – vertex

Name	Width – vertex		
Description	The signal “Width – vertex” (A.2.95) provides the measured width of the vertex at the vertex point of the signal “Vertex point {x, y, z}” (A.2.92). The width information is relative to the vertex which defines the centre of the width range. Additional information: The signal applies to the point of the signal “Vertex point {x, y, z}” (A.2.92).		
Value type	real value	**Unit**	m

A.2.96 Width – vertex – error

Table A.181 — Signal: Width – vertex – error

Name	Width – vertex – error		
Description	The signal “Width – vertex – error” (A.2.96) provides the error of the measured width at the vertex point [see signal “Vertex point {x, y, z}” (A.2.92)]. Additional information: The signal applies to the point of the signal “Vertex point {x, y, z}” (A.2.92).		
Value type	scalar/vector/matrix real value [see subclause “Error model implementation” (B.4.1)]	**Unit**	see signal “Width – vertex” (A.2.95)

A.2.97 Width – vertex – confidence

Table A.182 — Signal: Width – vertex – confidence

Name	Width – vertex – confidence

Table A.182 *(continued)*

Description	The signal "Width – vertex – confidence" (A.2.97) provides the confidence of the width of the vertex. Additional information: The signal applies to the point of the signal "Vertex point {x, y, z}" (A.2.92).		
Value type	[0...100] real value	**Unit**	%

A.2.98 Height – vertex

Table A.183 — Signal: Height – vertex

Name	Height – vertex		
Description	The signal "Height – vertex" (A.2.98) provides the measured height of the vertex at the vertex point of the signal "Vertex point {x, y, z}" (A.2.92). The height of the road object is added to the vertex point z value. Additional information: The signal applies to the point of the signal "Vertex point {x, y, z}" (A.2.92).		
Value type	real value	**Unit**	m

A.2.99 Height – vertex – error

Table A.184 — Signal: Height – vertex – error

Name	Height – vertex – error		
Description	The signal "Height – vertex – error" (A.2.99) provides the error of the measured height. Additional information: The signal applies to the point of the signal "Vertex point {x, y, z}" (A.2.92).		
Value type	scalar/vector/matrix real value [see subclause "Error model implementation" (B.4.1)]	**Unit**	see signal "Height – vertex" (A.2.98)

A.2.100 Height – vertex – confidence

Table A.185 — Signal: Height – vertex – confidence

Name	Height – vertex – confidence		
Description	The signal "Height – vertex – confidence" (A.2.100) provides the confidence of the vertex's height. Additional information: The signal applies to the point of the signal "Vertex point {x, y, z}" (A.2.92).		
Value type	[0...100] real value	**Unit**	%

A.2.101 Number of valid road boundary classifications

Table A.186 — Signal: Number of valid road boundary classifications

Name	Number of valid road boundary classifications

Table A.186 *(continued)*

Description	The signal "Number of valid road boundary classifications" (A.2.101) provides the current number of valid tuples for road boundary types. The tuples are defined by the signals "Road boundary type" (A.2.102), "Road boundary type – confidence" (A.2.103) and additional signals.		
Value type	[0...] integer value	**Unit**	1

A.2.102 Road boundary type

Table A.187 — Signal: Road boundary type

Name	Road boundary type		
Description	The signal "Road boundary type" (A.2.102) indicates the type of the road boundary.		
Value type	enumeration	**Unit**	1

Table A.188 — Enumeration: Road boundary type – Example enumerators

Name	Description	RL enumerator
RoadBoundaryType_GuardRail	The road boundary is a guard rail.	"exemplary"
RoadBoundaryType_Fence	The road boundary is a fence.	"exemplary"
RoadBoundaryType_Wall	The road boundary is a wall, for example, brick wall, building.	"exemplary"
RoadBoundaryType_Barrier	The road boundary is a barrier.	"exemplary"
RoadBoundaryType_TensionCableSystem	The road boundary is a tension cable system.	"exemplary"
RoadBoundaryType_RoadEdge	The road boundary is a road edge, for example, grass, vegetation, sand, gravel, soil.	"exemplary"
RoadBoundaryType_Curb	The road boundary is a curb stone.	"exemplary"

A.2.103 Road boundary type – confidence

Table A.189 — Signal: Road boundary type – confidence

Name	Road boundary type – confidence		
Description	The signal "Road boundary type – confidence" (A.2.103) provides the confidence of the classified signal "Road boundary type" (A.2.102).		
Value type	[0...100] real value	**Unit**	%

A.2.104 Number of valid general landmark classifications

Table A.190 — Signal: Number of valid general landmark classifications

Name	Number of valid general landmark classifications		
Description	The signal "Number of valid general landmark classifications" (A.2.104) provides the current number of valid tuples for general landmark classifications. The tuples are defined by the signals "General landmark classification type" (A.2.105) and "General landmark classification type – confidence" (A.2.106).		
Value type	[0...] integer value	**Unit**	1

A.2.105 General landmark classification type

Table A.191 — Signal: General landmark classification type

Name	General landmark classification type

Table A.191 *(continued)*

Description	The signal “General landmark classification type” (A.2.105) provides the classification of the general landmark.		
Value type	enumeration	**Unit**	1

Table A.192 — Enumeration: General landmark classification type – Example enumerators

Name	**Description**	**RL enumerator**
GeneralLandmarkType_Bridge	The general landmark is a bridge.	“exemplary”
GeneralLandmarkType_Beacon	The general landmark is a beacon.	“exemplary”
GeneralLandmarkType_Cone	The general landmark is a cone.	“exemplary”
GeneralLandmarkType_Barrel	The general landmark is a barrel.	“exemplary”
GeneralLandmarkType_GuidePost	The general landmark is a guide post.	“exemplary”
GeneralLandmarkType_LampPost	The general landmark is a lamp post.	“exemplary”
GeneralLandmarkType_VerticalStructure	The general landmark is a vertical structure.	“exemplary”
GeneralLandmarkType_OverheadObject	The general landmark is an overhead object.	“exemplary”
GeneralLandmarkType_RectangularStructure	The general landmark is a rectangular structure.	“exemplary”
GeneralLandmarkType_Tunnel	The general landmark is a tunnel.	“exemplary”
GeneralLandmarkType_Reflector	The general landmark is a reflector.	“exemplary”
GeneralLandmarkType_OverridableObject	The general landmark is an overridable object.	“exemplary”

A.2.106 General landmark classification type – confidence

Table A.193 — Signal: General landmark classification type – confidence

Name	General landmark classification type – confidence		
Description	The signal “General landmark classification type – confidence” (A.2.106) provides the probability for the corresponding signal “General landmark classification type” (A.2.105).		
Value type	[0...100] real value	**Unit**	%

A.2.107 Sign geometry

Table A.194 — Signal: Sign geometry

Name	Sign geometry		
Description	The signal “Sign geometry” (A.2.107) provides the shape geometry of the sign.		
Value type	enumeration	**Unit**	1

Table A.195 — Enumeration: Sign geometry – Example enumerators

Name	**Description**	**RL enumerator**
SignGeometry_Circle	The sign has a circle shape as sign geometry.	“exemplary”
SignGeometry_TriangleTop	The sign has a triangle shape with tip pointing downwards as sign geometry.	“exemplary”
SignGeometry_TriangleDown	The sign has a triangle shape with tip pointing upwards as sign geometry.	“exemplary”
SignGeometry_Square	The sign has a square shape as sign geometry.	“exemplary”
SignGeometry_Pole	The sign has a pole shape as sign geometry.	“exemplary”
SignGeometry_Rectangle	The sign has a rectangle shape as sign geometry.	“exemplary”
SignGeometry_Diamond	The sign has a diamond shape as sign geometry.	“exemplary”
SignGeometry_ArrowLeft	The sign has an arrow left, five edge shape as sign geometry.	“exemplary”

Table A.195 *(continued)*

Name	Description	RL enumerator
SignGeometry_ArrowRight	The sign has an arrow right, five edge shape as sign geometry.	“exemplary”
SignGeometry_Octagon	The sign has an octagon shape as sign geometry.	“exemplary”
SignGeometry_SaintAndrewsCross	The sign has a cross shape as sign geometry.	“exemplary”
SignGeometry_GateShape	The sign has a gate shape as sign geometry.	“exemplary”

A.2.108 Number of valid lane relevance classifications

Table A.196 — Signal: Number of valid lane relevance classifications

Name	Number of valid lane relevance classifications		
Description	The signal “Number of valid lane relevance classifications” (A.2.108) provides the current number of valid tuples for lane relevance classifications. The tuples are defined by the signals “Lane relevance classification type” (A.2.109) and “Lane relevance classification type – confidence” (A.2.110).		
Value type	[0...] integer value	**Unit**	1

A.2.109 Lane relevance classification type

Table A.197 — Signal: Lane relevance classification type

Name	Lane relevance classification type		
Description	The signal “Lane relevance classification type” (A.2.109) provides the information regarding if the sign is relevant for the ego-vehicle’s lane, the nearest lane to the ego-vehicle or other relevant lanes.		
Value type	enumeration	**Unit**	1

Table A.198 — Enumeration: Lane relevance classification type – Example enumerators

Name	Description	RL enumerator
LaneRelevanceType_EgoLane	The entity is relevant on track of ego-vehicle.	“exemplary”
LaneRelevanceType_Left1Lane	The entity is relevant for the next lane to the ego-vehicle on the left side.	“exemplary”
LaneRelevanceType_Right1Lane	The entity is relevant for the next lane to the ego-vehicle on the right side.	“exemplary”
LaneRelevanceType_Left2Lane	The entity is relevant for the second next lane to the ego-vehicle on the left side.	“exemplary”
LaneRelevanceType_Right2Lane	The entity is relevant for the second next lane to the ego-vehicle on the right side.	“exemplary”
LaneRelevanceType_EgoAndLeft1Lane	The entity is relevant on track of ego-vehicle and the next left lane.	“exemplary”
LaneRelevanceType_EgoAndRight1Lane	The entity is relevant on track of ego-vehicle and the next right lane.	“exemplary”
LaneRelevanceType_LeftmostLane	The entity is relevant for the leftmost lane.	“exemplary”
LaneRelevanceType_RightmostLane	The entity is relevant for the rightmost lane.	“exemplary”
LaneRelevanceType_AllLanes	The entity is relevant for all lanes, which essentially means the lane(s) to the right- and left side and on track.	“exemplary”
LaneRelevanceType_OtherLane	The entity is relevant for another far lane.	“exemplary”

A.2.110 Lane relevance classification type – confidence

Table A.199 — Signal: Lane relevance classification type – confidence

Name	Lane relevance classification type – confidence		
Description	The signal "Lane relevance classification type – confidence" (A.2.110) provides the confidence for the signal's "Lane relevance classification type" (A.2.109) classification information, for example, the degree of certainty that the lane relevance is of the reported type, for example, relevant for all lanes.		
Value type	[0...100] real value	**Unit**	%

A.2.111 Number of valid traffic supplementary signs

Table A.200 — Signal: Number of valid traffic supplementary signs

Name	Number of valid traffic supplementary signs		
Description	The signal "Number of valid traffic supplementary signs" (A.2.111) provides the current number of valid supplementary sign entities. Additional information: The entries in the array may or may not be sorted.		
Value type	[0...] integer value	**Unit**	1

A.2.112 Number of valid supplementary sign classifications

Table A.201 — Signal: Number of valid supplementary sign classifications

Name	Number of valid supplementary sign classifications		
Description	The signal "Number of valid supplementary sign classifications" (A.2.112) provides the current number of valid tuples for supplementary sign classifications. The tuples are defined by the signals "Supplementary sign classification type" (A.2.113) and "Supplementary sign classification type – confidence" (A.2.114).		
Value type	[0...] integer value	**Unit**	1

A.2.113 Supplementary sign classification type

Table A.202 — Signal: Supplementary sign classification type

Name	Supplementary sign classification type		
Description	The signal "Supplementary sign classification type" (A.2.113) provides the type of the sign.		
Value type	enumeration	**Unit**	1

Table A.203 — Enumeration: Supplementary sign classification type – Example enumerators

Name	Description	RL enumerator
SupplementarySignType_ValidInformationBegin	The sign displays a begin of valid zone.	"exemplary"
SupplementarySignType_ValidInformationEnd	The sign displays an end of valid zone.	"exemplary"
SupplementarySignType_Frost	The sign displays a frost sign.	"exemplary"
SupplementarySignType_WetRoad	The sign displays a wet road sign.	"exemplary"
SupplementarySignType_ValidInDistance	The sign displays a distance information and the main sign becomes valid in the defined distance.	"exemplary"

Table A.203 *(continued)*

Name	Description	RL enumerator
SupplementarySignType_ValidForDistance	The sign displays a distance information and the main sign is valid for the defined distance.	“exemplary”
SupplementarySignType_Limitation	The sign displays a limitation information.	“exemplary”

A.2.114 Supplementary sign classification type – confidence

Table A.204 — Signal: Supplementary sign classification type – confidence

Name	Supplementary sign classification type – confidence		
Description	The signal “Supplementary sign classification type – confidence” (A.2.114) provides the confidence for the signal’s “Supplementary sign classification type” (A.2.113) classification information, for example, the degree of certainty that the supplementary traffic sign is of the reported type, for example, limitation information.		
Value type	[0...100] real value	**Unit**	%

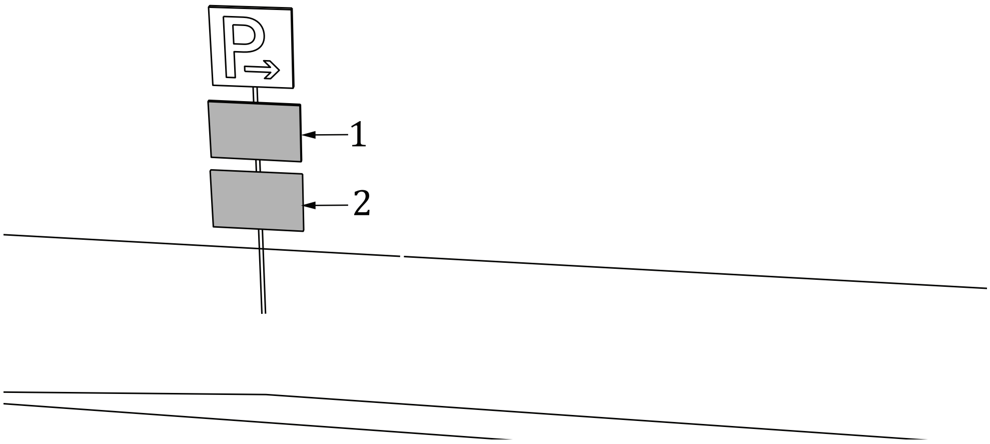
A.2.115 Relative position

Table A.205 — Signal: Relative position

Name	Relative position		
Description	The signal “Relative position” (A.2.115) provides the relative position of the supplemental sign with respect to its main sign (see Figure A.10).		
Value type	enumeration	**Unit**	1

Table A.206 — Enumeration: Relative position – Example enumerators

Name	Description	RL enumerator
RelativePosition_Above	The supplementary sign is above the main sign.	“exemplary”
RelativePosition_Below	The supplementary sign is below the main sign.	“exemplary”
RelativePosition_Left	The supplementary sign is left of the main sign.	“exemplary”
RelativePosition_Right	The supplementary sign is right of the main sign.	“exemplary”



Key

1 supplementary sign entity with signal “Relative position” (A.2.115) := enumerator “RelativePosition_Below” and signal “Relative position order” (A.2.116) := 1

2 supplementary sign entity with signal “Relative position” (A.2.115) := enumerator “RelativePosition_Below” and signal “Relative position order” (A.2.116) := 2

Figure A.10 — Example for a main traffic sign with two supplementary signs

A.2.116 Relative position order

Table A.207 — Signal: Relative position order

Name	Relative position order		
Description	The signal “Relative position order” (A.2.116) provides the relative order of a supplementary sign with respect to the main sign position. The value 1 defines the nearest position of the supplementary sign to the main sign (see Figure A.10).		
Value type	integer value	**Unit**	1

A.2.117 Number of valid structure light classifications

Table A.208 — Signal: Number of valid structure light classifications

Name	Number of valid structure light classifications		
Description	The signal “Number of valid structure light classifications” (A.2.117) provides the current number of valid tuples for structure classifications of the traffic light. The tuples are defined by the signals “Structure light classification type” (A.2.118) and “Structure light classification type – confidence” (A.2.119).		
Value type	[0...] integer value	**Unit**	1

A.2.118 Structure light classification type

Table A.209 — Signal: Structure light classification type

Name	Structure light classification type		
Description	The signal “Structure light classification type” (A.2.118) provides the classification of the traffic light’s shape.		
Value type	enumeration	**Unit**	1

Table A.210 — Enumeration: Structure light classification type – Example enumerators

Name	Description	RL enumerator
StructureLightType_Vertical3	The traffic light is composed of three vertical light spots.	“exemplary”
StructureLightType_Horizontal3	The traffic light is composed of three horizontal light spots.	“exemplary”
StructureLightType_DogHouse	The traffic light is composed of multiple light spots.	“exemplary”

A.2.119 Structure light classification type – confidence

Table A.211 — Signal: Structure light classification type – confidence

Name	Structure light classification type – confidence		
Description	The signal “Structure light classification type – confidence” (A.2.119) provides the confidence for the signal’s “Structure light classification type” (A.2.118) classification information, for example, the degree of certainty that the structure light type is of the reported type, for example, a traffic light with 3 vertical stacked traffic light spots.		
Value type	[0...100] real value	**Unit**	%

A.2.120 Minimum visibility distance

Table A.212 — Signal: Minimum visibility distance

Name	Minimum visibility distance		
Description	The signal “Minimum visibility distance” (A.2.120) provides the minimum distance to the traffic light until the traffic light is still in the FOV of the sensor cluster.		
Value type	[0...] real value	**Unit**	m

A.2.121 Total number of traffic light spots

Table A.213 — Signal: Total number of traffic light spots

Name	Total number of traffic light spots		
Description	The signal “Total number of traffic light spots” (A.2.121) provides the estimation of the number of traffic light spots which are part of the traffic light. Additional information: This signal is considered to be an estimation since turned-off spots are barely visible (especially at night) and therefore hard to detect. Signal “Total number of traffic light spots” (A.2.121) >= signal “Number of valid traffic light spots” (A.2.123).		
Value type	[0...] integer value	**Unit**	1

A.2.122 Total number of traffic light spots – confidence

Table A.214 — Signal: Total number of traffic light spots – confidence

Name	Total number of traffic light spots – confidence		
Description	The signal “Total number of traffic light spots – confidence” (A.2.122) provides the confidence for the signal’s “Total number of traffic light spots” (A.2.121) classification information, for example, the degree of certainty that the number of traffic light spots is of the reported type, for example, 3.		
Value type	[0...100] real value	**Unit**	%

A.2.123 Number of valid traffic light spots

Table A.215 — Signal: Number of valid traffic light spots

Name	Number of valid traffic light spots		
Description	The signal "Number of valid traffic light spots" (A.2.123) provides the current number of valid traffic light entities. Additional information: The entries in the array may or may not be sorted.		
Value type	[0...] integer value	**Unit**	1

A.2.124 Number of valid light shape classifications

Table A.216 — Signal: Number of valid light shape classifications

Name	Number of valid light mode classifications		
Description	The signal "Number of valid light shape classifications" (A.2.124) provides the current number of valid tuples for light shape classifications. The tuples are defined by the signals "Light shape classification type" (A.2.125) and "Light shape classification type – confidence" (A.2.126).		
Value type	[0...] integer value	**Unit**	1

A.2.125 Light shape classification type

Table A.217 — Signal: Light shape classification type

Name	Light shape classification type		
Description	The signal "Light shape classification type" (A.2.125) provides the light's shape which is displayed by the light spot.		
Value type	enumeration	**Unit**	1

Table A.218 — Enumeration: Light shape classification type – Example enumerators

Name	Description	RL enumerator
LightShapeType_NoShape	The traffic light spot displays no additional shape.	"exemplary"
LightShapeType_ArrowStraightAhead	The traffic light spot displays an arrow straight ahead shape.	"exemplary"
LightShapeType_ArrowLeft	The traffic light spot displays an arrow left shape.	"exemplary"
LightShapeType_ArrowDiagLeft	The traffic light spot displays an arrow diagonal left shape.	"exemplary"
LightShapeType_ArrowStraightAheadLeft	The traffic light spot displays an arrow straight ahead and arrow left shape.	"exemplary"
LightShapeType_ArrowRight	The traffic light spot displays an arrow right shape.	"exemplary"
LightShapeType_ArrowDiagRight	The traffic light spot displays an arrow diagonal right shape.	"exemplary"
LightShapeType_ArrowStraightAheadRight	The traffic light spot displays an arrow straight ahead and arrow right shape.	"exemplary"
LightShapeType_ArrowLeftRight	The traffic light spot displays an arrow left and arrow right shape.	"exemplary"
LightShapeType_ArrowDown	The traffic light spot displays an arrow down shape.	"exemplary"
LightShapeType_ArrowDownLeft	The traffic light spot displays an arrow U-turn left shape.	"exemplary"

Table A.218 *(continued)*

Name	Description	RL enumerator
LightShapeType_ArrowDownRight	The traffic light spot displays an arrow U-turn right shape.	"exemplary"
LightShapeType_Cross	The traffic light spot displays a cross figure.	"exemplary"
LightShapeType_Pedestrian	The traffic light spot displays a pedestrian figure.	"exemplary"
LightShapeType_Walk	The traffic light spot displays a text "walk" figure.	"exemplary"
LightShapeType_DontWalk	The traffic light spot displays a text "don't walk" figure.	"exemplary"
LightShapeType_Bicycle	The traffic light spot displays a bicycle figure.	"exemplary"
LightShapeType_PedestrianAndBicycle	The traffic light spot displays a pedestrian and bicycle figure.	"exemplary"
LightShapeType_CountdownSecond	The traffic light spot displays a countdown in seconds figure, the signal "Light shape value" (A.2.127) contains the value in s.	"exemplary"
LightShapeType_CountdownPercent	The traffic light spot displays a countdown in percent figure, the signal "Light shape value" (A.2.127) contains the value in %.	"exemplary"
LightShapeType_Train	The traffic light spot displays a train or tram figure.	"exemplary"
LightShapeType_Bus	The traffic light spot displays a bus figure.	"exemplary"
LightShapeType_BusAndTrain	The traffic light spot displays a bus and train or tram figure.	"exemplary"

A.2.126 Light shape classification type – confidence

Table A.219 — Signal: Light shape classification type – confidence

Name	Light shape classification type – confidence		
Description	The signal "Light shape classification type – confidence" (A.2.126) provides the confidence for the signal's "Light shape classification type" (A.2.125) classification information, for example, the degree of certainty that the light shape is of the reported type, for example, the shape of a bus.		
Value type	[0...100] real value	**Unit**	%

A.2.127 Light shape value

Table A.220 — Signal: Light shape value

Name	Light shape value		
Description	The signal "Light shape value" (A.2.127) provides an additional countdown value of the light shape which is defined by the signal "Light shape classification type" (A.2.125). Additional information: The value meaning and the value unit depends on the enumerator of the signal "Light shape classification type" (A.2.125).		
Value type	[0 ...] integer value	**Unit**	The unit is defined by the enumerator of the signal "Light shape classification type" (A.2.125). * The signal "Light shape classification type" (A.2.125) has an enumerator "LightShapeType_CountdownSecond", an enumerator "LightShapeType_CountdownPercent" or a similar enumerator defined during the system desing phase.

A.2.128 Number of valid colour classifications

Table A.221 — Signal: Number of valid colour classifications

Name	Number of valid colour classifications		
Description	The signal "Number of valid colour classifications" (A.2.128) provides the current number of valid tuples for colour classifications. The tuples are defined by the signals "Colour classification type" (A.2.129) and "Colour classification type – confidence" (A.2.130).		
Value type	[0...] integer value	**Unit**	1

A.2.129 Colour classification type

Table A.222 — Signal: Colour classification type

Name	Colour classification type		
Description	The signal "Colour classification type" (A.2.129) provides the colour of the light spot.		
Value type	enumeration	**Unit**	1

Table A.223 — Enumeration: Colour classification type – Example enumerators

Name	**Description**	**RL enumerator**
ColourType_Red	The light spot colour is red.	"exemplary"
ColourType_Yellow	The light spot colour is yellow.	"exemplary"
ColourType_Green	The light spot colour is green.	"exemplary"
ColourType_White	The light spot colour is white.	"exemplary"

A.2.130 Colour classification type – confidence

Table A.224 — Signal: Colour classification type – confidence

Name	Colour classification type – confidence		
Description	The signal "Colour classification type – confidence" (A.2.130) provides a probability for the corresponding signal "Colour classification type" (A.2.129).		
Value type	[0...100] real value	**Unit**	%

A.2.131 Number of valid light mode classifications

Table A.225 — Signal: Number of valid light mode classifications

Name	Number of valid light mode classifications		
Description	The signal "Number of valid light mode classifications" (A.2.131) provides the current number of valid tuples for light mode classifications. The tuples are defined by the signals "Light mode classification type" (A.2.132) and "Light mode classification type – confidence" (A.2.133).		
Value type	[0...] integer value	**Unit**	1

A.2.132 Light mode classification type

Table A.226 — Signal: Light mode classification type

Name	Light mode classification type		
Description	The signal "Light mode classification type" (A.2.132) provides the light's mode.		
Value type	enumeration	**Unit**	1

Table A.227 — Enumeration: Light mode classification type – Example enumerators

Name	Description	RL enumerator
LightModeType_Continuous	The light source is continuously on.	“exemplary”
LightModeType_Blinking	One light source is visibly blinking.	“exemplary”
LightModeType_TurnedOff	The light source is turned off.	“exemplary”
LightModeType_Counting	It is a light source with counting.	“exemplary”

A.2.133 Light mode classification type – confidence

Table A.228 — Signal: Light mode classification type – confidence

Name	Light mode classification type – confidence		
Description	The signal “Light mode classification type – confidence” (A.2.133) provides a confidence for the corresponding signal “Light mode classification type” (A.2.132).		
Value type	[0...100] real value	**Unit**	%

A.3 Feature level entity signals

Signals are defined for the entities of the subclause “Camera feature interface” (8.3) and the subclause “Ultrasonic feature interface” (8.4) in this subclause (see Tables A.229 - A.273). These interfaces are located at FLI. No FLI for radar or lidar sensors is defined.

The FLI originate from several sensors of a sensor cluster. Therefore, at feature level, the term sensor cluster is always used, even if a single sensor is serving the interface.

A.3.1 Existence probability – feature level

Table A.229 — Signal: Existence probability – feature level

Name	Existence probability – feature level		
Description	The signal “Existence probability – feature level” (A.3.1) provides the existence probability of the feature, based on history. Additional information: Use as confidence measure where a low value means less confidence and a high value indicates strong confidence.		
Value type	[0...100] real value	**Unit**	%

A.3.2 Feature ID

Table A.230 — Signal: Feature ID

Name	Feature ID		
Description	The signal “Feature ID” (A.3.2) provides the unique identification number of the feature within the feature list. Additional information: The signal’s “Feature ID” (A.3.2) value is only unique for each sensor cluster's feature entities of FLI within one measurement cycle.		
Value type	[0...] integer value	**Unit**	1

A.3.3 Object ID reference – feature level

Table A.231 — Signal: Object ID reference – feature level

Name	Object ID reference – feature level		
Description	The signal's "Object ID" (A.2.2) value of the object to which this feature is associated to; invalid = "no associated object", for example, first feature cycle. Additional information: Refers to an object (for example, potentially moving objects) with the highest probability for the association. The signal's "Object ID reference – feature level" (A.3.3) value could reference entities of the sensor cluster's OLI or entities which are used internally by the sensor cluster. Optimisation (B.3.2): the mapping of objects signal "Object ID" (A.2.2) and features signal "Object ID reference – feature level" (A.3.3) could be provided in a separate interface.		
Value type	[0...] integer value	**Unit**	1

A.3.4 Time stamp difference – feature level

Table A.232 — Signal: Time stamp difference – feature level

Name	Time stamp difference – feature level		
Description	Each feature entity could provide an individual time stamp, specifically a time stamp difference to the signal's "Time stamp – predicted" (A.1.5.1) value of the interface. Additional information: The time stamp of the feature can be calculated as the signal "Time stamp – predicted" (A.1.5.1) of the FLI header plus the signal "Time stamp difference – feature level" (A.3.4) of the feature.		
Value type	real value	**Unit**	s

A.3.5 Number of valid observations – feature level

Table A.233 — Signal: Number of valid observations – feature level

Name	Number of valid observations – feature level		
Description	The signal "Number of valid observations – feature level" (A.3.5) provides the current number of valid tuples [specifically the signal "Time stamp reference – feature level" (A.3.6) and the signal "Observation status – feature level" (A.3.7)].		
Value type	[0...] integer value	**Unit**	1

A.3.6 Time stamp reference – feature level

Table A.234 — Signal: Time stamp reference – feature level

Name	Time stamp reference – feature level		
Description	The signal "Time stamp reference – feature level" (A.3.6) provides a reference to a previous sent interface message with the referenced time stamp for signal "Time stamp – <...>" (A.1.5).		
Value type	real value	**Unit**	s

A.3.7 Observation status – feature level

Table A.235 — Signal: Observation status – feature level

Name	Observation status – feature level		
Description	The signal "Observation status – feature level" (A.3.7) provides the observation status of the feature, which was recognised in a previous cycle [see signal "Time stamp reference – feature level" (A.3.6)].		
Value type	enumeration	**Unit**	1

Table A.236 — Enumeration: Observation status – feature level – Example enumerators

Name	Description	RL enumerator
Observation_True	The feature was observed in the cycle.	"exemplary"
Observation_False	The feature was not observed in the cycle. It may be predicted in the current cycle.	"exemplary"

A.3.8 Feature grouping ID

Table A.237 — Signal: Feature grouping ID

Name	Feature grouping ID		
Description	A sensor can group features from the FLI which are linked together. All features from the same entity have the same signal's "Feature grouping ID" (A.3.8) value. The ID is unique considering all FLIs of a sensor cluster. EXAMPLE A pedestrian may be described with multiple camera shapes. Additional information: The signal "Feature grouping ID" (A.3.8) := 0 if the feature is not associated with another feature.		
Value type	integer value	**Unit**	1

A.3.9 Number of valid shape classifications – feature level

Table A.238 — Signal: Number of valid shape classifications – feature level

Name	Number of valid shape classifications – feature level		
Description	The signal "Number of valid shape classifications – feature level" (A.3.9) provides the current number of valid tuples for shape classifications. The tuples are defined by the signals "Shape classification type – feature level" (A.3.10) and "Shape classification type – confidence – feature level" (A.3.11).		
Value type	[0...] integer value	**Unit**	1

A.3.10 Shape classification type – feature level

Table A.239 — Signal: Shape classification type – feature level

Name	Shape classification type – feature level		
Description	The signal "Shape classification type – feature level" (A.3.10) provides the classification type for the shape.		
Value type	enumeration	**Unit**	1

Table A.240 — Enumeration: Shape classification type – feature level – Example enumerators

Name	Description	RL enumerator
ShapeType_Background	The shape is classified as a background entity.	"exemplary"
ShapeType_Foreground	The shape is classified as a foreground entity.	"exemplary"
ShapeType_Flat	The shape is classified as a flat entity.	"exemplary"
ShapeType_Upright	The shape is classified as an upright entity.	"exemplary"
ShapeType_Ground	The shape is classified as a ground entity.	"exemplary"
ShapeType_Building	The shape is classified as a building entity.	"exemplary"
ShapeType_Vegetation	The shape is classified as a vegetation entity.	"exemplary"
ShapeType_Road	The shape is classified as a road entity.	"exemplary"
ShapeType_NonRoad	The shape is classified as a non-road entity.	"exemplary"
ShapeType_Sidewalk	The shape is classified as a sidewalk entity.	"exemplary"
ShapeType_Pedestrian	The shape is classified as a pedestrian entity.	"exemplary"
ShapeType_Vehicle	The shape is classified as a vehicle entity.	"exemplary"
ShapeType_TrafficSign	The shape is classified as a traffic sign entity.	"exemplary"
ShapeType_PedestrianFront	The shape is classified as a pedestrian front-view entity.	"exemplary"
ShapeType_PedestrianSide	The shape is classified as a pedestrian side-view entity.	"exemplary"
ShapeType_PedestrianRear	The shape is classified as a pedestrian rear-view entity.	"exemplary"

A.3.11 Shape classification type – confidence – feature level

Table A.241 — Signal: Shape classification type – confidence – feature level

Name	Shape classification type – confidence – feature level		
Description	The signal "Shape classification type – confidence – feature level" (A.3.11) provides the classification confidence of the shape segment's possible content.		
Value type	[0...100] real value	**Unit**	%

A.3.12 Colour value – feature level

Table A.242 — Signal: Colour value – feature level

Name	Colour value – feature level		
Description	The signal "Colour value – feature level" (A.3.12) provides the definition of the colour value(s). The number of values depends on the signal "Colour model type" (A.1.15). Additional information: For example, enumerator "ColourModelType_RGB" has 3 colour values, specifically first value is red, second value is green, third value is blue.		
Value type	[0...100] real value	**Unit**	%

A.3.13 Colour tone – confidence – feature level

Table A.243 — Signal: Colour tone – confidence – feature level

Name	Colour tone – confidence – feature level		
Description	The signal "Colour tone – confidence – feature level" (A.3.13) provides the confidence of the shape area with the specified colour tone. The tone is defined by the signal "Colour model type" (A.1.15) and the list of signals "Colour value – feature level" (A.3.12).		
Value type	[0...100] real value	**Unit**	%

A.3.14 Shape type – feature level

Table A.244 — Signal: Shape type – feature level

Name	Shape type – feature level		
Description	The signal "Shape type – feature level" (A.3.14) provides the shape type which is used for the shape segment's borderline. Additional information: The shape defines how the referenced points are connected. The interpretation of the enumerators will be defined during the system design phase.		
Value type	enumeration	**Unit**	1

Table A.245 — Enumeration: Shape type – feature level – Example enumerators

Name	Description	RL enumerator
ShapeType_Point	The shape is a point.	"exemplary"
ShapeType_Box	The shape is a box (2 or 3 points).	"exemplary"
ShapeType_Ellipse	The shape is an ellipse (2 or 3 points).	"exemplary"
ShapeType_Polygon	The shape is a polygon (3 or more points).	"exemplary"
ShapeType_Polyline	The shape is a polyline (2 or more points).	"exemplary"
ShapeType_PointCloud	The shape is a point cloud (2 or more points).	"exemplary"

A.3.15 Number of valid shape points – feature level

Table A.246 — Signal: Number of valid shape points – feature level

Name	Number of valid shape points – feature level		
Description	The signal "Number of valid shape points – feature level" (A.3.15) provides the current number of valid shape points. The shape points are part of the polyline, which defines the border of the shape. Additional information: The signal "Number of valid shape points – feature level" (A.3.15) >= 1.		
Value type	[0...] integer value	**Unit**	1

A.3.16 Point existence probability – feature level

Table A.247 — Signal: Point existence probability – feature level

Name	Point existence probability – feature level		
Description	The signal "Point existence probability – feature level" (A.3.16) provides the existence probability of the point [see signal "Position – feature level {x, y, z}" (A.3.17)]. Each point is only used by one shape [maybe one shape is sent in several entities with different shape classification confidences – see signal "Shape classification type – confidence – feature level" (A.3.11)].		
Value type	[0...100] real value	**Unit**	%

A.3.17 Position – feature level {x, y, z}

Table A.248 — Signal: Position – feature level {x, y, z}

Name	Position – feature level {x, y, z}

Table A.248 *(continued)*

Description	The signal “Position – feature level {x, y, z}” (A.3.17) provides the 3D point in ego-vehicle coordinate system [see signal “Vehicle coordinate system type” (A.1.21)].		
Value type	3D vector real value	**Unit**	(m, m, m)

A.3.18 Position – feature level {x, y, z} – error

Table A.249 — Signal: Position – feature level {x, y, z} – error

Name	Position – feature level {x, y, z} – error		
Description	The signal “Position – feature level {x, y, z} – error” (A.3.18) provides the error representing the uncertainty of the state estimation of the signal “Position – feature level {x, y, z}” (A.3.17).		
Value type	scalar/vector/matrix real value [see subclause “Error model implementation” (B.4.1)]	**Unit**	see signal “Position – feature level {x, y, z}” (A.3.17)

A.3.19 Number of valid shape reference points – feature level

Table A.250 — Signal: Number of valid shape reference points – feature level

Name	Number of valid shape reference points – feature level		
Description	The signal “Number of valid shape reference points – feature level” (A.3.19) provides the current number of valid shape reference points. Each signal's “Position – feature level {x, y, z}” (A.3.17) position of a shape reference point is part of the shape surface.		
Value type	[0...] integer value	**Unit**	1

A.3.20 Shape surface normal {x, y, z}

Table A.251 — Signal: Shape surface normal {x, y, z}

Name	Shape surface normal {x, y, z}		
Description	The signal “Shape surface normal {x, y, z}” (A.3.20) provides the normal vector of the shape's approximated tangential plane at the corresponding signal “Position – feature level {x, y, z}” (A.3.17). Additional information: The vector is normalised.		
Value type	3D vector real value	**Unit**	(1, 1, 1)

A.3.21 Shape surface normal {x, y, z} – error

Table A.252 — Signal: Shape surface normal {x, y, z} – error

Name	Shape surface normal {x, y, z} – error		
Description	The signal “Shape surface normal {x, y, z} – error” (A.3.21) provides the error of the signal “Shape surface normal {x, y, z}” (A.3.20) which is the normal of the shape's approximated tangential plane at the shape reference point [see corresponding signal “Position – feature level {x, y, z}” (A.3.17)].		
Value type	scalar/vector/matrix real value [see subclause “Error model implementation” (B.4.1)]	**Unit**	see signal “Shape surface normal {x, y, z}” (A.3.20)

A.3.22 Translation rate {x, y, z} – feature level

Table A.253 — Signal: Translation rate {x, y, z} – feature level

Name	Translation rate {x, y, z} – feature level		
Description	The signal "Translation rate {x, y, z} – feature level" (A.3.22) provides the scaled translation of the shape's approximated tangential plane at the shape reference point [see corresponding signal "Position – feature level {x, y, z}" (A.3.17)]. Longitudinally (along the view axis) that is the inverse time to collision. No motion → (0, 0, 0) m/s		
Value type	3D vector real value	**Unit**	(m/s, m/s, m/s)

A.3.23 Translation rate {x, y, z} – feature level – error

Table A.254 — Signal: Translation rate {x, y, z} – feature level – error

Name	Translation rate {x, y, z} – feature level – error		
Description	The signal "Translation rate {x, y, z} – feature level – error" (A.3.23) provides the error of the signal "Translation rate {x, y, z} – feature level" (A.3.22) of the shape's approximated tangential plane at the shape reference point [see corresponding signal "Position – feature level {x, y, z}" (A.3.17)].		
Value type	scalar/vector/matrix real value [see subclause "Error model implementation" (B.4.1)]	**Unit**	see signal "Translation rate {x, y, z} – feature level" (A.3.22)

A.3.24 Rotation rate {yaw, pitch, roll}

Table A.255 — Signal: Rotation rate {yaw, pitch, roll}

Name	Rotation rate {yaw, pitch, roll}		
Description	The signal "Rotation rate {yaw, pitch, roll}" (A.3.24) provides the rotation change of the shape's approximated tangential plane at the shape reference point [see corresponding signal "Position – feature level {x, y, z}" (A.3.17)]. No rotation change → (0, 0, 0) rad/s		
Value type	3D vector real value	**Unit**	(rad/s, rad/s, rad/s)

A.3.25 Rotation rate {yaw, pitch, roll} – error

Table A.256 — Signal: Rotation rate {yaw, pitch, roll} – error

Name	Rotation rate {yaw, pitch, roll} – error		
Description	The signal "Rotation rate {yaw, pitch, roll} – error" (A.3.25) provides the error of the signal's "Rotation rate {yaw, pitch, roll}" (A.3.24) change of the shape's approximated tangential plane at the shape reference point [see corresponding signal "Position – feature level {x, y, z}" (A.3.17)].		
Value type	scalar/vector/matrix real value [see subclause "Error model implementation" (B.4.1)]	**Unit**	see signal "Rotation rate {yaw, pitch, roll}" (A.3.24)

A.3.26 Scale change – feature level

Table A.257 — Signal: Scale change – feature level

Name	Scale change – feature level