Document Title	Specification of Sensor Interfaces
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	912

Document Status	published
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	R20-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2020-11-30	R20-11	AUTOSAR Release Management	• Initial release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Introduction

This specification describes the functional description and interfaces of the function cluster `Sensor Interfaces` which belongs to `AUTOSAR Adaptive Platform` Services.The `Sensor Interfaces` has the responsibility of connecting sensors to AUTOSAR Adaptive computing unit via a service interface.

The `Sensor Interfaces` are based on the logical interface defined in [1]. The sensor types covered by the specification are the following:

- Camera Sensors
- Lidar Sensors
- Radar Sensors
- Ultrasonic Sensors

There are three levels for the sensor data reporting:

- Detection level
- Feature level
- Object level

Besides data reporting interfaces, the sensor supporing interfaces, i.e.sensor health and performance, are also defined.

The sensors as the `AUTOSAR service provider` and the `AUTOSAR Adaptive applications` are located in different ECUs. To allow the flexiblity usage of the `Sensor Interfaces`, it will expose its functionality via ara::com service interfaces, not direct APIs.

Further functional details are described in [2].

2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations relevant to the Sensor Interfaces that are not included in AUTOSAR Glossary [3].

Abbreviation / Acronym:	Description:
AD	Automated Driving
ADI	Automated Driving Interfaces
AEB	Autonomous Emergency Braking
HiL	Hardware in the Loop
ISO	International Organization for Standardization
LIDAR	LIght Detection And Ranging
MiL	Model in the Loop
OEM	Original Equipment Manufacturer
OSI	Open Simulation Interface
RADAR	RAdio Detection And Ranging
SAE	Society of Automotive Engineers
USS	UltraSonic Sensor
XiL	... in the Loop

Terms:	Description:
V2X	Vehicle-to-X-Communication is the generic term for various communication technologies in automotive, including vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. The information is either transmitted directly between vehicles, between vehicle and roadside infrastructure or by using existing mobile networks.

3 Related documentation

3.1 Input documents & related standards and norms

[1] ISO-DIS-23150 Road vehicles–Data communication between sensors and data fusion unit for automated driving functions–Logical interface

[2] Explanation of Sensor Interfaces
AUTOSAR_EXP_SensorInterfaces

[3] Glossary
AUTOSAR_TR_Glossary

[4] Specification of the Adaptive Core
AUTOSAR_SWS_AdaptiveCore

[5] Specification of Communication Management
AUTOSAR_SWS_CommunicationManagement

[6] Specification of Identity and Access Management
AUTOSAR_SWS_IdentityAndAccessManagement

[7] Requirements on Automated Driving Interfaces
AUTOSAR_RS_AutomatedDrivingInterfaces

3.2 Further applicable specification

AUTOSAR provides a core specification [4] which is also applicable for `Automated Driving Interfaces`. The chapter “General requirements for all FunctionalClusters” of this specification shall be considered as an additional and required specification for implementation of `Automated Driving Interfaces`.

4 Constraints and assumptions

Dedicated interfaces for particular sensors like radar, lidar, USS and camera are provided.

4.1 Limitations

4.1.1 ISO Optionals

All the ISO optionals are supported in the service defintiion, but the presence of an optional signal is decided during the design time. The optional signal presence is indicated by a capability vector. There is no change of optional presence during the run time.

4.1.2 Sensor Control Interfaces

Sensor control interfaces are not supported, e.g. actions like reset, initialization, and calibration.

4.1.3 Sensor Capabilities

The configuration of sensor capabilities is not supported, e.g. sensor opening angle and detection range.

4.2 Applicability to car domains

AUTOSAR Sensor Interfaces are used as service interfaces between dedicated sensors for enviroment precesion and the `AUTOSAR Adaptive applications`.

5 Dependencies to other modules

The Automated Driving Interfaces serve as common interfaces for the AUTOSAR Adaptive applications. The sensor information is provided by a non-platform service.The sensor interfaces are exposed to client applications via the ara::com middleware. Communication Management,[5] uses Identity and Access Management [6] to validate the authorization of requests made to the specific service interfaces, i.e. the Communication Management shall check if the invoker is allowed to access the requested service interface of a specific sensor. The following Figure shows a radar and camera sensor instance which provide sensor information to a Data Fusion instance consuming sensor data.

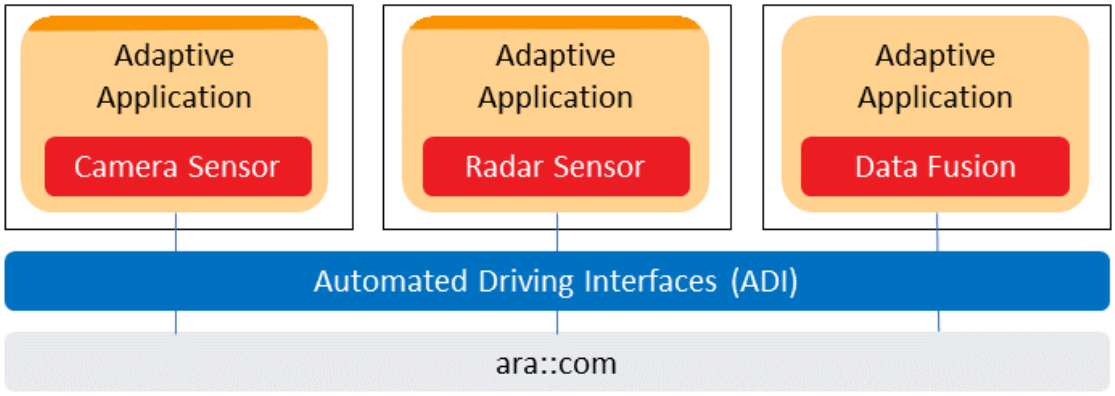

Figure 5.1: Sensor Interfaces dependencies to other AUTOSAR Adaptive Applications.

For each sensor type like radar, lidar, USS or camera, the ISO standard will contain lots of optional sensor data or signals in addition to the required signals, e.g. the classification of an object might be optional. During development of a driving function or data fusion system, a specific set of optional signals is selected by the function or data fusion developer according to the requirements of the driving function. To reduce development costs, especially in terms of functional safety, the set of optionals and thus the resulting logical sensor interface are required to be fixed during design-time, i.e. the specified sensor signals must not suddenly disappear from the interface of the sensor during runtime of the system so that a data fusion system can rely on the presence of the specified signals. Vice versa, a sensor interface must not add additional signals during runtime, e.g. not to risk an unintentional behavior change due to the increased bandwidth requirements.

6 Requirements Tracing

The following tables reference the requirements specified in [7] and links to the fulfillment of these. Please note that if column “Satisfied by” is empty for a specific requirement this means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by

7 Functional Specification

7.1 Outline

7.1.1 Goals and scope

The specification is to provide well-defined sensor interfaces which is compliant to the sensor interface specification [1]. The ISO document covers just semantic definitions of interfaces whereas this specification aims to cover all aspects of the interfaces to make them fully compliant to the AUTOSAR Adaptive platform.

The specification shall enable the compatibility of sensors and data fusion algorithms independently of the supplier. To ensure the compatibility between sensors the interfaces are provided on implementation data type level.

The Automated Driving Interfaces serve as common interfaces for the `AUTOSAR Adaptive applications`. Consumers of the interfaces are `AUTOSAR Adaptive applications` containing functional elements as sensor data receivers, sensor data processors, data fusion applications, and automated driving functions. The sensor information is provided by a non-platform service.The non-platform service consumer is running in the apdative platfom as an `AUTOSAR adaptive application`, where the sensor could be deployed on different AUTOSAR platforms (Classic and Adaptive) as shown in the following figure.

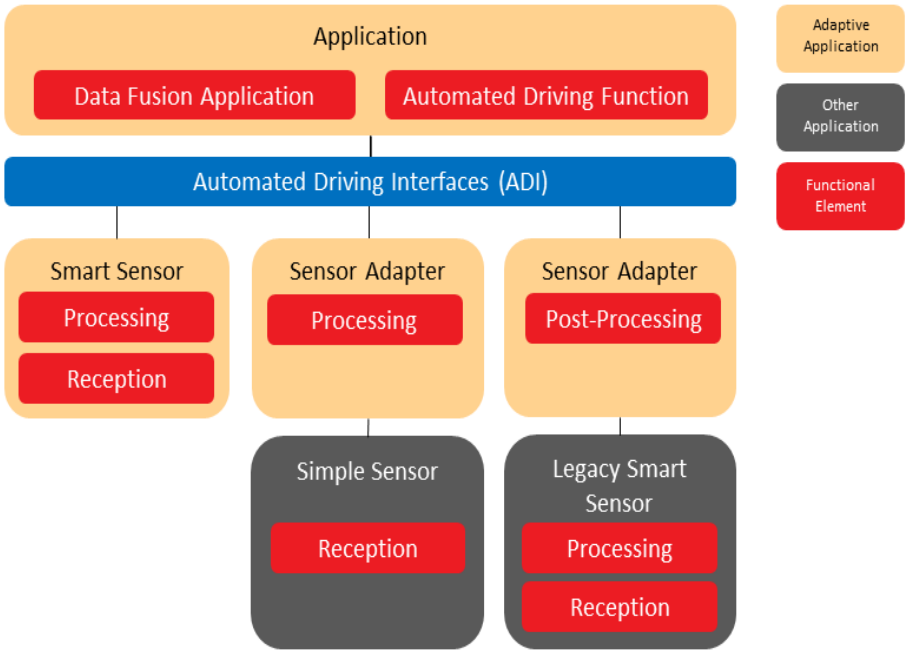

Figure 7.1: Functional elements communicating via ADI

Apart from the Automated Driving Interfaces (ADI), the other relevent elements are defined as follows :

- Sensor Data Reception: A functional element which receives or reads sensor data from a sensor. Such an element can provide raw sensor data. Raw sensor data means that it provides the actual data a sensor can measure. There is no processing of the sensor data. There are no data streams provided.
- Sensor Data Processing: This element which usually processes raw sensor data, provides pre-processed and post-processed sensor data. It provides data a sensor cannot directly measure but needs a processing stage for.
- Sensor Fusion Application: The data fusion algorithm is an application which collects sensor data and fuses it to provide a unified environmental model. Data fusion applications are typical consumers of sensor data. They provide their output to Automated Driving Functions.
- Automated Driving Function: Automated Driving Functions can also directly use the interfaces.

The Automated Driving interfaces can be used in following usages to enable the interoperability between different players and improve the efficiency for the development and validation of automated driving functions :

- Sensor Supplier Interface: The automated driving use case defines the requirements to the fusion algorithm and the used sensors. The OEM wants to rely on a standardized interface which provides well defined sensor information which are defined by the ISO-23150 and on an AUTOSAR standardized interface. The scope shall be the data format between a sensor and a computing unit running an AUTOSAR Adaptive application.
- Standardized Sensor API: Sensor developers need well described and standardized interfaces to provide an implementation which can be used by automated driving applications and data fusion algorithms. So, the same information is provided to all potential consumers of the data.
- Sensor Fusion Algorithm: To combine information from different sensors and to develop sensor fusion algorithms for automated driving a detailed description of all signals provided is needed.
- Sensor Fusion Integration: Sensor fusion algorithms shall be integrated on a high-performance computing platform within an Adaptive Application as the central unit of automated driving functions.
- Sensor Implementation Testing: Testing is performed at each stage of the development of the sensor system. To create a test specification for the system testable implementations are needed which are based on well-defined interfaces and descriptions.
- Sensor Simulation: By simulating the sensor fusion algorithms before actual target integration, it is possible to ensure smooth integration during the design, development, test and safeguarding steps. This ensures compatibility of algorithm integration into AUTOSAR Adaptive ECUs.

- Sensor Data Record and Replay: To be able to test and debug sensor fusion algorithms offline it is necessary to replay sensor data recorded in real world scenarios. This enables sensor fusion designers to reproduce failures and determine faults.

7.1.2 Relation to other standards

The currently ongoing standardization of the international standard ISO 23150 "Road vehicles - Data communication between sensors and data fusion unit for automated driving functions - Logical interface" will be applicable to road vehicles with automated driving functions. It specifies the logical interface between smart in-vehicle sensors that sense the environment (e.g. camera, lidar, radar, ultrasonic, etc.) and the vehicle fusion unit. The interface is described in a modular, semantic representation allowing different types of sensor technologies and fusion concepts. The "Open Simulation Interface" (OSI) is a generic software interface for the environmental perception of automated driving functions in virtual scenarios (https://github.com/OpenSimulationInterface/open-simulation-interface). OSI ensures modularity, integrability and interchangeability of the individual components. These can be environment simulation models, sensor models, logical models for the detected environment or sensor fusion and models for automated driving functions. Additional informations can be found on https://opensimulationinterface.github.io/open-simulation-interface/.

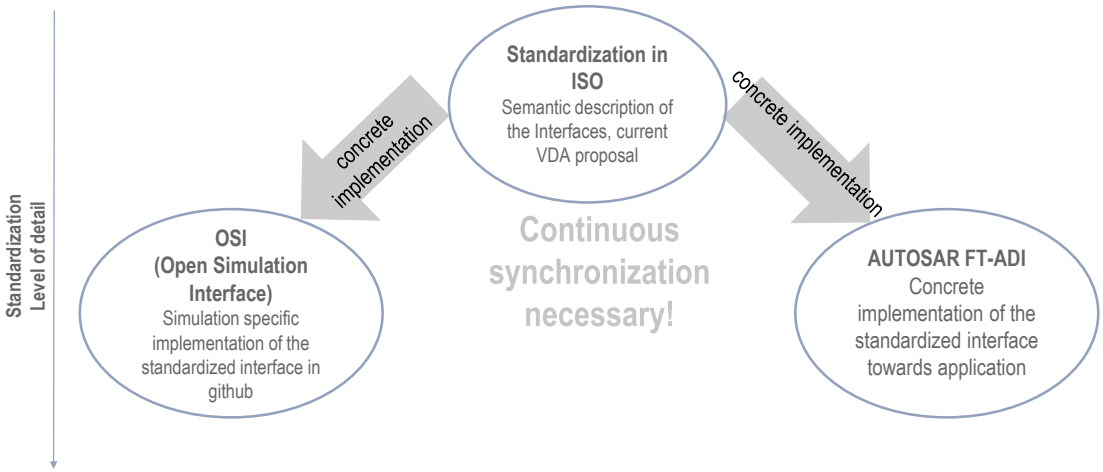

Figure 7.2: Scope of Standardization

The Open Simulation Interface and the Autosar ADI will support the ISO 23150.

7.2 AD Sensor Service Design

7.2.1 ISO mapping to Sensor Services

7.2.1.1 Sensor Types

To comply with the ISO 23150, the specific service is defined with different sensor types. Currently, Camera, Lidar, Radar, and Ultrasonic sercives are supported in the specification. The new service will be introduced along with the evolution of [1].
[SWS_ADI_00001]{DRAFT} **The Camera Sensor Data periodical Transmission** ⌈A Camera senor with Automated Driving interfaces shall provide the camera sensor data periodically to the `Adaptive Applications`⌋*(RS_ADI_00001)*

[SWS_ADI_00002]{DRAFT} **The Lidar Sensor Data periodical Transmission** ⌈A Lidar sensor with Automated Driving interfaces shall provide the lidar sensor data periodically to the `Adaptive Applications`⌋*(RS_ADI_00002)*

[SWS_ADI_00003]{DRAFT} **The Radar Sensor Data periodical Transmission** ⌈A radar sensor with Automated Driving interfaces shall be able to provide the radar sensor data periodically to the `Adaptive Applications`⌋*(RS_ADI_00003)*

[SWS_ADI_00004]{DRAFT} **The Ultrasonic Sensor Data periodical Transmission** ⌈A ultrasonic sensor with Automated Driving interfaces shall be able to provide the ultrasonic sensor data periodically to the `Adaptive Applications`⌋*(RS_ADI_00004)*

[SWS_ADI_00005]{DRAFT} **Receving periodical sensor data** ⌈The `Adaptive Applications` as the sevice consumer shall process the sensor data according to the timestamp in the package, not the package arrival time. The most recent data is always the highest priority for processing.⌋*(RS_ADI_00018)*

7.2.1.2 Sensor Data level

In ISO 23150, there are three ISO logic interface levels to support different levels of fusion: object level, feature level and detection level:

- Object level interface: Potentially moving object interface, Road object interface and Static object interface;
- Feature level interface: Camera feature interface and USS feature interface;
- Detection level interface: Radar detection interface,Lidar detection interface, Camera detection interface and USS detection interface.

A sensor could provide several ISO logical interfaces, and shall provide at least one ISO interface either on object, feature, or detection level.

[SWS_ADI_00006]{DRAFT} **Sensor specific services for different level ISO interfaces** ⌈To allow the flexibility of service providing by a sensor, each ISO interface shall be mapped to a propriate `ara::com` means like events. The service interface shall

support the defintions of ISO signals.The services for new types of sensors shall be added along with the evolution of [1]. The follwing AP services are defined in this specification:

- Camera,Lidar,Radar,Ultrasonic: Potentially moving object service, road object service, static object service
- Camera: Camera features service, camera detections service;
- Lidar: Lidar detections service;
- Radar: Radar detections service;
- Ultrasonic: Ultrasonic features service, Ultrasonic detection service.

⌋(*RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*, *RS_ADI_00019*, *RS_ADI_00005*)

7.2.1.3 Supportive Sensor interfaces

In ISO, there are two supportive senor interfaces for the `Adaptive Applications` process the reliability of the used measurement method or data, which are also used to the implementation of safety concept. The two supportive sensor interfaces are sensor technology indepedent, i.e. common to camera, lidar, radar and ultrasonic sensor, but with sensor technology specific data.

[SWS_ADI_00007]{DRAFT} **Sensor independent supportive services** ⌈The supportive interfaces shall be mapped to two sensor indpendent AP services and support the defintions of ISO signals.The follwing services are defined in this specification:

- Sensor Performance service;
- Sensor Health information service.

⌋(*RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*, *RS_ADI_00019*)

7.2.2 Service Optional Elements

An individual Sensor Service design faces the challenge that each ISO interface contains a lot of optional elements. The optional elements are known and fixed at design time by service providers and service users. The present of optional elements shall not be changed during run time.

[SWS_ADI_00010]{DRAFT} **Capability Vector** ⌈A service capability vector indicates which optional element is provided by the Service provider. The optional elements indicated by the capability vector shall be always sent during the runtime of the service. The service consumer should check the vector at the initialization time and subcribe the service if it includes all the optional elements that the consumer requires.The consumer may ignore the optional elements, if these elements are not required.⌋(*RS_ADI_00006*, *RS_ADI_00015*, *RS_ADI_00016*, *RS_ADI_00017*)

[SWS_ADI_00011]{DRAFT} **Service Profiles** ⌈In different use cases, there are different requirements on the sensor data i.e.for different data Types, behavior, content, performance, resolution, etc. Different Service Profiles for a Sensor Type enables the flexiblity of the corresponding sensor service for different use cases. In this release, only one profile is supported and more profiles should be considered in future releases.⌋*(RS_ADI_00007)*

[SWS_ADI_00012]{DRAFT} **Service Versioning** ⌈Service Versioning covers different configurations, i.e. optionals, additional data representation, etc. The same services with different service versioning, but with the same profile, shall be backward compatible.⌋*(RS_ADI_00006, RS_ADI_00015, RS_ADI_00016, RS_ADI_00017)*

8 API specification

There are no APIs defined in this release.

9 Service Interfaces

9.1 Type definitions

This chapter lists all types provided by the ADI.

9.1.1 CapabilityVector

[SWS_ADI_00100]{DRAFT} ⌈

Name	CapabilityVector
Kind	VECTOR
Subelements	`bool`
Derived from	-
Description	To identify presence of optional signal. The exact bit reference is denfined in the specification. For each service, there is exact one capabilityVector table, which is deifined in chapter 10.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.2 Version

[SWS_ADI_00101]{DRAFT} ⌈

Name	Version
Kind	STRUCTURE
Subelements	Major `uint8_t` Minor `uint8_t` Patch `uint8_t`
Derived from	-
Description	Represents the version of the service.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.3 InterfaceCycleTimeVariation

[SWS_ADI_00102]{DRAFT} ⌈

Name	InterfaceCycleTimeVariation
Kind	TYPE_REFERENCE
Derived from	`uint8_t`
Description	To indicate the cycle time variation. It represents a linear value between 0% and 100%.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.4 InterfaceID

[SWS_ADI_00103]{DRAFT} ⌈

Name	InterfaceID	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Uniquely identify the interface.	
Range / Symbol	***Limit***	***Description***
kPotentialMovingObject	0x00	The Potential Moving Object interface.
kRoadObject	0x01	The Road Object interface.
kStaticObject	0x02	The Static Object interface.
kCameraFeature	0x03	The Camera Feature interface.
kUltrasonicFeature	0x04	The Ultrasonic Feature interface.
kRadarDetection	0x05	The Radar Detection interface.
kLidarDetection	0x06	The Lidar Detection interface.
kCameraDetection	0x07	The Camera Detection interface.
kUltrasonicDetection	0x08	The Ultrasonic Detection interface.
kSensorPerformance	0x09	The Sensor Performance interface.
kSensorHealthInformation	0x0A	The Sensor Health interface.
kUnknown	0x0B	The interface ID is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.5 DataQualifier

[SWS_ADI_00104]{DRAFT} ⌈

Name	DataQualifier
Kind	TYPE_REFERENCE
Derived from	`uint8_t`
Description	To identify the data quality of the transmitted data.

▽

△

Range / Symbol	*Limit*	*Description*
kNormal	0x00	Information can be used without restriction.
kNotAvailable	0x01	Information from the senor is not available.
kReduceInCoverage	0x02	Reported data are from the sensor in a restricted view.
kReduceInPerformance	0x03	Reported data are from the sensor in a restricted performance.
kReduceInViewAnd Performance	0x04	Reported data are from the sensor in a restricted view and performance.
kTestMode	0x05	Reported data are from the sensor in a test mode.
kInvalid	0x06	Mesaurement cycle was invalid and no valid objects will be reported.
kUnknown	0x07	Datat quality is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.6 RecognizedStatus

[SWS_ADI_00105]{DRAFT} ⌈

Name	RecognizedStatus	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	To indicate the recognition process status of the sensor due to resource limitations.	
Range / Symbol	*Limit*	*Description*
kNormal	0x00	The performance of the sensor is enough to process all recognized entities. If additional entities would have been in the environment the sensor would have processed additional entities.
kPreLimits	0x01	The performance of the sensor is close to the limits to process all recognized entities.
kLimited	0x02	The performance of the sensor is not enough to process all recognized entities.
kUnknown	0x03	The performance status of the sensor is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.7 TrackingMotionModel

[SWS_ADI_00106]{DRAFT} ⌈

Name	TrackingMotionModel	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	The motion model for tracking moving objects.	
Range / Symbol	**Limit**	**Description**
kConstantVelocity	0x00	Motion model uses constant velocity.
kConstantAcceleration	0x01	Motion model uses constant acceleration.
kConstantTurnRate	0x02	Motion model uses constant turn rate.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.8 MotionType

[SWS_ADI_00107]{DRAFT} ⌈

Name	MotionType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	To identify the applied motion type in the message, i.e. absolute or relative motion values.	
Range / Symbol	**Limit**	**Description**
kRelativeValues	0x00	Relative motion values are used.
kAbsoluteValues	0x01	Absolute motion values are used.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.9 ColourModelType

[SWS_ADI_00108]{DRAFT} ⌈

Name	ColourModelType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	To identify the applied colour model in the service.	
Range / Symbol	**Limit**	**Description**
kGrey	0x00	Grey scale - 1 colour value is used for grey value.
kRGB	0x01	Red, green, blue - 3 colour values are used for the values for red, green, blue.
kHSV	0x02	Hue, saturation, value - 3 colour values are used for the values for hue, saturation, value.

▽

△

kLUV	0x03	Luminance and colour coordinates U, V - 3 colour values are used for the values for hue, saturation, value.
kColourList	0x04	Each value reference a pre-defined colour - 1 colour value.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.10 RadialVelocityAmbiguityDomain

[SWS_ADI_00109]{DRAFT} ⌈

Name	RadialVelocityAmbiguityDomain
Kind	STRUCTURE
Subelements	Begin `float` End `float`
Derived from	-
Description	The Doppler ambiguity caused by under sampling. m/s

⌋(*RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.11 RangeAmbiguityDomain

[SWS_ADI_00110]{DRAFT} ⌈

Name	RangeAmbiguityDomain
Kind	STRUCTURE
Subelements	Begin `float` End `float`
Derived from	-
Description	The range of the ambiguity domain. m

⌋(*RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.12 AngleAzimuthAmbiguityDomain

[SWS_ADI_00111]{DRAFT} ⌈

Name	AngleAzimuthAmbiguityDomain
Kind	STRUCTURE
Subelements	Begin `float` End `float`
Derived from	-
Description	The azimuth angle of the ambiguity domain is defined by {begin} and {end}. The unit is {rad,rad}

⌋(*RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.13 AngleElevationAmbiguityDomain

[SWS_ADI_00112]{DRAFT} ⌈

Name	AngleElevationAmbiguityDomain
Kind	STRUCTURE
Subelements	Begin `float` End `float`
Derived from	-
Description	The elevation angle of the ambiguity domain is defined by {begin} and {end}.The unit is {rad,rad}.

⌋(*RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.14 ScopeOfInterfaces

[SWS_ADI_00113]{DRAFT} ⌈

Name	ScopeOfInterfaces	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	To identify the related interface of the service.	
Range / Symbol	***Limit***	***Description***
kObjectLevelInterfaces	0x00	Object level interfaces.
kFeatureLevelInterface	0x01	Feature level interface.
kDetectionLevelInterface	0x02	Detection level interface.
kFLIandDLI	0x03	Feature level and detection level interfaces.
kPMOI	0x04	Potential Moving Objects Interface.
kRDOI	0x05	Road Objects Interface.
kSOI	0x06	Static Objects Interface.
kAll	0x07	All interfaces.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.15 VehicleCoordinateSystemType

[SWS_ADI_00114]{DRAFT} ⌈

Name	VehicleCoordinateSystemType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Defines the reference vehicle coordinate system for the interfaces of the sensor.	
Range / Symbol	***Limit***	***Description***
kRearAxle	0x00	Use vehicle rear axle coordinate system.
kRoadLevel	0x01	Use vehicle road level coordinate system.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.16 Point3D

[SWS_ADI_00115]{DRAFT} ⌈

Name	Point3D
Kind	STRUCTURE
Subelements	x `float` y `float` z `float` (optional)
Derived from	-
Description	Represents a 3 dimension vector, the unit will be vary according to the refering data type.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.17 Point3DError

[SWS_ADI_00116]{DRAFT} ⌈

Name	Point3DError
Kind	STRUCTURE
Subelements	x `float` y `float` z `float` (optional)
Derived from	-
Description	Represents Error value in 3 dimension vector, the unit will be vary according to the refering data type.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.18 Orientation3D

[SWS_ADI_00117]{DRAFT} ⌈

Name	Orientation3D
Kind	STRUCTURE
Subelements	Yaw `float` Pitch `float` Roll `float`
Derived from	-
Description	Represents a 3 dimension vector {yaw, pitch,roll}, the unit will be vary according to the refering data type.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.19 Orientation3DError

[SWS_ADI_00118]{DRAFT} ⌈

Name	Orientation3DError
Kind	STRUCTURE
Subelements	Yaw `float` Pitch `float` Roll `float`
Derived from	-
Description	Represents Error value in 3 dimension vector {yaw, pitch, roll}, the unit will be vary according to the refering data type.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.20 VanishingPointDirection2DSpheric

[SWS_ADI_00119]{DRAFT} ⌈

Name	VanishingPointDirection2DSpheric
Kind	STRUCTURE
Subelements	Azimuth `float` Elevation `float`
Derived from	-

▽

△

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 Vanishing point {azimuth, elevation}.The unit is {rad,rad}.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.21 VanishingPointDirection2DSphericError

[SWS_ADI_00120]{DRAFT} ⌈

Name	VanishingPointDirection2DSphericError
Kind	STRUCTURE
Subelements	Azimuth `float` Elevation `float`
Derived from	-
Description	Uncertainty of the Vanishing point {azimuth, elevation},using Error value.The unit is {rad,rad}.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.22 SensorMountingInfo

[SWS_ADI_00121]{DRAFT} ⌈

Name	SensorMountingInfo
Kind	STRUCTURE
Subelements	OriginPosition `Point3D` OriginPositionError `Point3DError` (optional) OriginOrientation `Orientation3D` OriginOrientationError `Orientation3DError` (optional)
Derived from	-
Description	Sensor Mounting information in vehicle coordinate system.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.23 VanishingPointInfo

[SWS_ADI_00122]{DRAFT} ⌈

Name	VanishingPointInfo
Kind	STRUCTURE
Subelements	VPDirection `VanishingPointDirection2DSpheric` VPError `VanishingPointDirection2DSphericError` (optional)
Derived from	-
Description	Represents the vanishing point info.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.24 HeaderBasic

[SWS_ADI_00123]{DRAFT} ⌈

Name	HeaderBasic
Kind	STRUCTURE
Subelements	VersionID `Version` SensorID `uint8_t` InterfaceID `InterfaceID` (optional) TimeStampP `uint64_t` CycleCounter `uint8_t` (optional) ItfCycleTime `uint32_t` (optional) ItfCycleTimeV `InterfaceCycleTimeVariation` (optional) Qualifier `DataQualifier`
Derived from	-
Description	Represents the basic info of message header.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.25 HeaderBasicDetection

[SWS_ADI_00124]{DRAFT} ⌈

Name	HeaderBasicDetection
Kind	STRUCTURE
Subelements	VersionID `Version` SensorID `uint8_t` IterfaceID `InterfaceID` (optional) TimeStampM `uint64_t` ▽

▽

△

	△ ItfCycleTime `uint32_t` (optional) ItfCycleTimeV `InterfaceCycleTimeVariation` (optional) Qualifier `DataQualifier`
Derived from	-
Description	Represents the basic info of service header.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.26 PMOHeader

[SWS_ADI_00125]{DRAFT} ⌈

Name	PMOHeader
Kind	STRUCTURE
Subelements	Basic `HeaderBasic` VehicleCoordinateST `VehicleCoordinateSystemType` TMModel `TrackingMotionModel` (optional) Motion `MotionType`
Derived from	-
Description	Represents the basic info of potential moving object message.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.27 GeneralHeader

[SWS_ADI_00126]{DRAFT} ⌈

Name	GeneralHeader
Kind	STRUCTURE
Subelements	Basic `HeaderBasic` VehicleCoordinateST `VehicleCoordinateSystemType` ColourModelType `ColourModelType` (optional)
Derived from	-
Description	Represents the Genreal header infomation.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.28 MeasurementStatus

[SWS_ADI_00200]{DRAFT} ⌈

Name	MeasurementStatus	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Represent the measurement Status.	
Range / Symbol	***Limit***	***Description***
kMeasured	0x00	The tracked object is measured.
kNew	0x01	The tracked object is new.
kPredicted	0x02	The result is by predicted.
kPartiallyMeasured	0x03	The tracked object is partically measured.
kInvalid	0x04	The measurement result is invalid.
kPredictedOccluded	0x05	Tracked object is temporarily occluded by another entity.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.29 ReferencePoints

[SWS_ADI_00201]{DRAFT} ⌈

Name	ReferencePoints	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Represent outer edges of the recognized object's bounding box.	
Range / Symbol	***Limit***	***Description***
kFrontLeftTop	0x00	Front Left Top.
kFrontMidwidthTop	0x01	Front Midwidth Top.
kFrontRightTop	0x02	Front Right Top.
kMidsideLeftTop	0x03	Midside Left Top.
kMidsideMidwidthTop	0x04	Midside Midwidth Top.
kMidside Right Top.	0x05	Midside Right Top.
kRearLeftTop	0x06	Rear Left Top.
kRearMidwidthTop	0x07	Rear Midwidth Top.
kRearRightTop	0x08	Rear Right Top.
kFrontLeftMidheight	0x09	Front Left Midheight.
kFrontMidwidthMidheight	0x0A	Front Midwidth Midheight.
kFrontRightMidheight	0x0B	Front Right Midheight.
kMidsideLeftMidheight	0x0C	Midside Left Midheight.
kMidsideMidwidthMidheight	0x0D	Midside Midwidth Midheight.
kMidsideRightMidheight	0x0E	Midside Right Midheight.

▽

△

kRearLeftMidheight	0x0F	Rear Left Midheight.
kRearMidwidthMidheight	0x10	Rear Midwidth Midheight.
kRearRightMidheight	0x11	Rear Right Midheight.
kFrontLeftBottom	0x12	Front Left Bottom.
kFrontMidwidthBottom	0x13	Front Midwidth Bottom.
kFrontRightBottom	0x14	Front Right Bottom.
kMidsideLeftBottom	0x15	Midside Left Bottom.
kMidsideMidwidthBottom	0x16	Midside Midwidth Bottom.
kMidsideRightBottom	0x17	Midside Right Bottom.
kRearLeftBottom	0x18	Rear Left Bottom.
kRearMidwidthBottom	0x19	Rear Midwidth Bottom.
kRearRightBottom	0x1A	Rear Right Bottom.
kUnknown	0x1B	Unkonwn.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.30 MovementStatus

[SWS_ADI_00202]{DRAFT} ⌈

Name	MovementStatus	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Represent the relevance to ego vehicle road level.	
Range / Symbol	***Limit***	***Description***
kUnknow	0x00	The status is unknow.
kStoppedMoving	0x01	The object stopped moving.
kStationary	0x02	The object is stationary.
kMoving	0x03	The object is moving.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.31 Roadlevel

[SWS_ADI_00203]{DRAFT} ⌈

Name	Roadlevel	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Represent the relevance to ego vehicle road level.	
Range / Symbol	**Limit**	**Description**
kUnknown	0x00	The road level is unknown.
kEgoRoadLevel	0x01	The object is on the same road level of the ego vehicle.
kRoadLevelAbove	0x02	The object is above the road level of the ego vehicle .
kRoadLevelBelow	0x03	The object is below the road level of the ego vehicle .

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.32 DimensionBox

[SWS_ADI_00204]{DRAFT} ⌈

Name	DimensionBox
Kind	STRUCTURE
Subelements	Length `float` Width `float` Height `float` (optional)
Derived from	-
Description	Represents the dimension of the bounding box. The unit is {m,m,m}.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.33 DimensionBoxError

[SWS_ADI_00205]{DRAFT} ⌈

Name	DimensionBoxError
Kind	STRUCTURE
Subelements	Length `float` Width `float` Height `float` (optional)
Derived from	-
Description	Represents the Error value of the bounding box, the unit is {m,m,m}.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.34 GeometricStructures

[SWS_ADI_00206]{DRAFT} ⌈

Name	GeometricStructures	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Represent the geometrical structures that are taken into account in the bounding boxes.	
Range / Symbol	***Limit***	***Description***
kWithoutMirrors	0x00	Geometrical structures don't include the mirrors.
kWithMirrors	0x01	Geometrical structures include the Mirrors.
kUnknow	0x02	Geometrical structures including the mirrors or not is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.35 PMOClassificationType

[SWS_ADI_00207]{DRAFT} ⌈

Name	PMOClassificationType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Represents the potentially moving object classification probability type.	
Range / Symbol	***Limit***	***Description***
kCar	0x00	Probability type car.
kHeavyTruck	0x01	Probability type heavy truck.
kVan	0x02	Probability type van.
kBus	0x03	Probability type bus.
kTrailer	0x04	Probability type trailer.
kSemitrailer	0x05	Probability type semitrailer.
kTram	0x06	Probability type tram.
kTrain	0x07	Probability type train.
kOtherVehicle	0x08	Probability type unidentified vehicle.
kMotorbike	0x09	Probability type motorbike.
kAnimal	0x0A	Probability type animal.
kBicycle	0x0B	Probability type bicycle.
kTricycle	0x0C	Probability type tricycle.
kPedestrian	0x0D	Probability type pedestrian.
kWheelchair	0x0E	Probability type wheelchair.
kUnknown	0x0F	Probability type unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.36 LightStatus

[SWS_ADI_00208]{DRAFT} ⌈

Name	LightStatus	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Represents the status of an object's light.	
Range / Symbol	***Limit***	***Description***
kOff	0x00	ight status is off.
kOn	0x01	Light status is on.
kFlash	0x02	Light status is cyclic flashing.
kBrake	0x03	Light status indicates braking.
kWarning	0x04	Light status indicates warning.
kOther	0x06	Light status could not be determined.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.37 LightType

[SWS_ADI_00209]{DRAFT} ⌈

Name	LightType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Represents the light classification.	
Range / Symbol	***Limit***	***Description***
kLeftFlashLight	0x00	Vehicle's left flash light.
kRightFlashLight	0x01	Probability type heavy truck.
kHazardFlashLight	0x02	Vehicle's hazard flash light.
kLeftBrakeLight	0x03	Vehicle's left break light.
kRightBrakeLight	0x04	Vehicle's right break light.
kCentreBrakeLight	0x05	Vehicle's centre break light.
kLeftOtherLight	0x06	Vehicle's left light (no flash or break light).
kRightOtherLight	0x07	Vehicle's right light (no flash or break light).
kCentreOtherLight	0x08	Vehicle's light not on the left or right site (no flash or break light).
kUnknown	0x09	Light type unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.38 PersonPoseType

[SWS_ADI_00210]{DRAFT} ⌈

Name	PersonPoseType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Represents the person pose possibility type.	
Range / Symbol	***Limit***	***Description***
kHead	0x00	The person head pose.
kUpperBody	0x01	The person upper body pose.
kLeftHand	0x02	Described angles are measured between the normal to the hand palm relative to the ego-vehicle axis system.
kRightHand	0x03	Described angles are measured between the normal to the hand palm relative to the ego-vehicle axis system.
kLeftLowerArm	0x04	Described angles are measured between long axis of the left lower arm relative to the ego-vehicle axis system.
kRightLowerArm	0x05	Described angles are measured between long axis of the right lower arm relative to the ego-vehicle axis system.
kRightUpperLeg	0x06	Described angles are measured between long axis of the right upper leg relative to the ego-vehicle axis system.
kLeftLowerLeg	0x07	Described angles are measured between long axis of the left lower leg relative to the ego-vehicle axis system.
kRightLowerLeg	0x08	Described angles are measured between long axis of the right lower leg relative to the ego-vehicle axis system.
kLeftLowerLeg	0x09	Described angles are measured between long axis of the left lower leg relative to the ego-vehicle axis system.
kLeftFoot	0x0A	Described angles are measured between long axis of the left foot relative to the ego-vehicle axis system.
kRightFoot	0x0B	Described angles are measured between long axis of the right foot relative to the ego-vehicle axis system.
kUnknown	0x0C	The person pose is unknown.

⌋*(RS_ADI_00001, RS_ADI_00002, RS_ADI_00003, RS_ADI_00004, RS_ADI_00012, RS_ADI_00013, RS_ADI_00014)*

9.1.39 ObjectLaneAssociation

[SWS_ADI_00211]{DRAFT} ⌈

Name	ObjectLaneAssociation	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Represents an association of the object to neighbouring lanes	
Range / Symbol	***Limit***	***Description***
kEgoLine	0x00	Object is in ego lane.

▽

△

kLeftNeighbouringLane	0x01	Object is in left neighbouring lane.
kRightNeighbouringLane	0x02	Object is in right neighbouring lane.
kEgoRightLane	0x03	Object is located between ego and right neighbouring lane.
kEgoLeftLane	0x04	Object is located between ego and left neighbouring lane.
kUnknown	0x05	Object's lane association is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.40 Angle2ObjectEdge2Line

[SWS_ADI_00212]{DRAFT} ⌈

Name	Angle2ObjectEdge2Line
Kind	STRUCTURE
Subelements	LeftEdgeRightLane `float` RightEdgeLeftLane `float`
Derived from	-
Description	Represents the Angles to object edge to lane. The unit is {rad,rad}.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.41 Angle2ObjectEdge2LineError

[SWS_ADI_00213]{DRAFT} ⌈

Name	Angle2ObjectEdge2LineError
Kind	STRUCTURE
Subelements	LeftEdgeRightLane `float` RightEdgeLeftLane `float`
Derived from	-
Description	Represents the Error value of the angles to object edge to lane. The unit is {rad,rad}.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.42 PercentageSideLane

[SWS_ADI_00214]{DRAFT} ⌈

Name	PercentageSideLane
Kind	STRUCTURE
Subelements	Left `uint_16` Right `uint_16`
Derived from	-
Description	Represents the percentage value of the object width in the corresponding {left, right} lane.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.43 PMObjectDynamics

[SWS_ADI_00215]{DRAFT} ⌈

Name	PMObjectDynamics
Kind	STRUCTURE
Subelements	PMORPs `ReferencePoints` (optional) PMOPosition `Point3D` PMOPositionError `Point3DError` PMOrientation `Orientation3D` (optional) PMOrientationError `Orientation3DError` (optional) PMOVelocity `Point3D` PMOVelocityError `Point3DError` (optional) PMOAcceleration `Point3D` (optional) PMOAccelerationError `Point3DError` (optional) KinematicPoint `Point3D` (optional) KPError `Point3DError` (optional) KPOrientation `float` (optional) KPOrentationError `float` (optional) PMORoadLevel `Roadlevel` (optional) PMOMovementStatus `MovementStatus` (optional)
Derived from	-
Description	Represents the dynamics of the possible moving objects.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.44 ObjectStatus

[SWS_ADI_00216]{DRAFT} ⌈

Name	ObjectStatus
Kind	STRUCTURE
Subelements	ObjectID `uint_16` (optional) GroupingObjectID `uint_8` (optional) Age `uint_64` Observation `uint_32` (optional) TrackQuality `uint_16` (optional) PMOMeasurementSatus `MeasurementSatus`
Derived from	-
Description	Represents the dynamics of the possible moving objects.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.45 PMOBoundingBox

[SWS_ADI_00217]{DRAFT} ⌈

Name	PMOBoundingBox
Kind	STRUCTURE
Subelements	BoxDimension `DimensionBox` BoxError `DimensionBoxError` (optional) GroundClearance `float` (optional) IncludedGeometricStructures `GeometricStructures` (optional)
Derived from	-
Description	Represents the bounding box information of the possible moving objects.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.46 PMOTypeCertainty

[SWS_ADI_00218]{DRAFT} ⌈

Name	PMOTypeCertainty
Kind	STRUCTURE
Subelements	ExistenceProbability `uint_16` NoValidClassification `uint_8` PMOCertaintyList `PMOCertaintyVector`
Derived from	-

▽

△

Description	Represents the certainty information regarding possible moving object types list.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.47 PMOCertainty

[SWS_ADI_00219]{DRAFT} ⌈

Name	PMOCertainty
Kind	STRUCTURE
Subelements	PMOClassificationType `PMOClassificationType` PMOTypeProbabilityValue `float`
Derived from	-
Description	Represents the certainty information regarding a possible moving object type.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.48 PMOCertaintyVector

[SWS_ADI_00220]{DRAFT} ⌈

Name	PMOCertaintyVector
Kind	VECTOR
Subelements	`PMOCertainty`
Derived from	-
Description	Represents a list of PMOCertainty.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.49 Light

[SWS_ADI_00221]{DRAFT} ⌈

Name	Light
Kind	STRUCTURE
Subelements	PMOLightType `LightType` PMOLightStatus `LightStatus`
Derived from	-
Description	Represents the light information including light type and light status.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.50 LightVector

[SWS_ADI_00222]{DRAFT} ⌈

Name	LightVector
Kind	VECTOR
Subelements	`Light`
Derived from	-
Description	Represents a list of light.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.51 PMOLights

[SWS_ADI_00223]{DRAFT} ⌈

Name	PMOLights
Kind	STRUCTURE
Subelements	noValidPMOLights `uint_8` PMOLightList `LightVector`
Derived from	-
Description	Represents the lights information of the tracked object.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.52 Pose

[SWS_ADI_00224]{DRAFT} ⌈

Name	Pose
Kind	STRUCTURE
Subelements	PoseType `PersonPoseType` PersonPoseInfo `Orientation3D` PoseError `Orientation3DError` (optional)
Derived from	-
Description	Represents the Pose information including pose type and pose status.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.53 PoseVector

[SWS_ADI_00225]{DRAFT} ⌈

Name	PoseVector
Kind	VECTOR
Subelements	`Pose`
Derived from	-
Description	Represents a list of Pose.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.54 PMOPersonPose

[SWS_ADI_00226]{DRAFT} ⌈

Name	PMOPersonPose
Kind	STRUCTURE
Subelements	noValidPersonPoses `uint_8` Poselist `PoseVector`
Derived from	-
Description	Represents the poses information of the tracked person.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.55 LaneInfo

[SWS_ADI_00227]{DRAFT} ⌈

Name	LaneInfo
Kind	STRUCTURE
Subelements	LaneAssociation `ObjectLaneAssociation` Angles `Angle2ObjectEdge2Line` (optional) AngleError `Angle2ObjectEdge2LineError` (optional) Percentage `PercentageSideLane` (optional)
Derived from	-
Description	Represents the lane related information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.56 MotionInfo

[SWS_ADI_00228]{DRAFT} ⌈

Name	MotionInfo
Kind	STRUCTURE
Subelements	AngularPositionA `float` angularVelocityA `float`
Derived from	-
Description	Represents the motion related information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.57 PMObject

[SWS_ADI_00229]{DRAFT} ⌈

Name	PMObject
Kind	STRUCTURE
Subelements	PMObjectInfo `ObjectStatus` ObjectDynamics `PMObjectDynamics` ObjectBoundingBox `PMOBoundingBox` (optional) ObjectTypeCertainty `PMOTypeCertainty` ObjectLights `PMOLights` (optional) PersonPoses `PMOPersonPose` (optional) LaneInforation `LaneInfo` (optional) PMOMotionInfo `MotionInfo` (optional)

▽

△

	△ ScaleChange `float` (optional) EntityRCS `float` (optional) EntityLidarReflectivity `float` (optional)
Derived from	-
Description	Represents the possible moving objects tracked by a camera,radar, lidar or Ultrasonic.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.58 PMObjectVector

[SWS_ADI_00230]{DRAFT} ⌈

Name	PMObjectVector
Kind	VECTOR
Subelements	`PMObject`
Derived from	-
Description	Represents a list of PMO.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.59 PMObjects

[SWS_ADI_00231]{DRAFT} ⌈

Name	PMObjects
Kind	STRUCTURE
Subelements	PMOHeader `PMOHeader` PMOCap `uint_16` (optional) PMOStatus `RecognizedStatus` (optional) NoValidPMO `uint_16` PMObjectList `PMObjectVector`
Derived from	-
Description	Represents the possible moving object informaiton provided by a camera,lidar, radar or Ultrasonic sensor.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.60 RoadType

[SWS_ADI_00300]{DRAFT} ⌈

Name	RoadType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	To identify the ego-vehicle relevant type of the road.	
Range / Symbol	***Limit***	***Description***
kHighway	0x00	Represents the highway.
kRural	0x01	Represents the rural road.
kCity	0x02	Represents the city road.
kOffRoad	0x03	Represents off the road.
kUnknown	0x04	Represents the road type unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.61 RoadSurfaceType

[SWS_ADI_00301]{DRAFT} ⌈

Name	RoadSurfaceType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	To identify the ego-vehicle relevant type of the road surface.	
Range / Symbol	***Limit***	***Description***
kRomanRoad	0x00	Roman Road.
kOffRoad	0x01	Off the road.
kFlat	0x02	Flat surface.
kBumpy	0x03	Bumpy surface.
kUnknown	0x04	Surface type is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.62 RoadConditionType

[SWS_ADI_00302]{DRAFT} ⌈

Name	RoadConditionType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	To identify the ego-vehicle relevant type of the road surface.	
Range / Symbol	**Limit**	**Description**
kDry	0x00	Dry Road.
kWet	0x01	Wet road.
kSnow	0x02	Snow road.
kIce	0x03	Ice Road.
kUnknown	0x04	Road Condition is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.63 RoadMarkingType

[SWS_ADI_00303]{DRAFT} ⌈

Name	RoadMarkingType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	To identify the ego-vehicle relevant type of the road surface.	
Range / Symbol	**Limit**	**Description**
kSolid	0x00	The road marking is solid. It could also be a stop line.
kCentreLineDashedMarking	0x01	The centre line road marking is dashed.
kEdgeLineDashedMarking	0x02	The edge line road marking is dashed.
kTriangular	0x03	The road marking is a line of triangles.
kDoubleLineSolid	0x04	The road marking has two lines and the most inner line (w.r.t. the ego-vehicle) is solid.
kCentreLineDoubleLineDashed	0x05	The centre line road marking has two lines and the most inner line (w.r.t. the ego-vehicle) is dashed.
kMultipleLineSolid	0x06	The road marking has more than two lines and the most inner line (w.r.t. the ego-vehicle) is solid.
kCentreLineMultipleLine Dashed	0x07	The centre line road marking has more than two lines and the most inner line (w.r.t. the ego-vehicle) is dashed.
kEdgeLineMultipleLineDashed	0x08	The edge line road marking has more than two lines and the most inner line (w.r.t. the ego-vehicle) is dashed.
kBottsDotsCatsEyes	0x09	The road marking consists of Botts' dots or cats' eyes.
kAttentionMarker	0x0A	The road marking is an attention marker e.g. US, China and Japan.
kHatched	0x0B	The edge line road marking is dashed.
kBox	0x0C	The road marking of a junction.
kColouredArea	0x0D	The road marking is a coloured area.
kArrow	0x0E	The road marking is an arrow.
kZebraCrossing	0x0F	The road marking is a zebra crossing / continental / ladder.

▽

△

kGenericSymbol	0x10	The edge line road marking has more than two lines and the most inner line (w.r.t. the ego-vehicle) is dashed.
kTrafficSignOnLane	0x11	The road marking is a traffic sign.
kGenericLine	0x12	The road marking is a generic line.
kParkingArea	0x13	The edge line road marking is dashed.
kTShapeMarkingBegin	0x14	The road marking is a parking T-shape beginning parking line.
kTShapeMarkingEnd	0x15	The road marking is a parking T-shape ending parking line.
kIShapeMarkingBegin	0x16	The road marking is a parking I-shape beginning parking line.
kIShapeMarkingEnd	0x17	The road marking is a zebra crossing / continental / ladder.
kLShapeMarkingBegin	0x18	The road marking is a parking L-shape beginning parking line.
kLShapeMarkingEnd	0x19	The road marking is a parking L-shape ending parking line.
kNets	0x1A	The road marking is a net, i.e. a non-stopping area.
kUnknown	0x1B	The road marking is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.64 ArrowDirection

[SWS_ADI_00304]{DRAFT} ⌈

Name	ArrowDirection	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	To identify estimated direction of the displayed arrow.	
Range / Symbol	***Limit***	***Description***
kForeward	0x00	Arrow has an estimated direction of 0 rad to 0.
kLeft	0x01	Arrow has an estimated direction of +pi/2 rad to +90.
kRight	0x02	Arrow has an estimated direction of -pi/2 rad to -90.
kStraightLeft	0x03	Arrow is straight left and has an estimated direction of +pi/2 rad to +90.
kStraightRight	0x04	Arrow is straight right and has an estimated direction of -pi/2 rad to -90.
kTurningPointLeft	0x05	Arrow has an estimated direction of +pi/2 rad to +180.
kTurningPointRight	0x06	Arrow has an estimated direction of -pi/2rad to -180.
k45DegLeft	0x07	Arrow has an estimated direction of +pi/4 rad to +45.
k45DegRight	0x08	Arrow has an estimated direction of -pi/4 rad to -45.
kNoArrow	0x09	No arrow is present.
kUnknown	0x0A	The arrow direction is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.65 SignType

[SWS_ADI_00305]{DRAFT} ⌈

Name	SignType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	To identify the type of the sign as main traffic sign or road marking.	
Range / Symbol	***Limit***	***Description***
kStopSign	0x00	Stop sign.
kYieldSign	0x01	Yield sign.
kSpeedLimitSign	0x02	Speed limit sign.
kNoMainSign	0x03	No main sign, only supplementary signs.
kGreenArrowSign	0x04	Green Arrow Sign.
kHeightLimitSign	0x05	Height limit sign.
kEmptySign	0x06	Sign may be a changeable traffic sign without displaying a traffic sign symbol.
kUnknown	0x07	The Sign type is unknown.

⌋*(RS_ADI_00001, RS_ADI_00002, RS_ADI_00003, RS_ADI_00004, RS_ADI_00012, RS_ADI_00013, RS_ADI_00014)*

9.1.66 SignValueUnit

[SWS_ADI_00306]{DRAFT} ⌈

Name	SignValueUnit	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	To identify the value unit of the linked sign.	
Range / Symbol	***Limit***	***Description***
kKilometrePerHour	0x00	Kilometre per hour
kMilePerHour	0x01	Mile per hour
kMeter	0x02	Meter
kKilometre	0x03	Kilometre
kFeet	0x04	Feet
kMile	0x05	Mile
kMetricTon	0x06	Metric Ton
kShortTon	0x07	Short Ton
kLongTon	0x08	LongTon.
kMinute	0x09	Minute
kHour	0x0A	Hour
Day	0x0B	Day
kWeekday	0x0C	Weekday

▽

△

kPercentage	0x0D	Percentage
kUnknown	0x0E	Unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.67 ConnectionType

[SWS_ADI_00307]{DRAFT} ⌈

Name	ConnectionType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	To identify the type of connection of at least two road markings, polylines or polynomials.	
Range / Symbol	***Limit***	***Description***
kMerge	0x00	The connection of road markings is a merge of road markings.
kSplit	0x01	The connection of road markings is a split of road markings.
kInterconnection	0x02	The connection of road markings is an interconnection of road markings.
kExtension	0x03	The connection of road markings is an extension of two road markings.
kUnknown	0x04	The connection type is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.68 PolynomialCoefficient

[SWS_ADI_00308]{DRAFT} ⌈

Name	PolynomialCoefficient
Kind	STRUCTURE
Subelements	C0 `float` C1 `float` C2 `float` C3 `float`
Derived from	-
Description	Calculated coefficient. (m, 1, 1/m, 1/m2)

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.69 RoadBoundaryProperties

[SWS_ADI_00309]{DRAFT} ⌈

Name	RoadBoundaryProperties
Kind	STRUCTURE
Subelements	ColourValueList `ColourValueVector` ColourConfidence `ProPercentage` (optional)
Derived from	-
Description	Road Boundary Colour information, only relevant to Camera.

⌋*(RS_ADI_00001, RS_ADI_00012, RS_ADI_00013, RS_ADI_00014)*

9.1.70 PolynomialRange

[SWS_ADI_00311]{DRAFT} ⌈

Name	PolynomialRange
Kind	STRUCTURE
Subelements	Start `float` End `float`
Derived from	-
Description	Valid range of the polynomial [x Start, x End].(m, m)

⌋*(RS_ADI_00001, RS_ADI_00002, RS_ADI_00003, RS_ADI_00004, RS_ADI_00012, RS_ADI_00013, RS_ADI_00014)*

9.1.71 SupportedDataRange

[SWS_ADI_00312]{DRAFT} ⌈

Name	SupportedDataRange
Kind	STRUCTURE
Subelements	Start `float` End `float`
Derived from	-
Description	Supported range of the polynomial [x Start, x End] covered with measured points.

⌋*(RS_ADI_00001, RS_ADI_00002, RS_ADI_00003, RS_ADI_00004, RS_ADI_00012, RS_ADI_00013, RS_ADI_00014)*

9.1.72 InterpolationMethod

[SWS_ADI_00313]{DRAFT} ⌈

Name	InterpolationMethod	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	To identify the type of connection of at least two road markings, polylines or polynomials.	
Range / Symbol	***Limit***	***Description***
kLinear	0x00	Linear interpolation between two sequential points.
kSpline	0x01	Spline interpolation between two sequential points.
kCubic	0x02	Cubic interpolation between two sequential points.
kUnknown	0x03	Interpolation Method is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.73 VertexPointConfidence

[SWS_ADI_00314]{DRAFT} ⌈

Name	VertexPointConfidence
Kind	STRUCTURE
Subelements	Vpcx `float` Vpcy `float` Vpcz `float`
Derived from	-
Description	Measured longitudinal, lateral and vertical distance of the vertex confidence.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.74 RoadBoundaryType

[SWS_ADI_00315]{DRAFT} ⌈

Name	RoadBoundaryType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Indicated the type of the road boundary.	
Range / Symbol	***Limit***	***Description***

▽

△

kGuardrail	0x00	The road boundary is a guardrail.
kFence	0x01	The road boundary is a fence.
kWand	0x02	The road boundary is a wand, a wall, a building, etc.
kBarrier	0x03	The road boundary is a barrier.
kTensionCableSystem	0x04	The road boundary is a tension cable system.
kRoadEdge	0x05	The road boundary is a road edge e.g. grass, vegetation, sand, gravel, soil, etc.
kCurb	0x06	The road boundary is a curb stone.
kUnclassifiedElevated	0x07	The road boundary is an unclassified elevated structure.
kUnknown	0x08	The Road boundary type is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.75 LaneBoundaryRole

[SWS_ADI_00316]{DRAFT} ⌈

Name	LaneBoundaryRole	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Definition of the road boundary role w.r.t. the ego-vehicle lane.	
Range / Symbol	***Limit***	***Description***
kHostLeft	0x00	The lane boundary role is left to the ego-vehicle.
kHostRight	0x01	The lane boundary role is right to the ego-vehicle.
kAdjacentLeft	0x02	The lane boundary role is to the left adjacent ego-vehicle lane.
kAdjacentRight	0x03	The lane boundary role is to the right adjacent ego-vehicle lane.
kNotValid	0x04	The road boundary is not valid as lane boundary.
kUnknown	0x05	The road boundary role is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.76 RoadSurfaceClass

[SWS_ADI_00318]{DRAFT} ⌈

Name	RoadSurfaceClass
Kind	STRUCTURE
Subelements	ROSurfaceType `RoadSurfaceType` SurfaceProbability `ProPercentage`
Derived from	-
Description	Represents the road suface type and probability.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.77 ProPercentage

[SWS_ADI_00319]{DRAFT} ⌈

Name	ProPercentage
Kind	TYPE_REFERENCE
Derived from	`float`
Description	Represet probability in percentage.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.78 RoadSurfaceClassVector

[SWS_ADI_00320]{DRAFT} ⌈

Name	RoadSurfaceClassVector
Kind	VECTOR
Subelements	`RoadSurfaceClass`
Derived from	-
Description	Represents a list of Road Surface.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.79 RoadSurfaceCon

[SWS_ADI_00321]{DRAFT} ⌈

Name	RoadSurfaceCon
Kind	STRUCTURE
Subelements	ROSurfaceConType `RoadConditionType` RoadConditionProbability `ProPercentage`
Derived from	-
Description	Represents the road suface condition and probability.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.80 RoadSurfaceConVector

[SWS_ADI_00322]{DRAFT} ⌈

Name	RoadSurfaceConVector
Kind	VECTOR
Subelements	`RoadSurfaceCon`
Derived from	-
Description	Represents a list of Road Surface condition.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.81 RoadSurface

[SWS_ADI_00323]{DRAFT} ⌈

Name	RoadSurface
Kind	STRUCTURE
Subelements	ROType `RoadType` NoRoadSurface `uint8_t` ROsurfaceList `RoadSurfaceClassVector` NoConditionClass `uint8_t` (optional) ROsurfaceConList `RoadSurfaceConVector` SurfaceRoughness `float` (optional)
Derived from	-
Description	Represents the road suface Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.82 ColourValueVector

[SWS_ADI_00324]{DRAFT} ⌈

Name	ColourValueVector
Kind	VECTOR
Subelements	`float`
Derived from	-
Description	Represents a list of Colour Value.

⌋*(RS_ADI_00001, RS_ADI_00002, RS_ADI_00003, RS_ADI_00004, RS_ADI_00012, RS_ADI_00013, RS_ADI_00014)*

9.1.83 ROMarkingT

[SWS_ADI_00325]{DRAFT} ⌈

Name	ROMarkingT
Kind	STRUCTURE
Subelements	ROMarkingType `RoadMarkingType` RoadMarkingTypeConfidence `ProPercentage` ArrowOrientation `float` (optional) ArrowDirect `ArrowDirection` SignT `SignType` (optional) SignTypeConfidence `ProPercentage` (optional) SignValue `float` (optional) SignUnit `SignValueUnit` (optional) ColourValueList `ColourValueVector` (optional) ColourConfidence `ProPercentage` (optional)
Derived from	-
Description	Represents the road marking type Information.

⌋*(RS_ADI_00001, RS_ADI_00002, RS_ADI_00003, RS_ADI_00004, RS_ADI_00012, RS_ADI_00013, RS_ADI_00014)*

9.1.84 ROMarkingTVector

[SWS_ADI_00326]{DRAFT} ⌈

Name	ROMarkingTVector
Kind	VECTOR
Subelements	`ROMarkingT`
Derived from	-
Description	Represents a list of Road Marking type.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.85 RoadMarkingTypeInfo

[SWS_ADI_00327]{DRAFT} ⌈

Name	RoadMarkingTypeInfo
Kind	STRUCTURE
Subelements	NoRoadMarkingType `uint8_t` ROMarkingTList `ROMarkingTVector`
Derived from	-
Description	Represents the road marking type Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.86 ConnectionInfo

[SWS_ADI_00328]{DRAFT} ⌈

Name	ConnectionInfo
Kind	STRUCTURE
Subelements	Type `ConnectionType` ConnectionID `uint8_t`
Derived from	-
Description	Represents the road marking connection information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.87 ConnectionInfoVector

[SWS_ADI_00329]{DRAFT} ⌈

Name	ConnectionInfoVector
Kind	VECTOR
Subelements	`ConnectionInfo`
Derived from	-
Description	Represents a list of Road marking connection information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.88 PolynomialsInfo

[SWS_ADI_00330]{DRAFT} ⌈

Name	PolynomialsInfo
Kind	STRUCTURE
Subelements	Coefficient `PolynomialCoefficient` PolynomialError `float` (optional) Range `PolynomialRange` WidthPolynomials `float` (optional) WidthPolynomialsError `float` (optional) WidthPolynomialsConfidence `ProPercentage` (optional) HeightPolynomials `float` (optional) HeightPolynomialsError `float` (optional) HeightPolynomialsConfidence `ProPercentage` (optional) NoDataRanges `uint8_t` (optional) DataRangeList `SupportedDataRangeVector` (optional)
Derived from	-
Description	Represents the road marking type Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.89 SupportedDataRangeVector

[SWS_ADI_00331]{DRAFT} ⌈

Name	SupportedDataRangeVector
Kind	VECTOR
Subelements	`SupportedDataRange`
Derived from	-

▽

△

Description	Represents a list of supported data range information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.90 PolynomialsInfoVector

[SWS_ADI_00332]{DRAFT} ⌈

Name	PolynomialsInfoVector
Kind	VECTOR
Subelements	`PolynomialsInfo`
Derived from	-
Description	Represents a list of polynomials information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.91 Polynomials

[SWS_ADI_00333]{DRAFT} ⌈

Name	Polynomials
Kind	STRUCTURE
Subelements	NoConnections `uint8_t` ConnectionList `ConnectionInfoVector` NoPolynomials `uint8_t` PolynomialsList `PolynomialsInfoVector`
Derived from	-
Description	Represents the Polynomials related Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.92 VertexPointInfo

[SWS_ADI_00334]{DRAFT} ⌈

Name	VertexPointInfo
Kind	STRUCTURE
Subelements	VertexPoint `Point3D` VertexPointError `Point3DError` Confidence `VertexPointConfidence` (optional) WidthVertex `float` (optional) WidthVertexError `float` (optional) WidthVertexConfidence `ProPercentage` (optional) HeightVertex `float` (optional) HeightVertexError `float` (optional) HeightVertexConfidence `ProPercentage` (optional)
Derived from	-
Description	Represents the vertex point Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.93 VertexPointInfoVector

[SWS_ADI_00335]{DRAFT} ⌈

Name	VertexPointInfoVector
Kind	VECTOR
Subelements	`VertexPointInfo`
Derived from	-
Description	Represents a list of vertex points information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.94 PolylineInfo

[SWS_ADI_00336]{DRAFT} ⌈

Name	PolylineInfo
Kind	STRUCTURE
Subelements	NoVertices `uint8_t` Vertexpointlist `VertexPointInfoVector`
Derived from	-
Description	Represents the Polyline related Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.95 PolylineinfoVector

[SWS_ADI_00337]{DRAFT} ⌈

Name	PolylineinfoVector
Kind	VECTOR
Subelements	`PolylineInfo`
Derived from	-
Description	Represents a list of Polyline information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.96 Polylines

[SWS_ADI_00338]{DRAFT} ⌈

Name	Polylines
Kind	STRUCTURE
Subelements	NoConnections `uint8_t` ConnectionList `ConnectionInfoVector` PInterpolationMethod `InterpolationMethod` NoPolynomials `uint8_t` PolylineList `PolylineinfoVector`
Derived from	-
Description	Represents the Polylines Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.97 Roadmarking

[SWS_ADI_00339]{DRAFT} ⌈

Name	Roadmarking
Kind	STRUCTURE
Subelements	ROStatus `ObjectStatus` ExistancePro `ProPercentage` ROmarkingTypeInfo `RoadMarkingTypeInfo` ROPolynomialsInfor `Polynomials` (optional) ROPolylines `Polylines` (optional)

▽

△

Derived from	-
Description	Represents the road marking Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.98 RoadmarkingVector

[SWS_ADI_00340]{DRAFT} ⌈

Name	RoadmarkingVector
Kind	VECTOR
Subelements	`Roadmarking`
Derived from	-
Description	Represents a list of road marking object information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.99 Roadmarkings

[SWS_ADI_00341]{DRAFT} ⌈

Name	Roadmarkings
Kind	STRUCTURE
Subelements	RecognizedRoadMarkingCap `uint16_t` (optional) RecognizedRoadMarkingStatus `RecognizedStatus` (optional) NoValidRoadMarking `uint16_t` ROmarkinglist `RoadmarkingVector`
Derived from	-
Description	Represents the road marking Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.100 ROBoundaryType

[SWS_ADI_00342]{DRAFT} ⌈

Name	ROBoundaryType
Kind	STRUCTURE
Subelements	Type `RoadBoundaryType` RoadBoundaryTypeConfidence `ProPercentage` Role `LaneBoundaryRole` LaneBoundaryRoleConfidence `ProPercentage` RBProperties `RoadBoundaryProperties` (optional)
Derived from	-
Description	Represents the road boundary type Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.101 ROBoundaryTypeVector

[SWS_ADI_00343]{DRAFT} ⌈

Name	ROBoundaryTypeVector
Kind	VECTOR
Subelements	`ROBoundaryType`
Derived from	-
Description	Represents a list of road boundary type object information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.102 ROBoundaryTypeInfo

[SWS_ADI_00344]{DRAFT} ⌈

Name	ROBoundaryTypeInfo
Kind	STRUCTURE
Subelements	NoRoadBoundaryType `uint8_t` ROBoundaryTypeList `ROBoundaryTypeVector`
Derived from	-
Description	Represents the road boundary type Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.103 RoadBoundaryInfo

[SWS_ADI_00345]{DRAFT} ⌈

Name	RoadBoundaryInfo
Kind	STRUCTURE
Subelements	ROBoundaryStatus `ObjectStatus` ExistancePro `ProPercentage` ROBoundaryInfo `ROBoundaryTypeInfo` RBPolynomialsInfor `Polynomials` (optional) RBPolylines `Polylines` (optional)
Derived from	-
Description	Represents the road boundary Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.104 RoadBoundaryInfoVector

[SWS_ADI_00346]{DRAFT} ⌈

Name	RoadBoundaryInfoVector
Kind	VECTOR
Subelements	`RoadBoundaryInfo`
Derived from	-
Description	Represents a list of road boundary object information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.105 RoadBoundaries

[SWS_ADI_00317]{DRAFT} ⌈

Name	RoadBoundaries
Kind	STRUCTURE
Subelements	RecognizedRoadBoundariesCap `uint16_t` (optional) RecognizedRoadBoundariesStatus `RecognizedStatus` (optional) NoValidRoadBoundaries `uint16_t` RoadBoundaryList `RoadBoundaryInfoVector`
Derived from	-

▽

△

Description	Represents the road boundaries Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.106 RObjects

[SWS_ADI_00310]{DRAFT} ⌈

Name	RObjects
Kind	STRUCTURE
Subelements	ROheader `GeneralHeader` RoadSurface `RoadSurface` (optional) ROmarkings `Roadmarkings` (optional) ROBoundaries `RoadBoundaries` (optional)
Derived from	-
Description	Represents the road object informaiton provided by a camera,lidar, radar or Ultrasonic sensor.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.107 LandmarkClass

[SWS_ADI_00401]{DRAFT} ⌈

Name	LandmarkClass	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Classification of the general landmark.	
Range / Symbol	***Limit***	***Description***
kBridge	0x00	Bridge as general landmark.
kBeacon	0x01	Beacon as general landmark.
kCone	0x02	Cone as general landmark.
kBarrel	0x03	TBarrel as general landmark.
kGuidePost	0x04	Guide post as general landmark.
kLampPost	0x05	Lamp post as general landmark.
kVerticalStructure	0x06	Vertical structure as general landmark.
kOverheadObject	0x07	Overhead object as general landmark.
kRectangularStructure	0x08	Rectangular structure as general landmark.

▽

△

kTunnel	0x09	Tunnel as general landmark.
kReflector	0x0A	Reflector as general landmark.
kUnknown	0x0B	The landmark class is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.108 SignGeometry

[SWS_ADI_00402]{DRAFT} ⌈

Name	SignGeometry	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	The shape of the sign.	
Range / Symbol	***Limit***	***Description***
kCircle	0x00	Circle shape as sign geometry.
kTriangleTop	0x01	Triangle with tip pointing downwards as sign geometry.
kTriangleDown	0x02	Triangle with tip pointing upwards as sign geometry.
kSquare	0x03	Square shape as sign geometry.
kPole	0x04	Pole shape as sign geometry.
kRectangle	0x05	Rectangle shape as sign geometry.
kPlate	0x06	Plate with multiple traffic information.
kDiamond	0x07	Diamond shape as sign geometry.
kArrowLeft	0x08	Arrow left five edge shape as sign geometry.
kArrowRight	0x09	Arrow right five edge shape as sign geometry.
kOctagon	0x0A	Octagon shape as sign geometry.
kUnknown	0x0B	The sign geometry is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.109 VariableMessageSign

[SWS_ADI_00403]{DRAFT} ⌈

Name	VariableMessageSign
Kind	TYPE_REFERENCE
Derived from	`uint8_t`

▽

△

Description	Describing whether or not the message of the sign is variable/changeable, static or if that information is not available.	
Range / Symbol	***Limit***	***Description***
kStatic	0x00	Message sign is no variable message sign.
kVariable	0x01	Spline interpolation between two sequential points.
kNotAvailable	0x02	Message sign is out of service.
kUnknown	0x03	Message changable is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.110 OutOfService

[SWS_ADI_00404]{DRAFT} ⌈

Name	OutOfService	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Traffic sign is out of service, e.g. the symbol is crossed out.	
Range / Symbol	***Limit***	***Description***
kInService	0x00	Message sign is in service.
kFullOutOfService	0x01	Message sign is full out of service.
kPartlyOutOfService	0x02	Part of the message sign is out of service.
kOutOfView	0x03	Message sign is rotated.
kUnknown	0x04	Message sign service status is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.111 LaneRelevance

[SWS_ADI_00405]{DRAFT} ⌈

Name	LaneRelevance	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Information if the sign is relevant for the ego-vehicle's lane, the nearest lane to the ego-vehicle or other relevant lanes.	
Range / Symbol	***Limit***	***Description***
kOnTrack	0x00	Relevant on track of ego-vehicle.

▽

△

kNextLaneLeft	0x01	Relevant for the next lane to the ego-vehicle on the left site.
kNextLaneRight	0x02	Relevant for the next lane to the ego-vehicle on the right site.
kNextNextLaneLeft	0x03	Relevant for the second next lane to the ego-vehicle on the left site.
kNextNextLaneRight	0x04	Relevant for the second next lane to the ego-vehicle on the right site.
kOnTrackAndNextLaneLeft	0x05	Relevant on track of ego-vehicle and the next left lane.
kOnTrackAndNextLaneRight	0x06	Relevant on track of ego-vehicle and the next right lane.
kMostLeftLane	0x07	Relevant for the leftest lane.
kMostRightLane	0x08	Relevant for the rightest lane.
kAllLanes	0x09	Relevant for all lanes, lane to the right and left site and on track.
kOtherLane	0x0A	Relevant for another far lane.
kUnknown	0x0B	The lane relevance is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.112 SupplementarySignType

[SWS_ADI_00406]{DRAFT} ⌈

Name	SupplementarySignType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	The type of the sign.	
Range / Symbol	***Limit***	***Description***
kValidInformationBegin	0x00	Begin of valid zone.
kValidInformationEnd	0x01	End of valid zone.
kFrost	0x02	Frost sign.
kWetRoad	0x03	Wet road sign.
kDistance	0x04	Distance information.
kLimitation	0x05	Limitation information.
kUnknown	0x06	Supplementary Sign type is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.113 RelativePosition

[SWS_ADI_00407]{DRAFT} ⌈

Name	RelativePosition	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	The relative position of the supplemental sign w.r.t. its main sign.	
Range / Symbol	**Limit**	**Description**
kAbove	0x00	Supplementary sign is above the main sign.
kLeft	0x01	Message sign is full out of service.
kBelow	0x02	Supplementary sign is below the main sign.
kRight	0x03	Supplementary sign is right of the main sign.
kUnknown	0x04	Relative position is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.114 StructureLightType

[SWS_ADI_00408]{DRAFT} ⌈

Name	StructureLightType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	The traffic light can have different shapes.	
Range / Symbol	**Limit**	**Description**
kVertical3	0x00	Three light sources vertical.
kHorizontal3	0x01	Three light sources horizontal.
kDogHouse	0x02	Multi light sources.
kUnknown	0x03	Structure is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.115 ColourType

[SWS_ADI_00409]{DRAFT} ⌈

Name	ColourType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Colour of light spot.	
Range / Symbol	**Limit**	**Description**

▽

△

kRed	0x00	The spot colour is red.
kYellow	0x01	The spot colour is yellow.
kGreen	0x02	The spot colour is green.
kWhite	0x03	The spot colour is white.
kUnknown	0x04	The spot colour is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.116 LightModeType

[SWS_ADI_00410]{DRAFT} ⌈

Name	LightModeType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	The light's mode.	
Range / Symbol	***Limit***	***Description***
kContinuous	0x00	Light source is continuous on.
kBlinking	0x01	One light source is blinking
kTurnedOff	0x02	Light source is turned off.
kCounting	0x03	Light source with counting.
kUnknown	0x04	Light Mode type is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.117 LightShape

[SWS_ADI_00411]{DRAFT} ⌈

Name	LightShape	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	The light's shape.	
Range / Symbol	***Limit***	***Description***
kNoShape	0x00	No additional shape.
kArrowStraightAhead	0x01	Arrow straight ahead shape.
kArrowLeft	0x02	Arrow left shape.
kArrowDiagLeft	0x03	Arrow diagonal left shape.

▽

△

kArrowStraightAheadLeft	0x04	Arrow straight ahead and arrow left shape.
kArrowRight	0x05	Arrow right shape.
kArrowDiagRight	0x06	Arrow diagonal right shape.
kArrowStraightAheadRight	0x07	Arrow straight ahead and arrow right shape.
kArrowLeftRight	0x08	Arrow left and arrow right shape.
kArrowDown	0x09	Arrow down shape.
kArrowDownLeft	0x0A	Arrow U-turn left shape.
kArrowDownRight	0x0B	Arrow U-turn right shape.
kCross	0x0C	Cross figure.
kPedestrian	0x0D	Pedestrian figure.
kWalk	0x0E	Text walk figure.
kDontWalk	0x0F	Text don't walk figure.
kBicycle	0x10	Bicycle figure.
kPedestrianAndBicycle	0x11	Pedestrian and bicycle figure.
kCountdownSecond	0x12	Countdown in seconds figure.Signal Light shape value contains the value in s.
kCountdownPercent	0x13	Countdown in percent figure. Signal Light shape value contains the value in %.
kTram	0x14	Tram figure.
kBus	0x15	Bus figure.
kBusAndTram	0x16	Bus and Tram figure.
kUnknown	0x17	The light shape is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.118 LandmarkType

[SWS_ADI_00412]{DRAFT} ⌈

Name	LandmarkType
Kind	STRUCTURE
Subelements	GeneralLandMarkType `LandmarkClass` LandmarkClassProbability `ProPercentage`
Derived from	-
Description	Represents the general landmark type and probability.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.119 LandmarkTypeVector

[SWS_ADI_00413]{DRAFT} ⌈

Name	LandmarkTypeVector
Kind	VECTOR
Subelements	`LandmarkType`
Derived from	-
Description	Represents a list of Landmark type.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.120 LandMarkTypeCertainty

[SWS_ADI_00414]{DRAFT} ⌈

Name	LandMarkTypeCertainty
Kind	STRUCTURE
Subelements	ExistPro `ProPercentage` NoGeneralLandMarkClass `uint8_t` LandmarkTypelist `LandmarkTypeVector` LandmarkClassConfidence `ProPercentage`
Derived from	-
Description	Represents the ladmark type certainty Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.121 Landmarkposition

[SWS_ADI_00415]{DRAFT} ⌈

Name	Landmarkposition
Kind	STRUCTURE
Subelements	RPs `ReferencePoints` (optional) Position `Point3D` PositionError `Point3DError` OrientationVertical `float` (optional) OrientationVerticalError `float` (optional)
Derived from	-
Description	Represents the landmark position.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.122 BboxInfo

[SWS_ADI_00416]{DRAFT} ⌈

Name	BboxInfo
Kind	STRUCTURE
Subelements	BoxDimension `DimensionBox` BoxError `DimensionBoxError` (optional)
Derived from	-
Description	Represents the bounding box information of the static objects. This is only for camera

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.123 Landmark

[SWS_ADI_00417]{DRAFT} ⌈

Name	Landmark
Kind	STRUCTURE
Subelements	Landmarkstatus `ObjectStatus` LandmarkTypeCen `LandMarkTypeCertainty` LandmarkPos `Landmarkposition` LandmarkBoundingbox `BboxInfo` (optional)
Derived from	-
Description	Represents the landmark Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.124 LandmarkVector

[SWS_ADI_00418]{DRAFT} ⌈

Name	LandmarkVector
Kind	VECTOR
Subelements	`Landmark`
Derived from	-
Description	Represents a list of Landmark.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.125 GeneralLandmarks

[SWS_ADI_00419]{DRAFT} ⌈

Name	GeneralLandmarks
Kind	STRUCTURE
Subelements	RecognizedGeneralLandmarksCap `uint16_t` (optional) RecognizedGeneralLandmarksStatus `RecognizedStatus` (optional) NoValidGeneralLandmarks `uint16_t` LandmarkList `LandmarkVector`
Derived from	-
Description	Represents the general landmarks Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.126 MSignTypeInfo

[SWS_ADI_00420]{DRAFT} ⌈

Name	MSignTypeInfo
Kind	STRUCTURE
Subelements	ExistPro `ProPercentage` MSignType `SignType` SignTypeConfidence `ProPercentage`
Derived from	-
Description	Represents the main sign type certainty Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.127 MSignInfo

[SWS_ADI_00421]{DRAFT} ⌈

Name	MSignInfo
Kind	STRUCTURE

▽

△

Subelements	MSignValue `float` MSignUnit `SignValueUnit` MSignGeometry `SignGeometry` MVariableSign `VariableMessageSign` MOoS `OutOfService`
Derived from	-
Description	Represents the main sign Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.128 MSignPos

[SWS_ADI_00422]{DRAFT} ⌈

Name	MSignPos
Kind	STRUCTURE
Subelements	MSignLaneR `LaneRelevance` LaneRelevanceConfidence `ProPercentage` Position `Point3D` PositionError `Point3DError`
Derived from	-
Description	Represents the main sign position.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.129 SubObjectStatus

[SWS_ADI_00423]{DRAFT} ⌈

Name	SubObjectStatus
Kind	STRUCTURE
Subelements	Age `uint_64` Observation `uint_32` (optional) TrackQuality `uint_16` (optional) PMOMeasurementStatus `MeasurementStatus`
Derived from	-
Description	Represents the dynamics of the static object.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.130 SSignType

[SWS_ADI_00424]{DRAFT} ⌈

Name	SSignType
Kind	STRUCTURE
Subelements	ExistPro `ProPercentage` Type `SupplementarySignType` SupplementarSignTypeConfidence `ProPercentage`
Derived from	-
Description	Represents the Supplementary sign type Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.131 SSignInfo

[SWS_ADI_00425]{DRAFT} ⌈

Name	SSignInfo
Kind	STRUCTURE
Subelements	SSignValue `float` SSignUnit `SignValueUnit` SOoS `OutOfService`
Derived from	-
Description	Represents the Supplementary sign Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.132 SSignPos

[SWS_ADI_00426]{DRAFT} ⌈

Name	SSignPos
Kind	STRUCTURE
Subelements	SSRelativePosition `RelativePosition` RelativePosOrder `uint_8`
Derived from	-
Description	Represents the Supplementary sign position Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.133 SSign

[SWS_ADI_00427]{DRAFT} ⌈

Name	SSign
Kind	STRUCTURE
Subelements	SSStatus `SubObjectStatus` SStype `SSignType` SSInfo `SSignInfo` SSPos `SSignPos`
Derived from	-
Description	Represents the Supplementary sign Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.134 SSginVector

[SWS_ADI_00428]{DRAFT} ⌈

Name	SSginVector
Kind	VECTOR
Subelements	`SSign`
Derived from	-
Description	Represents a list of Supplementary sign.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.135 SSigns

[SWS_ADI_00429]{DRAFT} ⌈

Name	SSigns
Kind	STRUCTURE
Subelements	NoTrafficSupplementarSigns `uint8_t` SSignList `SSginVector`
Derived from	-
Description	Represents the Supplementary signs Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.136 TrafficSign

[SWS_ADI_00430]{DRAFT} ⌈

Name	TrafficSign
Kind	STRUCTURE
Subelements	MainSignstatus `ObjectStatus` MainSignType `MSignTypeInfo` MainSignInfo `MSignInfo` MainSignPos `MSignPos` SupplementarySigns `SSigns`
Derived from	-
Description	Represents the traffic sign Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.137 TrafficSignVector

[SWS_ADI_00431]{DRAFT} ⌈

Name	TrafficSignVector
Kind	VECTOR
Subelements	`TrafficSign`
Derived from	-
Description	Represents a list of traffic sign.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.138 TrafficSigns

[SWS_ADI_00432]{DRAFT} ⌈

Name	TrafficSigns
Kind	STRUCTURE
Subelements	RecognizedTrafficSignsCap `uint16_t` (optional) RecognizedTrafficSignsStatus `RecognizedStatus` (optional) NoValidTrafficSigns `uint16_t` TrafficSignsList `TrafficSignVector`
Derived from	-

▽

△

Description	Represents the traffic sings Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.139 TrafficLightTypeInfo

[SWS_ADI_00433]{DRAFT} ⌈

Name	TrafficLightTypeInfo
Kind	STRUCTURE
Subelements	ExistPro `ProPercentage` TLightType `StructureLightType` StructureLightTypeConfidence `ProPercentage`
Derived from	-
Description	Represents the traffic light type certainty Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.140 TrafficLightPos

[SWS_ADI_00434]{DRAFT} ⌈

Name	TrafficLightPos
Kind	STRUCTURE
Subelements	MinimumVisibilityDistance `uint16_t` RPs `ReferencePoints` (optional) Position `Point3D` PositionError `Point3DError` OrientationVertical `float` (optional) OrientationVerticalError `float` (optional)
Derived from	-
Description	Represents the traffic light position.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.141 ColourTypeInfo

[SWS_ADI_00435]{DRAFT} ⌈

Name	ColourTypeInfo
Kind	STRUCTURE
Subelements	CType `ColourType` ColourTypeProbability `ProPercentage`
Derived from	-
Description	Represents the colour type and probability.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.142 ColourTypeInfoVector

[SWS_ADI_00436]{DRAFT} ⌈

Name	ColourTypeInfoVector
Kind	VECTOR
Subelements	`ColourTypeInfo`
Derived from	-
Description	Represents a list of colour type.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.143 TrafficLightColour

[SWS_ADI_00437]{DRAFT} ⌈

Name	TrafficLightColour
Kind	STRUCTURE
Subelements	NoValidColourClass `uint8_t` ColourTypeList `ColourTypeInfoVector` ColourConfidence `ProPercentage`
Derived from	-
Description	Represents the trafic light colour type certainty Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.144 LightModeTypeInfo

[SWS_ADI_00438]{DRAFT} ⌈

Name	LightModeTypeInfo
Kind	STRUCTURE
Subelements	LModeType `LightModeType` LightModeProbability `ProPercentage`
Derived from	-
Description	Represents the traffic light mode type and probability.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.145 LightModeTypeInfoVector

[SWS_ADI_00439]{DRAFT} ⌈

Name	LightModeTypeInfoVector
Kind	VECTOR
Subelements	`ColourTypeInfo`
Derived from	-
Description	Represents a list of traffic light mode type.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.146 TrafficLightModeType

[SWS_ADI_00440]{DRAFT} ⌈

Name	TrafficLightModeType
Kind	STRUCTURE
Subelements	NoValidLightMode `uint8_t` LightModeTypeList `LightModeTypeInfoVector` LightModeConfidence `ProPercentage`
Derived from	-
Description	Represents the trafic light traffic light mode type certainty Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.147 TrafficLightSpotShape

[SWS_ADI_00441]{DRAFT} ⌈

Name	TrafficLightSpotShape
Kind	STRUCTURE
Subelements	TLightSpotShape `LightShape` LightShapeConfidence `ProPercentage` LightShapeValue `uint8_t` (optional)
Derived from	-
Description	Represents the trafic light spot shape Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.148 TrafficLightSpotPos

[SWS_ADI_00442]{DRAFT} ⌈

Name	TrafficLightSpotPos
Kind	STRUCTURE
Subelements	Position `Point3D` PositionError `Point3DError` (optional)
Derived from	-
Description	Represents the Traffic Light Spot position.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.149 TrafficLightSpot

[SWS_ADI_00443]{DRAFT} ⌈

Name	TrafficLightSpot
Kind	STRUCTURE
Subelements	TLightSpotStatus `SubObjectStatus` TLightSpotTypeCertainty `ProPercentage` TLightSpotColour `TrafficLightColour` TLightModeType `TrafficLightModeType` TLightSpotShape `TrafficLightSpotShape` TLightSpotPos `TrafficLightSpotPos`
Derived from	-
Description	Represents the traffic light sopt Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.150 TrafficLightSpotVector

[SWS_ADI_00444]{DRAFT} ⌈

Name	TrafficLightSpotVector
Kind	VECTOR
Subelements	`TrafficLightSpot`
Derived from	-
Description	Represents a list of traffic light spot.

⌋*(RS_ADI_00001, RS_ADI_00002, RS_ADI_00003, RS_ADI_00004, RS_ADI_00012, RS_ADI_00013, RS_ADI_00014)*

9.1.151 TrafficLightSpots

[SWS_ADI_00445]{DRAFT} ⌈

Name	TrafficLightSpots
Kind	STRUCTURE
Subelements	NoTrafficLightSpots `uint8_t` (optional) NoTrafficLightSpotsConfidence `ProPercentage` (optional) NoValidLightSpots `uint8_t` TrafficSLightSpotList `TrafficLightSpotVector`
Derived from	-
Description	Represents the trafic light spots Information.

⌋*(RS_ADI_00001, RS_ADI_00002, RS_ADI_00003, RS_ADI_00004, RS_ADI_00012, RS_ADI_00013, RS_ADI_00014)*

9.1.152 TrafficLight

[SWS_ADI_00446]{DRAFT} ⌈

Name	TrafficLight
Kind	STRUCTURE
Subelements	TLightStatus `ObjectStatus` TLightTypeCer `TrafficLightTypeInfo` TLightPos `TrafficLightPos` TLightBox `BboxInfo` (optional) TLightSpots `TrafficLightSpots`
Derived from	-

▽

△

Description	Represents the traffic light Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.153 TrafficLightVector

[SWS_ADI_00447]{DRAFT} ⌈

Name	TrafficLightVector
Kind	VECTOR
Subelements	`TrafficLight`
Derived from	-
Description	Represents a list of traffic light.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.154 TrafficLights

[SWS_ADI_00448]{DRAFT} ⌈

Name	TrafficLights
Kind	STRUCTURE
Subelements	RecognizedTrafficLightsCap `uint16_t` (optional) RecognizedTrafficLightsStatus `RecognizedStatus` (optional) NoValidTrafficLights `uint8_t` TrafficLightList `TrafficLightVector`
Derived from	-
Description	Represents the traffic lights Information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.155 StaticObjects

[SWS_ADI_00449]{DRAFT} ⌈

Name	StaticObjects
Kind	STRUCTURE
Subelements	Header `GeneralHeader` SOLandmarks `GeneralLandmarks` (optional) SOTrafficSigns `TrafficSigns` (optional) SOTrafficLights `TrafficLights` (optional)
Derived from	-
Description	Represents the static object informaiton provided by a camera,lidar, radar or Ultrasonic sensor.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.156 HorizontalAngle

[SWS_ADI_00501]{DRAFT} ⌈

Name	HorizontalAngle
Kind	STRUCTURE
Subelements	Begin `float` End `float`
Derived from	-
Description	FOV defined by opening angles in sensor XY-plane. (rad, rad)

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.157 VerticalAngle

[SWS_ADI_00502]{DRAFT} ⌈

Name	VerticalAngle
Kind	STRUCTURE
Subelements	Begin `float` End `float`
Derived from	-
Description	FOV defined by opening angles in sensor XZ-plane. (rad, rad)

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.158 SensorResolution

[SWS_ADI_00503]{DRAFT} ⌈

Name	SensorResolution
Kind	STRUCTURE
Subelements	Horizontal `float` Vertical `float`
Derived from	-
Description	The resolution of the sensor within the specified segment is the angle between the maxima of two adjacent measurement points. (rad, rad)

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.159 BeamDivergence

[SWS_ADI_00504]{DRAFT} ⌈

Name	BeamDivergence
Kind	STRUCTURE
Subelements	Horizontal `float` Vertical `float`
Derived from	-
Description	The Beam divergence {horizontal, vertical} of the sensor within the specified segment is the full width at half maximum (FWHM) of the beam (given as the angle in rad).(rad, rad)

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.160 SensorPerformanceStatus

[SWS_ADI_00505]{DRAFT} ⌈

Name	SensorPerformanceStatus
Kind	STRUCTURE
Subelements	HAngle `HorizontalAngle` VerticalAngle `VerticalAngle` SResolution `SensorResolution` (optional) BeamDivergence `BeamDivergence` (optional) RangeGain `ProPercentage` (optional) BStatus `BlockageStatus`

▽

△

Derived from	-
Description	Represents the sensor performance information.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.161 BlockageStatus

[SWS_ADI_00506]{DRAFT} ⌈

Name	BlockageStatus	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Defines the overall blockage of the FOV segment.	
Range / Symbol	***Limit***	***Description***
kFullBlockage	0x00	The sensor is completely blocked, no more feature and functionality working due to blockage condition.
kPartialBlockageHighImpact	0x01	The sensor has detected a blockage condition which has a significant impact on sensor performance (e.g. range).
kPartialBlockageMediumImpact	0x02	The sensor has detected a blockage condition which already has impact on sensor performance (e.g. range).
kPartialBlockageLowImpact	0x03	The sensor detects that a blockage condition is present or is increasing, but the degree of blockage has not yet had a significant impact on sensor performance and functionality.
kDefect	0x04	The full specified range is blocked, due to e.g. a pixel defect. This segment may overlap with other segments.
kNone	0x05	Normal mode.
kUnknown	0x06	The blockage Status is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.162 FOVReductionInfo

[SWS_ADI_00507]{DRAFT} ⌈

Name	FOVReductionInfo
Kind	STRUCTURE
Subelements	NoValidReductionReasons `uint8_t` FOVReductionTypeList `FOVReductionTypeInfoVector`
Derived from	-
Description	Represents the FOV reduction related information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.163 FieldOfViewReductionType

[SWS_ADI_00508]{DRAFT} ⌈

Name	FieldOfViewReductionType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Defines the overall blockage of the FOV segment.	
Range / Symbol	***Limit***	***Description***
kSnow	0x00	Range reduction due to snow.
kRain	0x01	Range reduction due to rain.
kClutter	0x02	Range reduction due to clutter.
kFlyingLeaves	0x03	Range reduction due to flying leaves.
kNightAndLights	0x04	Range reduction due to night and lights.
kShades	0x05	Range reduction due to shades.
kContrastIssues	0x06	Range reduction due to contrast issues.
kJamming	0x07	Range reduction, e.g. electromagnetic compatibility.
kDeviceInterference	0x08	Range reduction, e.g. electromagnetic compatibility.
kSand	0x09	Range reduction due to sand.
kWetRoads	0x0A	Range reduction due to wet roads.
kGhosts	0x0B	Range reduction due to ghosts.
kSnowOnSensorSurface	0x0C	Near range blockage due to snow on the sensor surface.
kWaterOnSensorSurface	0x0D	Near range blockage due to water on the sensor surface.
kSoilOnSensorSurface	0x0E	Near range blockage due to soil on the sensor surface.
kScratchesOnSensorSurface	0x0F	Near range blockage due to scratches on the sensor surface.
kUnknown	0x10	FOV Reduction type is unknown.

⌋*(RS_ADI_00001, RS_ADI_00002, RS_ADI_00003, RS_ADI_00004, RS_ADI_00012, RS_ADI_00013, RS_ADI_00014)*

9.1.164 FOVReductionTypeInfoVector

[SWS_ADI_00509]{DRAFT} ⌈

Name	FOVReductionTypeInfoVector
Kind	VECTOR
Subelements	`FOVReductionTypeInfo`
Derived from	-
Description	Represents a list of FOV reduction reason type information.

⌋*(RS_ADI_00001, RS_ADI_00002, RS_ADI_00003, RS_ADI_00004, RS_ADI_00012, RS_ADI_00013, RS_ADI_00014)*

9.1.165 FOVReductionTypeInfo

[SWS_ADI_00510]{DRAFT} ⌈

Name	FOVReductionTypeInfo
Kind	STRUCTURE
Subelements	FOVReductionType `FieldOfViewReductionType` FieldOfViewReductionTypePro `ProPercentage`
Derived from	-
Description	Represents the FOV Reduction type information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.166 RecognizedObjectType

[SWS_ADI_00511]{DRAFT} ⌈

Name	RecognizedObjectType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Defines the overall blockage of the FOV segment.	
Range / Symbol	***Limit***	***Description***
kCar	0x00	Recognized entity is a car.
kTruck	0x01	Recognized entity is a truck.
kMotorBike	0x02	Recognized entity is a motor bike.
kBicycle	0x03	Recognized entity is a bicycle.
kPedestrian	0x04	Recognized entity is a pedestrian.
kMovingObject	0x05	Recognized entity is an unknown moving object.
kRoadBoundary	0x06	Recognized entity is a road boundary.
kRoadMarking	0x07	Recognized entity is a road marking.
kStaticObject	0x08	Recognized entity is a static object.
kTrafficSign	0x09	Recognized entity is a traffic sign.
kTrafficLight	0x0A	Recognized entity is a traffic light.
kUnknown	0x0B	The Recognized Object Type is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.167 DetectionRange

[SWS_ADI_00512]{DRAFT} ⌈

Name	DetectionRange
Kind	STRUCTURE
Subelements	Min `float` Max `float`
Derived from	-
Description	Sensor detection range for one object type with Minimum classification rate this object type and Maximum false positive rate for this object type. (m, m)

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.168 ObjectDetectionRateInfo

[SWS_ADI_00513]{DRAFT} ⌈

Name	ObjectDetectionRateInfo
Kind	STRUCTURE
Subelements	RObjectType `RecognizedObjectType` ODetectionRange `DetectionRange` MinimunClassRate `ProPercentage` (optional) MaxFalsePositiveRate `ProPercentage` (optional) PositivePredictiveValue `ProPercentage` (optional)
Derived from	-
Description	Represents the Object Detection Rate information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.169 ObjectDetectionRateInfoVector

[SWS_ADI_00514]{DRAFT} ⌈

Name	ObjectDetectionRateInfoVector
Kind	VECTOR
Subelements	`ObjectDetectionRateInfo`
Derived from	-
Description	Represents a list of Object Detection Rate information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.170 ObjectDetectionRates

[SWS_ADI_00515]{DRAFT} ⌈

Name	ObjectDetectionRates
Kind	STRUCTURE
Subelements	NoValidRecognizedObjectType `uint8_t` ObjectDetectionRatesList `ObjectDetectionRateInfoVector`
Derived from	-
Description	Represents the Object Detection Rate related information.

⌋*(RS_ADI_00001, RS_ADI_00002, RS_ADI_00003, RS_ADI_00004, RS_ADI_00012, RS_ADI_00013, RS_ADI_00014)*

9.1.171 ReferTargetRateInfo

[SWS_ADI_00517]{DRAFT} ⌈

Name	ReferTargetRateInfo
Kind	STRUCTURE
Subelements	RCSRefTarget `float` (optional) ReflectivityRefTarget `float` (optional) DeltaEDiff `float` (optional) RTSpeedRange `RelativeSpeedRange` (optional) RTDetectionRange `DetectionRange` SNRatio `float`
Derived from	-
Description	Represents the Reference Target Rate information.

⌋*(RS_ADI_00001, RS_ADI_00002, RS_ADI_00003, RS_ADI_00004, RS_ADI_00012, RS_ADI_00013, RS_ADI_00014)*

9.1.172 ReferTargetRateInfoVector

[SWS_ADI_00518]{DRAFT} ⌈

Name	ReferTargetRateInfoVector
Kind	VECTOR
Subelements	`ReferTargetRateInfo`
Derived from	-
Description	Represents a list of Reference Target Rate information.

⌋*(RS_ADI_00001, RS_ADI_00002, RS_ADI_00003, RS_ADI_00004, RS_ADI_00012, RS_ADI_00013, RS_ADI_00014)*

9.1.173 ReferenceTargetDetectionRates

[SWS_ADI_00519]{DRAFT} ⌈

Name	ReferenceTargetDetectionRates
Kind	STRUCTURE
Subelements	NoValidRefDetectionClass `uint8_t` ReferTargetRateList `ReferTargetRateInfoVector`
Derived from	-
Description	Represents the Reference Target Rate related information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.174 RelativeSpeedRange

[SWS_ADI_00520]{DRAFT} ⌈

Name	RelativeSpeedRange
Kind	STRUCTURE
Subelements	Min `float` Max `float`
Derived from	-
Description	Describes the relative radial speed range in the sensor coordinate system.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.175 PefSegmentInfo

[SWS_ADI_00521]{DRAFT} ⌈

Name	PefSegmentInfo
Kind	STRUCTURE
Subelements	PefStatus `SensorPerformanceStatus` PefFOVReductionInfo `FOVReductionInfo` RealWorldODRates `ObjectDetectionRates` (optional) RefDetectionRates `ReferenceTargetDetectionRates` (optional)
Derived from	-
Description	Represents the Performance sgement information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.176 SensorOperationState

[SWS_ADI_00522]{DRAFT} ⌈

Name	SensorOperationState	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Status information of the sensor.	
Range / Symbol	***Limit***	***Description***
kNormal	0x00	All sensor components up and running: e.g. all initialization given and communication fully established.
kLimited	0x01	Only limited processing power available.
kOff	0x02	Status during shut-off.
kError	0x03	Internal error detected: e.g. memory check.
kInitializing	0x04	During sensor initialization is carried out.
kNotDefined	0x05	Sensor operation state is not defined.
kUnknown	0x06	The sensor operation state is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.177 SensorOperationMode

[SWS_ADI_00523]{DRAFT} ⌈

Name	SensorOperationMode	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Status information of the sensor.	
Range / Symbol	***Limit***	***Description***
kSensorMeasuringActive	0x00	Sensor is active and performs measurements.
kSensorMeasuringDisabled	0x01	Sensor is disabled and performs no measurement at the moment.
kSensorMeasuringTestmode	0x02	Sensor is in active measurement mode, however in test mode.
kUnknown	0x03	The Sensor operation mode is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.178 SensorDefectDetected

[SWS_ADI_00524]{DRAFT} ⌈

Name	SensorDefectDetected	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Signal for a sensor defect is detected.	
Range / Symbol	**Limit**	**Description**
kSensorFullyFunctional	0x00	Sensor has no defects detected.
kNotFullyFunctionalDueTo Defect	0x01	Sensor has detected defects. Sensor can measure with limited performance.
kOutOfOrder	0x02	Sensor has detected defects and cannot perform measurements anymore.
kUnknown	0x03	The sensor defect detected is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.179 SensorDefectReason

[SWS_ADI_00525]{DRAFT} ⌈

Name	SensorDefectReason	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Signal for detailed information why the signal Sensor defect detected is notifying a sensor problem.	
Range / Symbol	**Limit**	**Description**
kNoDefectDetected	0x00	No defects detected by the sensor.
kInternalMemoryError	0x01	Sensor has detected an internal memory error.
kElectronicDefect	0x02	
kThermalDefect	0x03	Sensor has detected a thermal problem error.
kSurgeDefect	0x04	Sensor has detected a surge defect.
kCalibrationError	0x05	Sensor has detected a calibration error.
kImplausibleSensor Parametrisation	0x06	Sensor has detected an implausible parametrisation.
kMechanicalDefect	0x07	Sensor has detected a mechanical defect.
kSoftwareDefect	0x08	Sensor has detected a software defect.
kComputingPowerNotSufficient	0x09	Sensor has detected a to low power supply.
kOutOfTimeSyncronisation	0x10	Sensor has detected an out of time synchronisation.
kSensorExternalDisturbed	0x11	Sensor has detected an external disturbance.
kUnknown	0x12	The sensor defect reason is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.180 SensorDiagnosticMode

[SWS_ADI_00526]{DRAFT} ⌈

Name	SensorDiagnosticMode	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Signal for the current Sensor diagnostic mode.	
Range / Symbol	**Limit**	**Description**
kIdle	0x00	Sensor not in diagnostic mode. Sensor in normal operation.
kSensorInEndOfLineMode	0x01	End of line alignment in operation.
kSensorInSelfDiagnoseMode	0x02	Sensor in self-diagnosis mode, normal sensor operation is not provided.
kUnknown	0x03	The Sensor diagnostic mode is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.181 StatusSupplyVoltage

[SWS_ADI_00527]{DRAFT} ⌈

Name	StatusSupplyVoltage	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Signal for the current Status supply voltage status.	
Range / Symbol	**Limit**	**Description**
kWithinLimits	0x00	Supply voltage is optimal.
kLow	0x01	Supply voltage is out of valid range. Supply voltage is too low.
kPreLow	0x02	Supply voltage still in the valid range, but close to the limit and expected to leave the valid range soon.
kPreHigh	0x03	Supply voltage still in the valid range, but close to the limit and expected to leave the valid range soon.
kHigh	0x04	Supply voltage is out of valid range. Supply voltage is too high.
kUnknown	0x05	The sensor supply voltage status is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.182 SensorTemperatureStatus

[SWS_ADI_00528]{DRAFT} ⌈

Name	SensorTemperatureStatus	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Signal for the current Sensor temperature status status.	
Range / Symbol	**Limit**	**Description**
kUnderTemperature	0x00	No measurement updates available.
kPreUnderTemperature	0x01	Close before under temperature.
kTemperatureInLimits	0x02	Normal mode.
kPreOverTemperature	0x03	Close before over temperature.
kOverTemperature	0x04	No measurement updates available.
kUnknown	0x05	The sensor temperature status is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.183 PefSegmentInfoVector

[SWS_ADI_00529]{DRAFT} ⌈

Name	PefSegmentInfoVector
Kind	VECTOR
Subelements	`PefSegmentInfo`
Derived from	-
Description	Represents a list of performance segment information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.184 SensorInputSignalType

[SWS_ADI_00530]{DRAFT} ⌈

Name	SensorInputSignalType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Classification of the Sensor input signal - type , which defines a group of sensor input signals received by the sensor.	
Range / Symbol	**Limit**	**Description**
kDynamicMotionControl	0x00	Dynamic motion control sensor input signals.
kVehicleDynamic	0x01	Vehicle dynamic sensor input signals.

▽

△

kUnknown	0x02	Sensor input signal type is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.185 SensorInputSignalStatus

[SWS_ADI_00531]{DRAFT} ⌈

Name	SensorInputSignalStatus	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Enumeration if valid input signals for Sensor input signal - type are received by the sensor.	
Range / Symbol	***Limit***	***Description***
kValid	0x00	Normal mode.
kImplausible	0x01	Signal in context of sensor signals is not plausible compared with other signals or internal calculations.
kMissing	0x02	Signal was never received.
kOufOfRange	0x03	Signal violated the signal range.
kTimeout	0x04	Signal was received, however not in time period as expected.
kUnknown	0x05	The sensor input signal status is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.186 SensorExternalDisturbed

[SWS_ADI_00532]{DRAFT} ⌈

Name	SensorExternalDisturbed	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Signal about the disturbance of the sensor by an external source.	
Range / Symbol	***Limit***	***Description***
kFullDisturbance	0x00	The sensor is completely disturbed, no more feature and functionality working due to external disturbance.
kDisturbanceHighImpact	0x01	The sensor has detected an external disturbance which has a significant impact on sensor performance.
kDisturbanceMediumImpact	0x02	The sensor has detected an external disturbance which already has impact on sensor performance.

▽

△

kDisturbanceLowImpact	0x03	The sensor detects that an external disturbance is present or is increasing, but the degree of disturbance has not yet had a significant impact on sensor performance and functionality.
kNone	0x04	Normal mode.
kUnknown	0x05	The sensor external disturbed status is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.187 SensorTransmitPowerReduced

[SWS_ADI_00533]{DRAFT} ⌈

Name	SensorTransmitPowerReduced	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Enumeration if the sensor works with full output power.	
Range / Symbol	***Limit***	***Description***
kNormalOperation	0x00	Transmit output power normal.
kOutputPowerLimited	0x01	Transmit output power reduced.
kUnknown	0x02	Sensor Transmit Power status is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.188 StatusSensorHeating

[SWS_ADI_00534]{DRAFT} ⌈

Name	StatusSensorHeating	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Status of the sensor heating.	
Range / Symbol	***Limit***	***Description***
kHeatingOff	0x00	No heating active.
kHeatingLevel	0x01	Sensor heating active.
kHeatingError	0x02	Sensor heating is defect.
kUnknown	0x03	The sensor heating status is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.189 StatusSensorCleaning

[SWS_ADI_00535]{DRAFT} ⌈

Name	StatusSensorCleaning	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Status of the sensor cleaning.	
Range / Symbol	***Limit***	***Description***
kCleaningIdle	0x00	Sensor cleaning is not active.
kCleaningActive	0x01	Sensor cleaning is active
kCleaningNeeded	0x02	Sensor cleaning should be performed.
kUnknown	0x03	The sensor cleaning status is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.190 StatusTimSync

[SWS_ADI_00536]{DRAFT} ⌈

Name	StatusTimSync	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Status of the sensor time synchronisation.	
Range / Symbol	***Limit***	***Description***
kWithinLimits	0x00	Time synchronization inside limits.
kOutOfLimits	0x01	Time synchronization time accuracy limits violated.
kTimeout	0x02	Time synchronization timeout elapsed (no valid time synchronization cycle within timeout interval).
kOffset	0x03	Time offset value. Requires: Sensor time sync offset value
kUnknown	0x04	The time synchronization status is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.191 PerformanceSegmentsInfo

[SWS_ADI_00537]{DRAFT} ⌈

Name	PerformanceSegmentsInfo
Kind	STRUCTURE
Subelements	NoValidFoVSegments `uint8_t` PefSegmentsList `PefSegmentInfoVector`
Derived from	-
Description	Represents the performance segments related information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.192 SensorPerformance

[SWS_ADI_00538]{DRAFT} ⌈

Name	SensorPerformance
Kind	STRUCTURE
Subelements	SPHeader `GeneralHeader` SPScopeIntf `ScopeOfInterfaces` (optional) SensorMountingPos `SensorMountingInfo` VPInfo `VanishingPointInfo` (optional) PerformanceSegments `PerformanceSegmentsInfo`
Derived from	-
Description	Represents the sensor performance interface information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.193 SensorCalibratableComponent

[SWS_ADI_00539]{DRAFT} ⌈

Name	SensorCalibratableComponent	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Enumeration for the sensor component which may be calibrated.	
Range / Symbol	***Limit***	***Description***
kIntrinsic	0x00	Calibration status for the intrinsic parameters of the sensor.
kExtrinsic	0x01	Calibration status for the extrinsic parameters of the sensor.
kOnline	0x02	Calibration status for the online parameters of the sensor.
kUnknown	0x03	The sensor calibratable component is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.194 SensorCalibrationStatus

[SWS_ADI_00540]{DRAFT} ⌈

Name	SensorCalibrationStatus	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Enumeration for the current calibration status of the Sensor calibratable component.	
Range / Symbol	***Limit***	***Description***
kCalibrated	0x00	Sensor calibration was successful and within nominal tolerance range.
kNotCalibrated	0x01	Calibration not done or calibration failed.
kDegraded	0x02	Sensor calibrated, however performance degraded due to limited correction accuracy.
kInitialCalibrationNotPerformed	0x03	Sensor initial calibration not performed yet.
kInitialCalibrationFailed	0x04	Sensor initial calibration process failed.
kRecalibrationNeededIntrinsic	0x05	Recalibration of sensor's intrinsic parameters required.
kRecalibrationNeededExtrinsic	0x06	Recalibration of sensor's extrinsic parameters required.
kRecalibrationNeededFull	0x07	Recalibration of the complete sensor's parameters required.
kUnknown	0x08	The sensor Calibration status is unknown.

⌋*(RS_ADI_00001, RS_ADI_00002, RS_ADI_00003, RS_ADI_00004, RS_ADI_00012, RS_ADI_00013, RS_ADI_00014)*

9.1.195 InputSignalInfo

[SWS_ADI_00541]{DRAFT} ⌈

Name	InputSignalInfo
Kind	STRUCTURE
Subelements	InSignalType `SensorInputSignalType` InSingalStatus `SensorInputSignalStatus`
Derived from	-
Description	Represents the input signal information.

⌋*(RS_ADI_00001, RS_ADI_00002, RS_ADI_00003, RS_ADI_00004, RS_ADI_00012, RS_ADI_00013, RS_ADI_00014)*

9.1.196 InputSignalInfoVector

[SWS_ADI_00542]{DRAFT} ⌈

Name	InputSignalInfoVector
Kind	VECTOR
Subelements	`InputSignalInfo`
Derived from	-
Description	Represents a list of input singal information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.197 SensorHealthStatus

[SWS_ADI_00543]{DRAFT} ⌈

Name	SensorHealthStatus
Kind	STRUCTURE
Subelements	OperationState `SensorOperationState` OperationMode `SensorOperationMode` DefectDetected `SensorDefectDetected` DefectReason `SensorDefectReason` DiagMode `SensorDiagnosticMode` SupplyVoltage `StatusSupplyVoltage` TempStatus `SensorTemperatureStatus` NoValidSensorInput `uint8_t` ValidSenorInputSignalList `InputSignalInfoVector` ExternalDisturbed `SensorExternalDisturbed` (optional) TPReduced `SensorTransmitPowerReduced` (optional) HeatingStatus `StatusSensorHeating` (optional) CleaningStatus `StatusSensorCleaning` (optional) TSYStatus `StatusTimSync` (optional) SensorTimeSynccOffset `float` (optional)
Derived from	-
Description	Represents the sensor health related information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.198 CaliComponentInfo

[SWS_ADI_00544]{DRAFT} ⌈

Name	CaliComponentInfo
Kind	STRUCTURE
Subelements	CaliComponent `SensorCalibratableComponent` CaliStatus `SensorCalibrationStatus`
Derived from	-
Description	Represents the caliberation component information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.199 CaliComponentInfoVector

[SWS_ADI_00545]{DRAFT} ⌈

Name	CaliComponentInfoVector
Kind	VECTOR
Subelements	`CaliComponentInfo`
Derived from	-
Description	Represents a list of caliberation component information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.200 SensorCaliInfo

[SWS_ADI_00546]{DRAFT} ⌈

Name	SensorCaliInfo
Kind	STRUCTURE
Subelements	NoValidSensorCalibratableComponents `uint8_t` SensorCaliCompList `CaliComponentInfoVector` CoorectionOriginPos `Point3D` (optional) CoorectionOriginPosError `Point3DError` (optional) CoorectionPosLimitMin `Point3D` (optional) CoorectionPosLimitMax `Point3D` (optional) CoorectionOrientation `Orientation3D` (optional) CoorectionOrientationError `Orientation3DError` (optional) CoorectionAngleLimitMin `Orientation3D` (optional) CoorectionAngleLimitMax `Orientation3D` (optional)
Derived from	-

▽

△

Description	Represents the sensor caliberation related information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.201 SensorHealthInfo

[SWS_ADI_00547]{DRAFT} ⌈

Name	SensorHealthInfo
Kind	STRUCTURE
Subelements	SHHeader `GeneralHeader` SensorMountingPos `SensorMountingInfo` SHStatus `SensorHealthStatus` (optional) SensorCali `SensorCaliInfo` (optional)
Derived from	-
Description	Represents the sensor Health interface information.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.202 ShapeType

[SWS_ADI_00601]{DRAFT} ⌈

Name	ShapeType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Classification of the general landmark.	
Range / Symbol	***Limit***	***Description***
kPoint	0x00	Shape is a point.
kBox	0x01	Shape is a box (2 or 3 points).
kEllipse	0x02	Shape is an ellipse (2 or 3 points).
kPolygon	0x03	Shape is a polygon (3 or more points).
kPolyline	0x04	Shape is a polyline (2 or more points).
kPointCloud	0x05	Shape is a point cloud (2 or more points).
kUnknown	0x06	Shape type is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.203 ShapeClassType

[SWS_ADI_00602]{DRAFT} ⌈

Name	ShapeClassType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	The classification type for the shape.	
Range / Symbol	***Limit***	***Description***
kBackground	0x00	Shape is classified as background entity.
kForeground	0x01	Shape is classified as foreground entity.
kFlat	0x02	Shape is classified as flat entity.
kUpright	0x03	Shape is classified as upright entity.
kGround	0x04	Shape is classified as ground entity.
kBuilding	0x05	Shape is classified as building entity.
kVegetation	0x06	Shape is classified as vegetation entity.
kRoad	0x07	Shape is classified as road entity.
kNonRoad	0x08	Shape is classified as non-road entity.
kSidewalk	0x09	Shape is classified as sidewalk entity.
kPedestrian	0x0A	Shape is classified as pedestrian entity.
kVehicle	0x0B	Shape is classified as vehicle entity.
kTrafficSign	0x0C	Shape is classified as traffic sign entity.
kPedestrianFront	0x0D	Shape is classified as pedestrian front-view entity.
kPedestrianSide	0x0E	Shape is classified as pedestrian side-view entity.
kPedestrianRear	0x0F	Shape is classified as pedestrian rear-view entity.
kUnknown	0x10	Shape Class type is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.204 UltrasonicFeatureType

[SWS_ADI_00603]{DRAFT} ⌈

Name	UltrasonicFeatureType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Ultrasonic feature type contains information about the current measurement of this feature.	
Range / Symbol	***Limit***	***Description***
kPoint	0x00	Defined by one point.
kLineSegment	0x01	Defined by two or more points.
kUnknown	0x02	The Ultrasonic feature type is unknown.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.205 Point2D

[SWS_ADI_00604]{DRAFT} ⌈

Name	Point2D
Kind	STRUCTURE
Subelements	x `float` y `float`
Derived from	-
Description	Position with respect to the vehicle origin.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.206 Point2DError

[SWS_ADI_00605]{DRAFT} ⌈

Name	Point2DError
Kind	STRUCTURE
Subelements	x `float` y `float`
Derived from	-
Description	Error values for feature's 2D position {x, y}.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.207 TrilaterationStatus

[SWS_ADI_00606]{DRAFT} ⌈

Name	TrilaterationStatus	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Information if feature is trilaterated with multiple signal ways or is not trilaterated.	
Range / Symbol	***Limit***	***Description***
kNormal	0x00	The 2D position {x, y} measurement is based on at least three points.
kNotTrilaterated	0x01	The 2D position {x, y} measurement is based on less than three points.
kUnknown	0x02	The trilateration status is unknown.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.208 MeasurementStatusFeature

[SWS_ADI_00607]{DRAFT} ⌈

Name	MeasurementStatusFeature	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Information about the measurement status of the feature.	
Range / Symbol	***Limit***	***Description***
kBackground	0x00	Shape is classified as background entity.
kForeground	0x01	Shape is classified as foreground entity.
kFlat	0x02	Shape is classified as flat entity.
kUpright	0x03	Shape is classified as upright entity.
kGround	0x04	Shape is classified as ground entity.
kBuilding	0x05	Shape is classified as building entity.
kVegetation	0x06	Shape is classified as vegetation entity.
kRoad	0x07	Shape is classified as road entity.
kNonRoad	0x08	Shape is classified as non-road entity.
kSidewalk	0x09	Shape is classified as sidewalk entity.
kPedestrian	0x0A	Shape is classified as pedestrian entity.
kVehicle	0x0B	Shape is classified as vehicle entity.
kTrafficSign	0x0C	Shape is classified as traffic sign entity.
kPedestrianFront	0x0D	Shape is classified as pedestrian front-view entity.
kPedestrianSide	0x0E	Shape is classified as pedestrian side-view entity.
kPedestrianRear	0x0F	Shape is classified as pedestrian rear-view entity.
kUnknown	0x10	Shape Class type is unknown.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.209 ShapeClassTypeInfo

[SWS_ADI_00608]{DRAFT} ⌈

Name	ShapeClassTypeInfo
Kind	STRUCTURE
Subelements	CShapeClassType `ShapeClassType` ShapeClassProValue `ProPercentage`
Derived from	-
Description	Represents the shape class type information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.210 ShapeClassTypeInfoVector

[SWS_ADI_00609]{DRAFT} ⌈

Name	ShapeClassTypeInfoVector
Kind	VECTOR
Subelements	`ShapeClassTypeInfo`
Derived from	-
Description	Represents a list of shape class type information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.211 ShapeInfo

[SWS_ADI_00610]{DRAFT} ⌈

Name	ShapeInfo
Kind	STRUCTURE
Subelements	CFShapeType `ShapeType` NoValidShapeClass `uint8_t` CFShapeClassTypeList `ShapeClassTypeInfoVector` ColourValueList `ColourValueVector` ColourPro `ProPercentage` (optional)
Derived from	-
Description	Represents the shape related information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.212 ShapePointInfo

[SWS_ADI_00611]{DRAFT} ⌈

Name	ShapePointInfo
Kind	STRUCTURE
Subelements	PointExistPro `ProPercentage` Position3D `Point3D` Position3DError `Point3DError`
Derived from	-
Description	Represents the Shape points information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.213 ShapePointInfoVector

[SWS_ADI_00612]{DRAFT} ⌈

Name	ShapePointInfoVector
Kind	VECTOR
Subelements	ShapePointInfo
Derived from	-
Description	Represents a list of shape point information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.214 ShapePoints

[SWS_ADI_00613]{DRAFT} ⌈

Name	ShapePoints
Kind	STRUCTURE
Subelements	NoValidShapePoints uint16_t ShapePointsList ShapePointInfoVector
Derived from	-
Description	Represents the Shape points related information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.215 ShapeReferPointInfo

[SWS_ADI_00614]{DRAFT} ⌈

Name	ShapeReferPointInfo
Kind	STRUCTURE
Subelements	PointExistPro ProPercentage Position3D Point3D Position3DError Point3DError OrientationNormal Point3D (optional) OrientationNormalError Point3DError (optional) TranslationRate Point3D (optional) TranslationRateError Point3DError (optional) RotationRate Orientation3D (optional) ▽

▽

△

	△ RotationRateError `Orientation3DError` (optional) ScaleChange `float` (optional) ScaleChangeError `float` (optional)
Derived from	-
Description	Represents the Shape reference points information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.216 ShapeReferPointInfoVector

[SWS_ADI_00615]{DRAFT} ⌈

Name	ShapeReferPointInfoVector
Kind	VECTOR
Subelements	`ShapeReferPointInfo`
Derived from	-
Description	Represents a list of shape Reference point information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.217 ShapeReferPoints

[SWS_ADI_00616]{DRAFT} ⌈

Name	ShapeReferPoints
Kind	STRUCTURE
Subelements	NoValidShapeRPs `uint8_t` ShapeReferPointsList `ShapeReferPointInfoVector`
Derived from	-
Description	Represents the Shape Reference points related information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.218 FeatureStatus

[SWS_ADI_00617]{DRAFT} ⌈

Name	FeatureStatus
Kind	STRUCTURE
Subelements	FeatureExistPro `ProPercentage` FeatureID `uint_16` (optional) ObjectID `uint_16` (optional) TimestampDiff `uint_64` Observation `uint_32` (optional)
Derived from	-
Description	Represents the dynamics of the features.

⌋(*RS_ADI_00001*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.219 CameraFeature

[SWS_ADI_00618]{DRAFT} ⌈

Name	CameraFeature
Kind	STRUCTURE
Subelements	CFStatus `FeatureStatus` CFShapeInfo `ShapeInfo` CFShapePoints `ShapePoints` CFShapeReferPoints `ShapeReferPoints` (optional)
Derived from	-
Description	Represents the Camera feature information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.220 CameraGeneralHeader

[SWS_ADI_00619]{DRAFT} ⌈

Name	CameraGeneralHeader
Kind	STRUCTURE
Subelements	Basic `HeaderBasic` VehicleCoordinateST `VehicleCoordinateSystemType` ColourModelType `ColourModelType`
Derived from	-
Description	Represents the Genreal header infomation for Camera.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.221 CameraFeatureVector

[SWS_ADI_00620]{DRAFT} ⌈

Name	CameraFeatureVector
Kind	VECTOR
Subelements	`CameraFeature`
Derived from	-
Description	Represents a list of camerqa feature information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.222 CameraFeatures

[SWS_ADI_00621]{DRAFT} ⌈

Name	CameraFeatures
Kind	STRUCTURE
Subelements	CFHeader `CameraGeneralHeader` RecognizedFeaturesCap `uint_32` (optional) RecognizedFeaturesStatus `RecognizedStatus` (optional) NoValidFeatures `uint_32` CFeatueList `CameraFeatureVector`
Derived from	-
Description	Represents the camera feature interface information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.223 UltrasonicSegmentInfo

[SWS_ADI_00622]{DRAFT} ⌈

Name	UltrasonicSegmentInfo
Kind	STRUCTURE
Subelements	UltrasonicFType `UltrasonicFeatureType` UltrasonicProValue `ProPercentage`
Derived from	-
Description	Represents the Ultrasonic segmengt type information.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.224 ValidPointInfo

[SWS_ADI_00623]{DRAFT} ⌈

Name	ValidPointInfo
Kind	STRUCTURE
Subelements	Position2D `Point2D` Position2DError `Point2DError` OrientationUltrasonic `float` (optional) OrientationUltrasonicError `float` (optional) HeightUltrasonic `float` (optional) HeightUltrasonicError `float` (optional) VelocityUltrasonic `Point2D` (optional) VelocityUltrasonicError `Point2DError` (optional) TriStatus `TrilaterationStatus` FeatureMStatus `MeasurementStatusFeature` (optional)
Derived from	-
Description	Represents the valid points information.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.225 ValidPointInfoVector

[SWS_ADI_00624]{DRAFT} ⌈

Name	ValidPointInfoVector
Kind	VECTOR
Subelements	`ValidPointInfo`
Derived from	-
Description	Represents a list of shape points information.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.226 SegmentPointsInfo

[SWS_ADI_00625]{DRAFT} ⌈

Name	SegmentPointsInfo
Kind	STRUCTURE
Subelements	NoValidPointsUltrasonic `uint16_t` ValidPointsList `ValidPointInfoVector`

▽

△

Derived from	-
Description	Represents the segment points information.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.227 UltrasonicFeature

[SWS_ADI_00626]{DRAFT} ⌈

Name	UltrasonicFeature
Kind	STRUCTURE
Subelements	UltrasonicFStatus `FeatureStatus` SegInfo `UltrasonicSegmentInfo` SegmentPointsList `SegmentPointsInfo`
Derived from	-
Description	Represents the Ultrasonic feature information.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.228 UltrasonicFeatureVector

[SWS_ADI_00627]{DRAFT} ⌈

Name	UltrasonicFeatureVector
Kind	VECTOR
Subelements	`UltrasonicFeature`
Derived from	-
Description	Represents a list of Ultrasonic feature information.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.229 UltrasonicFeatures

[SWS_ADI_00628]{DRAFT} ⌈

Name	UltrasonicFeatures
Kind	STRUCTURE

▽

△

Subelements	UltrasonicFHeader `GeneralHeader` RecognizedFeaturesCap `uint_32` (optional) RecognizedFeaturesStatus `RecognizedStatus` (optional) NoValidFeatures `uint_32` UltrasonicFeatueList `UltrasonicFeatureVector`
Derived from	-
Description	Represents the Ultrasonic feature interface information.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.230 Position3DSpheric

[SWS_ADI_00701]{DRAFT} ⌈

Name	Position3DSpheric
Kind	STRUCTURE
Subelements	elevation `float` azimuth `float` distance `float` (optional)
Derived from	-
Description	Represents a 3 dimension vector, the unit will be vary according to the refering data type.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.231 Position3DSphericError

[SWS_ADI_00702]{DRAFT} ⌈

Name	Position3DSphericError
Kind	STRUCTURE
Subelements	elevation `float` azimuth `float` distance `float` (optional)
Derived from	-
Description	Error values of the {Azimuth, Elevation, Distance} to the Position {Azimuth, Elevation, Distance}.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.232 DetectionClassType

[SWS_ADI_00703]{DRAFT} ⌈

Name	DetectionClassType	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	The classification type for the shape.	
Range / Symbol	***Limit***	***Description***
kNoClassification	0x00	Detection entity is not classified.
kNoise	0x01	Detection entity is noise.
kObstacle	0x02	Detection entity is an obstacle for vehicle.
kUnderdriveable	0x03	Detection entity is underdrivable for vehicle.
kOverdrivable	0x04	Detection entity is overdrivable for vehicle.
kNearest	0x05	Detection entity is the nearest detection of a measurement.
kStrongest	0x06	Detection entity has the strongest signal of a measurement.
kUnknown	0x07	The detection type is unknown.

⌋(*RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_-00014*)

9.1.233 ShapeClassVariant

[SWS_ADI_00704]{DRAFT} ⌈

Name	ShapeClassVariant	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	Description of the segment's possible content for Shape classification type (Shape Classification_False means confidence less tahn 50 % otherwise "ShapeClassification_ True").	
Range / Symbol	***Limit***	***Description***
kTrue	0x00	Shape is shape classification type.
kFalse	0x01	Shape is not shape classification type.
kUnknown	0x02	ShapeClassVariant is unknown.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.234 EllipseAxesError

[SWS_ADI_00705]{DRAFT} ⌈

Name	EllipseAxesError
Kind	STRUCTURE
Subelements	SemiMinor `float` SemiMajor `float` (optional)
Derived from	-
Description	EDistributed sender/receiver sensor (virtual sensor) the distance is the ellipse's semi-major axis of the detection. Single sensor (real sensor) the semi-major axis is the radius of the detection.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.235 DetectionStatus

[SWS_ADI_00706]{DRAFT} ⌈

Name	DetectionStatus
Kind	STRUCTURE
Subelements	DetectionExistPro `ProPercentage` FeatureID `uint_16` (optional) ObjectID `uint_16` (optional) TimestampDiff `uint_64`
Derived from	-
Description	Represents the dynamics of the detections.

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.236 DetectionClassTypeInfo

[SWS_ADI_00707]{DRAFT} ⌈

Name	DetectionClassTypeInfo
Kind	STRUCTURE
Subelements	DClassType `DetectionClassType` (optional) DetectionClassPro `ProPercentage` (optional)
Derived from	-
Description	Represents the detection class type information.

⌋(*RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.237 DetectionClassTypeInfoVector

[SWS_ADI_00708]{DRAFT} ⌈

Name	DetectionClassTypeInfoVector
Kind	VECTOR
Subelements	`DetectionClassTypeInfo`
Derived from	-
Description	Represents a list of detection class type information.

⌋(*RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.238 RadarDetectionInfo

[SWS_ADI_00709]{DRAFT} ⌈

Name	RadarDetectionInfo
Kind	STRUCTURE
Subelements	PositionSpheric `Position3DSpheric` PositionSphericError `Position3DSphericError` RadialRelVelocity `float` RadialRelVelocityError `float` (optional) RCS `float` RCSError `float` (optional) SNR `float` SNRError `float` (optional) MultiTargetPro `ProPercentage` (optional) AmbiguityID `uint_16` (optional) DetectionAmbiguityPro `ProPercentage` (optional) FreeSpacePro `ProPercentage` (optional) NoValidDetectionClass `uint_32` (optional) RDetectionClassTypeList `DetectionClassTypeInfoVector` (optional)
Derived from	-
Description	Represents the radar detection information.

⌋(*RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.239 RadarDetection

[SWS_ADI_00710]{DRAFT} ⌈

Name	RadarDetection
Kind	STRUCTURE
Subelements	RDetectionStatus `DetectionStatus` RDetetionInfo `RadarDetectionInfo`

▽

△

Derived from	-
Description	Represents the Radar detection information.

⌋(*RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.240 RadarDetectionVector

[SWS_ADI_00711]{DRAFT} ⌈

Name	RadarDetectionVector
Kind	VECTOR
Subelements	`RadarDetection`
Derived from	-
Description	Represents a list of radar detection information.

⌋(*RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.241 RadarDetectionsInfo

[SWS_ADI_00712]{DRAFT} ⌈

Name	RadarDetectionsInfo
Kind	STRUCTURE
Subelements	RecognizedDetectionsCap `uint_32` (optional) RecognizedDetectionsStatus `RecognizedStatus` (optional) NoValidDetections `uint_32` RadarDetectionList `RadarDetectionVector`
Derived from	-
Description	Represents the radar detections information.

⌋(*RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.242 AmbiguityDomainInfo

[SWS_ADI_00713]{DRAFT} ⌈

Name	AmbiguityDomainInfo
Kind	STRUCTURE
Subelements	VelocityAmbiguity `RadialVelocityAmbiguityDomain` (optional) RangeAmbiguity `RangeAmbiguityDomain` (optional) AzimuthAmbiguity `AngleAzimuthAmbiguityDomain` (optional) ElevationAmbiguity `AngleElevationAmbiguityDomain` (optional)
Derived from	-
Description	Radar Sensor Ambiguity Domain informaiton if the sensor has related capability.

⌋(*RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.243 RadarDetections

[SWS_ADI_00714]{DRAFT} ⌈

Name	RadarDetections
Kind	STRUCTURE
Subelements	Header `GeneralHeader` SensorMountingPos `SensorMountingInfo` HeaderAmbiguityInfo `AmbiguityDomainInfo` RDetectionsInfo `RadarDetectionsInfo`
Derived from	-
Description	Represents the radar detection interface information.

⌋(*RS_ADI_00003*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.244 LidarDetectionInfo

[SWS_ADI_00715]{DRAFT} ⌈

Name	LidarDetectionInfo
Kind	STRUCTURE
Subelements	PositionSpheric `Position3DSpheric` PositionSphericError `Position3DSphericError` HeightLidar `float` (optional) HeightLidarError `float` (optional) RadialRelVelocity `float` (optional) RadialRelVelocityError `float` (optional) Reflectivity `float`

▽

▽

△

	△ ReflectivityError `float` (optional) FreeSpacePro `ProPercentage` (optional) LidarClassType `DetectionClassType` (optional)
Derived from	-
Description	Represents the lidar detection information.

⌋(*RS_ADI_00002*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.245 LidarDetection

[SWS_ADI_00716]{DRAFT} ⌈

Name	LidarDetection
Kind	STRUCTURE
Subelements	LDetectionStatus `DetectionStatus` LDetetionInfo `LidarDetectionInfo`
Derived from	-
Description	Represents the lidar detection information.

⌋(*RS_ADI_00002*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.246 LidarDetectionVector

[SWS_ADI_00717]{DRAFT} ⌈

Name	LidarDetectionVector
Kind	VECTOR
Subelements	`LidarDetection`
Derived from	-
Description	Represents a list of lidar detection information.

⌋(*RS_ADI_00002*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.247 LidarDetectionsInfo

[SWS_ADI_00718]{DRAFT} ⌈

Name	LidarDetectionsInfo
Kind	STRUCTURE
Subelements	RecognizedDetectionsCap `uint_32` (optional) RecognizedDetectionsStatus `RecognizedStatus` (optional) NoValidDetections `uint_32` LidarDetectionList `LidarDetectionVector`
Derived from	-
Description	Represents the lidar detections information.

⌋(*RS_ADI_00002*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.248 LidarDetections

[SWS_ADI_00719]{DRAFT} ⌈

Name	LidarDetections
Kind	STRUCTURE
Subelements	Header `GeneralHeader` SensorMountingPos `SensorMountingInfo` LDetectionsInfo `LidarDetectionsInfo`
Derived from	-
Description	Represents the lidar detection interface information.

⌋(*RS_ADI_00002*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.249 CDShapeClassTypeInfo

[SWS_ADI_00720]{DRAFT} ⌈

Name	CDShapeClassTypeInfo
Kind	STRUCTURE
Subelements	DetectionShapClassType `ShapeClassType` DetectionShapeClassVariant `ShapeClassVariant`
Derived from	-
Description	Represents the shape class type information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.250 CDShapeClassTypeInfoVector

[SWS_ADI_00721]{DRAFT} ⌈

Name	CDShapeClassTypeInfoVector
Kind	VECTOR
Subelements	CDShapeClassTypeInfo
Derived from	-
Description	Represents a list of shape class type information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.251 CameraDetectionInfo

[SWS_ADI_00722]{DRAFT} ⌈

Name	CameraDetectionInfo
Kind	STRUCTURE
Subelements	DetectionShapeType ShapeType FreeSpacePro ProPercentage (optional) NoValidShapeClass uint8_t ShapeClassTypeList CDShapeClassTypeInfoVector ShapeAmbiguityID uint16_t (optional) ColourValueList ColourValueVector ColourConfidence ProPercentage (optional)
Derived from	-
Description	Represents the Camera detection related information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.252 CDShapePoint

[SWS_ADI_00723]{DRAFT} ⌈

Name	CDShapePoint
Kind	STRUCTURE
Subelements	PointProValue ProPercentage PositionSpheric Position3DSpheric PositionSphericError Position3DSphericError
Derived from	-
Description	Represents the Shape point information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.253 CDShapePointVectorType

[SWS_ADI_00724]{DRAFT} ⌈

Name	CDShapePointVector
Kind	VECTOR
Subelements	`CDShapePoint`
Derived from	-
Description	Represents a list of shape point information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.254 CDShapePointsInfo

[SWS_ADI_00725]{DRAFT} ⌈

Name	CDShapePointsInfo
Kind	STRUCTURE
Subelements	NoValidShapePoints `uint16_t` CDShapePointsList `CDShapePointVector`
Derived from	-
Description	Represents the Shape points related information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.255 CameraDetection

[SWS_ADI_00726]{DRAFT} ⌈

Name	CameraDetection
Kind	STRUCTURE
Subelements	CDetectionStatus `DetectionStatus` CDetetionInfo `CameraDetectionInfo` CDShapePoints `CDShapePointsInfo`
Derived from	-
Description	Represents the Camera detection information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.256 CameraDetectionVector

[SWS_ADI_00727]{DRAFT} ⌈

Name	CameraDetectionVector
Kind	VECTOR
Subelements	`CameraDetection`
Derived from	-
Description	Represents a list of Camera detection information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.257 CameraDetectionsInfo

[SWS_ADI_00728]{DRAFT} ⌈

Name	CameraDetectionsInfo
Kind	STRUCTURE
Subelements	RecognizedDetectionsCap `uint_32` (optional) RecognizedDetectionsStatus `RecognizedStatus` (optional) NoValidDetections `uint_32` CameraDetectionList `CameraDetectionVector`
Derived from	-
Description	Represents the Camera detections information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.258 CameraDetections

[SWS_ADI_00729]{DRAFT} ⌈

Name	CameraDetections
Kind	STRUCTURE
Subelements	CDHeader `GeneralHeader` SensorMountingPos `SensorMountingInfo` CDetectionsInfo `CameraDetectionsInfo`
Derived from	-
Description	Represents the Camera detection interface information.

⌋(*RS_ADI_00001*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.259 UltrasonicDetectionInfo

[SWS_ADI_00730]{DRAFT} ⌈

Name	UltrasonicDetectionInfo
Kind	STRUCTURE
Subelements	Reflectivity `float` (optional) EllipseSemiMinorAxis `float` EllipseSemiMajorAxis `float` (optional) UltrasonicEllipseAxesError `EllipseAxesError` HeightUltrasonic `float` (optional) HeightUltrasonicError `float` (optional)
Derived from	-
Description	Represents the Ultrasonic detection information.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.260 UltrasonicDetection

[SWS_ADI_00731]{DRAFT} ⌈

Name	UltrasonicDetection
Kind	STRUCTURE
Subelements	UltrasonicDetectionStatus `DetectionStatus` UltrasonicDetetionInfo `UltrasonicDetectionInfo`
Derived from	-
Description	Represents the Ultrasonic detection information.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.261 UltrasonicDetectionVector

[SWS_ADI_00732]{DRAFT} ⌈

Name	UltrasonicDetectionVector
Kind	VECTOR
Subelements	`UltrasonicDetection`
Derived from	-
Description	Represents a list of Ultrasonic detection information.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.262 UltrasonicDetectionsInfo

[SWS_ADI_00733]{DRAFT} ⌈

Name	UltrasonicDetectionsInfo
Kind	STRUCTURE
Subelements	RecognizedDetectionsCap `uint_32` (optional) RecognizedDetectionsStatus `RecognizedStatus` (optional) NoValidDetections `uint_32` UltrasonicDetectionList `UltrasonicDetectionVector`
Derived from	-
Description	Represents the Ultrasonic detections information.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.1.263 UltrasonicDetections

[SWS_ADI_00734]{DRAFT} ⌈

Name	UltrasonicDetections
Kind	STRUCTURE
Subelements	Header `GeneralHeader` SensorMountingPos `SensorMountingInfo` UltrasonicDetectionsInfo `UltrasonicDetectionsInfo`
Derived from	-
Description	Represents the Ultrasonic detection interface information.

⌋(*RS_ADI_00004*, *RS_ADI_00012*, *RS_ADI_00013*, *RS_ADI_00014*)

9.2 Service Interfaces

This chapter lists all provided service interfaces of the `ADI`.

9.2.1 Sensor Interfaces Port

[SWS_ADI_01000]{DRAFT} ⌈

Name	SensorInterface		
Kind	ProvidedPort	***Interface***	`SensorInterface`
Description			
Variation			

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*)

9.2.2 Object level Interfaces

[SWS_ADI_01001]{DRAFT} ⌈

Name	PMObjectsService
NameSpace	ara::adi::sensoritf

Method	Capability	
Description	Get the capability vector of the service.	
FireAndForget	false	
Parameter	capVector	
	Description	The capability vector of the service indicates the presence of the optional signals of the event.
	Type	`CapabilityVector`
	Variation	
	Direction	OUT

Events	PMObjects
Description	The potentially moving object list is reported by a sensor in a measurement cycle.
Type	`PMObjects`

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*)

[SWS_ADI_01002]{DRAFT} ⌈

Name	RObjectsService
NameSpace	ara::adi::sensoritf

Method	Capability	
Description	Get the capability vector of the service.	
FireAndForget	false	
Parameter	capVector	
	Description	The capability vector of the servie indicates the presence of the optional signals of the event.
	Type	`CapabilityVector`
	Variation	
	Direction	OUT

Events	RoadObject
Description	The road object list is reported by a sensor in a measurement cycle.
Type	`RObjects`

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*)

[SWS_ADI_01003]{DRAFT} ⌈

Name	SObjectsService
NameSpace	ara::adi::sensoritf

Method	Capability	
Description	Get the capability vector of the service.	
FireAndForget	false	
Parameter	capVector	
	Description	The capability vector of the servie indicates the presence of the optional signals of the event.
	Type	`CapabilityVector`
	Variation	
	Direction	OUT

Events	StaticObject
Description	The static object list is reported by a sensor in a measurement cycle.
Type	`StaticObjects`

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*)

9.2.3 Feature level Interfaces

[SWS_ADI_01004]{DRAFT} ⌈

Name	CameraFeaturesService
NameSpace	ara::adi::sensoritf

Method	Capability
Description	Get the capability vector of the service.
FireAndForget	false
Parameter	capVector

▽

△

	Description	The capability vector of the servie indicates the presence of the optional signals of the event.
	Type	`CapabilityVector`
	Variation	
	Direction	OUT

Events	CameraFeatureEvent
Description	The camera feature list is reported by a sensor during one measurement cycle.
Type	`CameraFeatures`

⌋(*RS_ADI_00001*)

[SWS_ADI_01005]{DRAFT} ⌈

Name	UltrasonicFeaturesService
NameSpace	ara::adi::sensoritf

Method	Capability	
Description	Get the capability vector of the service.	
FireAndForget	false	
Parameter	capVector	
	Description	The capability vector of the servie indicates the presence of the optional signals of the event.
	Type	`CapabilityVector`
	Variation	
	Direction	OUT

Events	UltrasonicFeaturesEvent
Description	The Ultrasonic Feature list is reported by a sensor in a measurement cycle.
Type	`UltrasonicFeatures`

⌋(*RS_ADI_00004*)

9.2.4 Detection level Interfaces

[SWS_ADI_01006]{DRAFT} ⌈

Name	RadarDetectionsService
NameSpace	ara::adi::sensoritf

Method	Capability	
Description	Get the capability vector of the service.	
FireAndForget	false	
Parameter	capVector	
	Description	The capability vector of the servie indicates the presence of the optional signals of the event.
	Type	`CapabilityVector`
	Variation	
	Direction	OUT

Events	RadarDetectionsEvent
Description	The radar detection list is reported by a sensor in a measurement cycle.
Type	`RadarDetections`

⌋(*RS_ADI_00003*)

[SWS_ADI_01007]{DRAFT} ⌈

Name	LidarDetectionsService
NameSpace	ara::adi::sensoritf

Method	Capability	
Description	Get the capability vector of the service.	
FireAndForget	false	
Parameter	capVector	
	Description	The capability vector of the servie indicates the presence of the optional signals of the event.
	Type	`CapabilityVector`
	Variation	
	Direction	OUT

Events	LidarDetectionsEvent
Description	The Lidar detection list is reported by a sensor in a measurement cycle.
Type	`LidarDetections`

⌋(*RS_ADI_00002*)

[SWS_ADI_01008]{DRAFT} ⌈

Name	CameraDetectionsService
NameSpace	ara::adi::sensoritf

Method	Capability	
Description	Get the capability vector of the service.	
FireAndForget	false	
Parameter	capVector	
	Description	The capability vector of the servie indicates the presence of the optional signals of the event.
	Type	`CapabilityVector`
	Variation	
	Direction	OUT

Events	CameraDetectionsEvent
Description	The camera detection list is reported by a sensor in a measurement cycle.
Type	`CameraDetections`

⌋(*RS_ADI_00001*)

[SWS_ADI_01009]{DRAFT} ⌈

Name	UltrasonicDetectionsService
NameSpace	ara::adi::sensoritf

Method	Capability	
Description	Get the capability vector of the service.	
FireAndForget	false	
Parameter	capVector	
	Description	The capability vector of the servie indicates the presence of the optional signals of the event.
	Type	`CapabilityVector`
	Variation	
	Direction	OUT

Events	UltrasonicDetectionsEvent
Description	The Ultrasonic Detection list is reported by a sensor in a measurement cycle.
Type	`UltrasonicDetections`

⌋(*RS_ADI_00004*)

9.2.5 Supportive Interfaces

[SWS_ADI_01010]{DRAFT} ⌈

Name	SensorPerformanceService
NameSpace	ara::adi::sensoritf

Method	Capability	
Description	Get the capability vector of the service.	
FireAndForget	false	
Parameter	capVector	
	Description	The capability vector of the servie indicates the presence of the optional signals of the event.
	Type	`CapabilityVector`
	Variation	
	Direction	OUT

Events	SPerformance
Description	The sensor performance information is reported by a sensor.
Type	`SensorPerformance`

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*)

[SWS_ADI_01011]{DRAFT} ⌈

Name	SensorHealthService
NameSpace	ara::adi::sensoritf

Method	Capability	
Description	Get the capability vector of the service.	
FireAndForget	false	
Parameter	capVector	
	Description	The capability vector of the servie indicates the presence of the optional signals of the event.
	Type	`CapabilityVector`
	Variation	
	Direction	OUT

Events	SHealth
Description	The sensor health information is reported by a sensor.
Type	`SensorHealthInfo`

⌋(*RS_ADI_00001*, *RS_ADI_00002*, *RS_ADI_00003*, *RS_ADI_00004*)

10 Capability Configuration

10.1 Object Level Service

10.1.1 PMObjectsService Capability Vector

The table below includes the capability bit setting for the optional elements for PMObjectsService, which also refers to ISO 23150.The Bit setting to 1 means the presence of the optional element, while 0 means absent.

Bit	Reference Singal in ISO23150	Reference Element in PMObjectService	Option
0	Interface ID (A.1.3)	InterfaceID	
1	Cycle counter (A.1.5.1)	CycleCounter in HeaderBasic	
2	Interface cycle time (A.1.6)	ItfCycleTime in HeaderBasic	
3	Interface cycle time variation (A.1.7)	ItfCycleTimeV in HeaderBasic	
4	Tracking motion model (A.1.12)	TrackingMotionModel	
5	Motion type (A.1.13)	MotionType	Madatory except USS
6	Recognized potentially moving objects capability (A.1.9.1)	PMOCap in PMObjects	
7	Recognized potentially moving objects status (A.1.10.1)	RecognizedStatus	
8	Object grouping ID (A.2.2)	GroupingObjectID in ObjectStatus	
9	Observation object level (A.2.4)	Observation in ObjectStatus	
10	Track quality (A.2.5)	TrackQuality in ObjectStatus	
11	Reference point (A.2.7)	ReferencePoints	
12	Position (x, y, z) (A.2.8) z vector	PMOPosition in PMObjectDynamics	
13	Orientation (yaw, pitch, roll)(A.2.10)	PMOrientation in PMObjectDynamics	
14	Velocity (x, y, z) (A.2.12)	PMOVelocity in PMObjectDynamics	
15	Acceleration (x, y, z) (A.2.14)	PMOAcceleration in PMObjectDynamics	
16	Kinematic point (x, y, z) (A.2.16)	KinematicPoint in PMObjectDynamics	depend on bit 17 (A.2.18)
17	Orientation rate kinematic point yaw (A.2.18)	KPOrientation in PMObjectDynamics	
18	Road level (A.2.20)	PMORoadLevel in PMObjectDynamics (Roadlevel)	
19	Movement status (A.2.21)	PMOMovementStatus in PMObjectDynamics (MovementStatus)	
20	Potentially moving object bounding box LSG	PMOBoundingBox	Relevant:Camera
21	Dimension (length, width, height) (A.2.22) Height only	DimensionBox in PMOBoundingBox	subset of bit 20
22	Bounding box ground clearance (A.2.24)	GroundClearance in PMOBoundingBox	subset of bit 20
23	Included geometric structures (A.2.25)	GeometricStructures in PMOBoundingBox	subset of bit 20
24	Potentially moving object lights LSG	PMOLights	Relevant:Camera
25	Potentially moving object type person LSG	PMOPersonPose	Relevant:Camera
26	Lane related information LSG	LaneInfo	LSG
27	Angle to object edge to lane (left edge right lane, right edge left lane)(A.2.38)	Angle2ObjectEdge2Line in LaneInfo	subset of bit 26

▽

△

Bit	Reference Singal in ISO23150	Reference Element in PMObjectService	Option
28	Percentage side lane (left, right)(A.2.40)	PercentageSideLane in LaneInfo	subset of bit 26
29	Motion related information LSG	MotionInfo	LSG
30	Scale change object level (A.2.43)	ScaleChange in PMObject	Relevant: camera
31	Entity radar cross section (A.2.44)	EntityRCS in PMObject	Relevant: radar
32	Entity lidar reflectivity (A.2.45)	EntityLidarReflectivity in PMObject	Relevant: lidar
33	Position error (x, y, z) (A.2.9)	PMOPositionError in PMObjectDynamics	
34	Orientation error (yaw, pitch, roll) (A.2.11)	PMOrientationError in PMObjectDynamics	
35	Velocity error (x, y, z) (A.2.13)	PMOVelocityError in PMObjectDynamics	
36	Acceleration error (x, y, z) (A.2.15)	PMOAccelerationError in PMObjectDynamics	
37	Kinematic point error (x, y, z) (A.2.17)	KPError in PMObjectDynamics	
38	Orientation rate kinematic point error yaw (A.2.19)	KPOrentationError in PMObjectDynamics	
39	Dimension error (length, width, height) (A.2.23)	PMOBoundingBox in PMOBoundingBox	subset of bit 20
40	Person pose error (yaw, pitch, roll) (A.2.36)	PoseError in Pose	subset of bit 25
41	Angle to object edge to lane error (left edge right lane, right edge left lane) (A.2.39)	Angle2ObjectEdge2LineError in LaneInfo	subset of bit 26

Table 10.1: Capability Vector of PMObjectService

10.1.2 RObjectsService Capability Vector

The table below includes the capability bit setting for the optional elements for RObjectService, which also refers to ISO 23150.The Bit setting to 1 means the presence of the optional element, while 0 means absent.

Bit	Reference Singal in ISO23150	Reference Element in RObjectsService	Option
0	Interface ID (A.1.3)	InterfaceID	
1	Cycle counter (A.1.5.1)	CycleCounter in HeaderBasic	
2	Interface cycle time (A.1.6)	ItfCycleTime in HeaderBasic	
3	Interface cycle time variation (A.1.7)	ItfCycleTimeV in HeaderBasic	
4	Tracking motion model (A.1.12)	TrackingMotionModel	
5	Colour model type (A.1.14)	ColourModelType	Relevant for Camera
6	Road surface list	RoadSurface	optional list
7	Number of valid road surface condition classifications (A.2.50)	NoConditionClass in RoadSurface	subset of bit 6
8	Road surface roughness (A.2.53)	SurfaceRoughness in RoadSurface	subset of bit 6
9	Road markings list	Roadmarkings	optional list
10	Recognized road markings capability (A.1.9.2)	RecognizedRoadMarkingCap in Roadmarkings	subset of bit 9
11	Recognized road markings status (A.1.10.2)	RecognizedStatus in Roadmarkings	subset of bit 9
12	Object grouping ID (A.2.2)	GroupingObjectID in ObjectStatus in Roadmarking	subset of bit 9

▽

△

Bit	Reference Singal in ISO23150	Reference Element in RObjectsService	Option
13	Observation object level (A.2.4)	Observation in ObjectStatus in Roadmarking	subset of bit 9
14	Track quality (A.2.5)	TrackQuality in ObjectStatus in Roadmarking	subset of bit 9
15	Arrow orientation (A.2.57)	ArrowOrientation in ROMarkingT	subset of bit 9
16	Sign type (A.2.59)	SignType in ROMarkingT	subset of bit 9
17	Sign type confidence (A.2.60)	SignTypeConfidence in ROMarkingT	subset of bit 9
18	Sign value (A.2.61)	SignValue in ROMarkingT	subset of bit 9
19	Sign value unit (A.2.62)	SignUnit in ROMarkingT (SignValueUnit)	subset of bit 9
20	Colour classification confidence (A.2.64)	ColourConfidence in ROMarkingT	subset of bit 9
21	Road marking polynomials LSG	ROPolylines in Roadmarking (PolynomialsInfo)	LSG Relevant camera, lidars; subset of bit 9
22	Width polynomial (A.2.72)	WidthPolynomials in PolynomialsInfo	subset of bit 21
23	Width polynomial confidence (A.2.74)	WidthPolynomialsConfidence in PolynomialsInfo	subset of bit 21
24	Height polynomial (A.2.75)	HeightPolynomials in PolynomialsInfo	subset of bit 21
25	Height polynomial confidence (A.2.77)	HeightPolynomialsConfidence in PolynomialsInfo	subset of bit 21
26	Number of valid data ranges (A.2.78)	NoDataRanges in PolynomialsInfo	subset of bit 21
27	Road marking polyline	ROPolylines in Roadmarking (Polylines)	LSG Relevant camera, lidars; subset of bit 9
28	Vertex point confidence (x,y,z) (A.2.85)	Confidence in VertexPointInfo	subset of bit 27
29	Width vertex confidence (A.2.88)	WidthVertexConfidence in VertexPointInfo	subset of bit 27
30	Height vertex (A.2.89)	HeightVertex in VertexPointInfo	subset of bit 27
31	Height vertex confidence (A.2.91)	HeightVertexConfidence in VertexPointInfo	subset of bit 27
32	Road boundaries	RoadBoundaries	Optional list
33	Recognized road boundaries capability (A.1.9.3)	RecognizedRoadMarkingCap in RoadBoundaries	subset of bit 32
34	Recognized road boundaries status (A.1.10.2)	RecognizedStatus in RoadBoundaries	subset of bit 32
35	Object grouping ID (A.2.2)	GroupingObjectID in ROBoundaryStatus(ObjectStatus) in RoadBoundaries	subset of bit 32
36	Observation object level (A.2.4)	Observation in ROBoundaryStatus(ObjectStatus) in RoadBoundaries	subset of bit 32
37	Track quality (A.2.5)	TrackQuality in ROBoundaryStatus(ObjectStatus) in RoadBoundaries	subset of bit 32
38	Lane boundary role (A.2.95)	Role in ROBoundaryType(LaneBoundaryRole)	subset of bit 32
39	Lane boundary role confidence (A.2.96)	Role in ROBoundaryType(LaneBoundaryRole)	subset of bit 32
40	Road boundary properties	RBProperties (RoadBoundaryProperties) in ROBoundaryType	LSG Relevant: camera, subset of bit 32
41	Colour classification confidence (A.2.64)	ColourConfidence in RoadBoundaryProperties	subset of bit 41
42	Road boundary polynomials	RBPolynomialsInfor in RoadBoundaryInfo (PolynomialsInfo)	LSG Relevant: camera, lidar, radar,subset of bit 32

▽

△

Bit	Reference Singal in ISO23150	Reference Element in RObjectsService	Option
43	Width polynomial (A.2.72)	WidthPolynomials in PolynomialsInfo	subset of bit 42
44	Width polynomial confidence (A.2.74)	WidthPolynomialsConfidence in PolynomialsInfo	subset of bit 42
45	Height polynomial (A.2.75)	HeightPolynomials in PolynomialsInfo	subset of bit 42
46	Height polynomial confidence (A.2.77)	HeightPolynomialsConfidence in PolynomialsInfo	subset of bit 42
47	Number of valid data ranges (A.2.78)	NoDataRanges in PolynomialsInfo	subset of bit 42
48	Road boundary polyline	RBPolylines in RoadBoundaryInfo	LSG Relevant: camera, lidar, radar subset of bit 32
49	Vertex point confidence (x,y,z) (A.2.85)	Confidence in VertexPointInfo	subset of bit 48
50	Width vertex (A.2.86)	WidthVertex in VertexPointInfo	subset of bit 48
51	Width vertex confidence (A.2.88)	WidthVertexConfidence in VertexPointInfo	subset of bit 48
52	Height vertex (A.2.89)	HeightVertex in VertexPointInfo	subset of bit 48
53	Height vertex confidence (A.2.91)	HeightVertexConfidence in VertexPointInfo	subset of bit 48
54	Polynomial error (A.2.70)	PolynomialError in PolynomialsInfo	subset of bit 21
55	Width polynomial error (A.2.73)	WidthPolynomialsError in PolynomialsInfo	subset of bit 21
56	Height polynomial error (A.2.76)	HeightPolynomialsError in PolynomialsInfo	subset of bit 21
57	Width vertex error (A.2.87)	WidthVertexError in VertexPointInfo	subset of bit 27
58	Height vertex error (A.2.90)	HeightVertexError in VertexPointInfo	subset of bit 27
59	Polynomial error (A.2.70)	PolynomialError in PolynomialsInfo	subset of bit 42
60	Width polynomial error (A.2.73)	WidthPolynomialsError in PolynomialsInfo	subset of bit 42
61	Height polynomial error (A.2.76)	HeightPolynomialsError in PolynomialsInfo	subset of bit 42
62	Width vertex error (A.2.87)	WidthVertexError in VertexPointInfo	subset of bit 48
63	Height vertex error (A.2.90)	HeightVertexError in VertexPointInfo	subset of bit 48

Table 10.2: Capability Vector of RObjectsService

10.1.3 SObjectsService Capability Vector

The table below includes the capability bit setting for the optional elements for SObjectsService, which also refers to ISO 23150.The Bit setting to 1 means the presence of the optional element, while 0 means absent.

Bit	Reference Singal in ISO23150	Reference Element in SObjectsService	Option
0	Interface ID (A.1.3)	InterfaceID	
1	Cycle counter (A.1.5.1)	CycleCounter in HeaderBasic	
2	Interface cycle time (A.1.6)	ItfCycleTime in HeaderBasic	
3	Interface cycle time variation (A.1.7)	ItfCycleTimeV in HeaderBasic	
4	General landmarks	GeneralLandmarks	optional list
5	Traffic signs	TrafficSigns	optional list
6	Traffic lights	TrafficLights	optional list

▽

△

Bit	Reference Singal in ISO23150	Reference Element in SObjectsService	Option
7	Recognized general landmarks capability (A.1.9.1)	RecognizedGeneralLandmarksCap in GeneralLandmarks	subset of bit 4
8	Recognized general landmarks status (A.1.10.1)	RecognizedGeneralLandmarksStatus (RecognizedStatus) in GeneralLandmarks	subset of bit 4
9	Object grouping ID (A.2.2)	GroupingObjectID in ObjectStatus of GeneralLandmarks	subset of bit 4
10	Observation object level (A.2.4)	Observation in ObjectStatus of GeneralLandmarks	subset of bit 4
11	Track quality (A.2.5)	TrackQuality in ObjectStatus of GeneralLandmarks	subset of bit 4
12	Reference point (A.2.7)	RPs (ReferencePoints) in Landmarkposition	subset of bit 4
13	Orientation object level-vertical (A.2.101)	OrientationVertical in Landmarkposition	subset of bit 4
14	General landmark object bounding box LSG	LandmarkBoundingbox (BboxInfo) in Landmark	Relevant Camera,LSG,subset of bit 4,
15	Dimension (length, width, height) (A.2.22) height	BoxDimension in BboxInfo	subset of bit 14
16	Recognized traffic signs capability (A.1.9.1)	RecognizedTrafficSignsCap in TrafficSigns	subset of bit 5
17	Recognized traffic signs status (A.1.10.1)	RecognizedTrafficSignsStatus (RecognizedStatus) in TrafficSigns	subset of bit 4
18	Object grouping ID (A.2.2)	GroupingObjectID in MainSignstatus (ObjectStatus) of TrafficSign	subset of bit 5
19	Observation object level (A.2.4)	Observation in MainSignstatus (ObjectStatus) of TrafficSign	subset of bit 5
20	Track quality (A.2.5)	TrackQuality in MainSignstatus (ObjectStatus) of TrafficSign	subset of bit 5
21	Recognized traffic lights capability (A.1.9.1)	RecognizedTrafficLightsCap in TrafficSigns	subset of bit 6
22	Recognized traffic lights status (A.1.10.1)	RecognizedTrafficLightsStatus (RecognizedStatus) in TrafficSigns	subset of bit 6
23	Object grouping ID (A.2.2)	GroupingObjectID in TLightStatus (ObjectStatus) of TrafficLight	subset of bit 6
24	Observation object level (A.2.4)	Observation in TLightStatus (ObjectStatus) of TrafficLight	subset of bit 6
25	Track quality (A.2.5)	TrackQuality in TLightStatus (ObjectStatus) of TrafficLight	subset of bit 6
26	Reference point (A.2.7)	RPs (ReferencePoints) in TrafficLightPos	subset of bit 6
27	Orientation object level - vertical (A.2.101)	OrientationVertical in TrafficLightPos	subset of bit 6
28	Traffic light bounding box LSG	TLightBox (BboxInfo)	LSG subset of bit 6 Relevant Camera
29	Dimension (length, width, height) (A.2.22) length	BoxDimension (DimensionBox) in BboxInfo	subset of bit 28
30	Total number of traffic light spots (A.2.116)	NoTrafficLightSpots in TrafficLightSpots	subset of bit 6
31	Total number of traffic light spots confidence (A.2.117)	NoTrafficLightSpotsConfidence in TrafficLightSpots	subset of bit 6
32	Observation object level (A.2.4)	Observation in TLightSpotStatus (Observation)	subset of bit 6
33	Track quality (A.2.5)	TrackQuality in TLightSpotStatus (Observation)	subset of bit 6
34	Light shape value (A.2.129)	LightShapeValue in TrafficLightSpotShape	subset of bit 6

▽

△

Bit	Reference Singal in ISO23150	Reference Element in SObjectsService	Option
35	Orientation object level error - vertical (A.2.102)	OrientationVerticalError in Landmarkposition	subset of bit 4
36	Dimension error (length, width, height) (A.2.23)	BoxDimension in BboxInfo	subset of bit 14
37	Orientation object level error - vertical (A.2.102)	OrientationVerticalError in TrafficLightPos	subset of bit 6
38	Dimension error (length, width, height) (A.2.23)	BoxError in BboxInfo	subset of bit 28

Table 10.3: Capability Vector of SObjectsService

10.2 Feature Level Service

10.2.1 CameraFeatureService Capability Vector

The table below includes the capability bit setting for the optional elements for CameraFeatureService, which also refers to ISO 23150.The Bit setting to 1 means the presence of the optional element, while 0 means absent.

Bit	Reference Singal in ISO23150	Reference Element in CameraFeatureService	Option
0	Interface ID (A.1.3)	InterfaceID	
1	Cycle counter (A.1.5.1)	CycleCounter in HeaderBasic	
2	Interface cycle time (A.1.6)	ItfCycleTime in HeaderBasic	
3	Interface cycle time variation (A.1.7)	ItfCycleTimeV in HeaderBasic	
4	Recognized features capability (A.1.9.1)	RecognizedFeaturesCap in CameraFeatures	
5	Recognized features status (A.1.10.1)	RecognizedFeaturesStatus (RecognizedStatus) in CameraFeatures	
6	Feature ID (A.3.2)	FeatureID in FeatureStatus of CameraFeature	
7	Observation feature level (A.3.5)	Observation in FeatureStatus of CameraFeature	
8	Object ID reference feature level (A.3.3)	ObjectID in FeatureStatus of CameraFeature	
9	Colour probability feature level (A.3.11)	ColourPro in ShapeInfo	
10	Shape reference points LSG	CFShapeReferPoints (ShapeReferPoints) in CameraFeature	
11	Orientation normal (x,y,z) (A.3.17)	OrientationNormal in ShapeReferPointInfo	subset of bit 10
12	Orientation normal error (x,y,z) (A.3.18)	OrientationNormalError in ShapeReferPointInfo	subset of bit 10
13	Scaled translation rate (x,y,z) (A.3.19)	TranslationRate in ShapeReferPointInfo	subset of bit 10
14	Scaled translation rate error (x,y,z) (A.3.20)	TranslationRateError in ShapeReferPointInfo	subset of bit 10
15	Rotation rate (yaw,pitch,roll) (A.3.21)	RotationRate in ShapeReferPointInfo	subset of bit 10
16	Rotation rate error (yaw,pitch,roll) (A.3.22)	RotationRateError in ShapeReferPointInfo	subset of bit 10
17	Scale change feature level (A.3.23)	ScaleChange in ShapeReferPointInfo	subset of bit 10
18	Scale change feature level error (A.3.24)	ScaleChangeError in ShapeReferPointInfo	subset of bit 10

Table 10.4: Capability Vector of CameraFeatureService

10.2.2 USSFeatureService Capability Vector

The table below includes the capability bit setting for the optional elements for USSFeatureService, which also refers to ISO 23150.The Bit setting to 1 means the presence of the optional element, while 0 means absent.

Bit	Reference Singal in ISO23150	Reference Element in USSFeatureService	Option
0	Interface ID (A.1.3)	InterfaceID	
1	Cycle counter (A.1.5.1)	CycleCounter in HeaderBasic	
2	Interface cycle time (A.1.6)	ItfCycleTime in HeaderBasic	
3	Interface cycle time variation (A.1.7)	ItfCycleTimeV in HeaderBasic	
4	Recognized features capability (A.1.9.1)	RecognizedFeaturesCap in USSFeatures	
5	Recognized features status (A.1.10.1)	RecognizedFeaturesStatus (RecognizedStatus) in USSFeatures	
6	Feature ID (A.3.2)	FeatureID in FeatureStatus of USSFeature	
7	Observation feature level (A.3.5)	Observation in FeatureStatus of USSFeature	
8	Object ID reference feature level (A.3.3)	ObjectID in FeatureStatus of USSFeature	
9	Number of valid points (A.3.27)	NoValidPointsUSS in SegmentPointsInfo	
10	Orientation feature level - vertical (A.3.30)	OrientationUSS in ValidPointInfo	
11	Orientation feature level error - vertical (A.3.31)	OrientationUSSError in ValidPointInfo	
12	Height feature level (A.3.32)	HeightUSS in ValidPointInfo	
13	Height feature level error (A.3.33)	HeightUSSError in ValidPointInfo	
14	Velocity (x, y) (A.3.34)	VelocityUSS in ValidPointInfo	
15	Velocity error (x, y) (A.3.35)	VelocityUSSError in ValidPointInfo	
16	Measurement status feature level (A.3.37)	FeatureMStatus in ValidPointInfo	

Table 10.5: Capability Vector of USSFeatureService

10.3 Detection Level Service

10.3.1 RadarDetectionService Capability Vector

The table below includes the capability bit setting for the optional elements for RadarDetectionService, which also refers to ISO 23150.The Bit setting to 1 means the presence of the optional element, while 0 means absent.

Bit	Reference Singal in ISO23150	Reference Element in RadarDetectionService	Option
0	Interface ID (A.1.3)	InterfaceID	
1	Cycle counter (A.1.5.1)	CycleCounter in HeaderBasic	
2	Interface cycle time (A.1.6)	ItfCycleTime in HeaderBasic	
3	Interface cycle time variation (A.1.7)	ItfCycleTimeV in HeaderBasic	
4	Radial velocity ambiguity domain (begin, end) (A.1.15)	VelocityAmbiguity (RadialVelocityAmbiguityDomain) in AmbiguityDomainInfo	

▽

△

Bit	Reference Singal in ISO23150	Reference Element in RadarDetectionService	Option
5	Range ambiguity domain (begin, end) (A.1.16)	RangeAmbiguity (RangeAmbiguityDomain) in AmbiguityDomainInfo	
6	Angle azimuth ambiguity domain (begin, end) (A.1.17)	AzimuthAmbiguity (AngleAzimuthAmbiguityDomain) in AmbiguityDomainInfo	
7	Angle elevation ambiguity domain (begin, end) (A.1.18)	ElevationAmbiguity (AngleElevationAmbiguityDomain) in AmbiguityDomainInfo	
8	Recognized detections capability (A.1.9.8)	RecognizedDetectionsCap in RadarDetectionsInfo	
9	Recognized detections status (A.1.10.8)	RecognizedDetectionsStatus (RecognizedStatus) in RadarDetectionsInfo	
10	Object ID reference detection level (A.4.2)	ObjectID in DetectionStatus	
11	Radial relative velocity error (A.4.8)	RadialRelVelocityError in RadarDetectionInfo	
12	Radar cross section error (A.4.10)	RCSError in RadarDetectionInfo	
13	Signal to noise ratio detection level error (A.4.12)	SNRError in RadarDetectionInfo	
14	Multi target probability (A.4.13)	MultiTargetPro in RadarDetectionInfo	
15	Ambiguity ID (A.4.14)	AmbiguityID in RadarDetectionInfo	
16	Detection ambiguity probability (A.4.15)	DetectionAmbiguityPro in RadarDetectionInfo	
17	Free space probability (A.4.16)	FreeSpacePro in RadarDetectionInfo	
18	Number of valid detection classifications (A.4.17)	NoValidDetectionClass in RadarDetectionInfo	
19	Detection classification type (A.4.18)	DClassType (DetectionClassType)in DetectionClassTypeInfo	
20	Detection classification probability value (A.4.19)	DetectionClassPro in DetectionClassTypeInfo	
21	Sensor origin position error (x,y,z) (A.1.22)	OriginPositionError in MountingInfo	
22	Sensor orientation error (yaw,pitch,roll) (A.1.24)	OriginOrientationError in MountingInfo	

Table 10.6: Capability Vector of RadarDetectionService

10.3.2 LidarDetectionService Capability Vector

The table below includes the capability bit setting for the optional elements for LidarDetectionService, which also refers to ISO 23150.The Bit setting to 1 means the presence of the optional element, while 0 means absent.

Bit	Reference Singal in ISO23150	Reference Element in LidarDetectionService	Option
0	Interface ID (A.1.3)	InterfaceID	
1	Cycle counter (A.1.5.1)	CycleCounter in HeaderBasic	
2	Interface cycle time (A.1.6)	ItfCycleTime in HeaderBasic	
3	Interface cycle time variation (A.1.7)	ItfCycleTimeV in HeaderBasic	
4	Recognized detections capability (A.1.9.8)	RecognizedDetectionsCap in LidarDetectionsInfo	

▽

△

Bit	Reference Singal in ISO23150	Reference Element in LidarDetectionService	Option
5	Recognized detections status (A.1.10.8)	RecognizedDetectionsStatus (RecognizedStatus) in LidarDetectionsInfo	
6	Object ID reference detection level (A.4.2)	ObjectID in DetectionStatus	
7	Height lidar (A.4.20)	HeightLidar in LidarDetectionInfo	
8	Height lidar error (A.4.21)	HeightLidarError in LidarDetectionInfo	depends on bit7
9	Radial relative velocity (A.4.7)	RadialRelVelocity in LidarDetectionInfo	
10	Radial relative velocity error (A.4.8)	RadialRelVelocityError in LidarDetectionInfo	
11	Reflectivity error (A.4.23)	ReflectivityError in LidarDetectionInfo	
12	Free space probability (A.4.16)	FreeSpacePro in LidarDetectionInfo	
13	Detection classification type (A.4.18)	LidarClassType in LidarDetectionInfo	
14	Sensor origin position error (x,y,z) (A.1.22)	OriginPositionError in MountingInfo	
15	Sensor orientation error (yaw,pitch,roll) (A.1.24)	OriginOrientationError in MountingInfo	

Table 10.7: Capability Vector of LidarDetectionService

10.3.3 CameraDetectionService Capability Vector

The table below includes the capability bit setting for the optional elements for CameraDetectionService, which also refers to ISO 23150.The Bit setting to 1 means the presence of the optional element, while 0 means absent.

Bit	Reference Singal in ISO23150	Reference Element in CameraDetectionService	Option
0	Interface ID (A.1.3)	InterfaceID	
1	Cycle counter (A.1.5.1)	CycleCounter in HeaderBasic	
2	Interface cycle time (A.1.6)	ItfCycleTime in HeaderBasic	
3	Interface cycle time variation (A.1.7)	ItfCycleTimeV in HeaderBasic	
4	Recognized detections capability (A.1.9.8)	RecognizedDetectionsCap in CameraDetectionsInfo	
5	Recognized detections status (A.1.10.8)	RecognizedDetectionsStatus (RecognizedStatus) in CameraDetectionsInfo	
6	Object ID reference detection level (A.4.2)	ObjectID in DetectionStatus	
7	Feature ID reference (A.4.3)	FeatureID in DetectionStatus	
8	Free space probability (A.4.16)	FreeSpacePro in CameraDetectionInfo	depends on bit7
9	Shape ambiguity ID (A.4.28)	ShapeAmbiguityID in CameraDetectionInfo	
10	Colour probability detection level (A.4.30)	ColourConfidence in CameraDetectionInfo	
11	Distance in Position (Azimuth, Elevation, Distance) (A.4.5) Position error (Azimuth, Elevation, Distance) (A.4.6)	Distance vector of PositionSpheric and PositionSphericError in CDShapePoint	
12	Sensor origin position error (x,y,z) (A.1.22)	OriginPositionError in MountingInfo	
13	Sensor orientation error (yaw,pitch,roll) (A.1.24)	OriginOrientationError in MountingInfo	

Table 10.8: Capability Vector of CameraDetectionService

10.3.4 USSDetectionService Capability Vector

The table below includes the capability bit setting for the optional elements for USSDetectionService, which also refers to ISO 23150.The Bit setting to 1 means the presence of the optional element, while 0 means absent.

Bit	Reference Singal in ISO23150	Reference Element in USSDetectionService	Option
0	Interface ID (A.1.3)	InterfaceID	
1	Cycle counter (A.1.5.1)	CycleCounter in HeaderBasic	
2	Interface cycle time (A.1.6)	ItfCycleTime in HeaderBasic	
3	Interface cycle time variation (A.1.7)	ItfCycleTimeV in HeaderBasic	
4	Recognized detections capability (A.1.9.8)	RecognizedDetectionsCap in USSDetectionsInfo	
5	Recognized detections status (A.1.10.8)	RecognizedDetectionsStatus (RecognizedStatus) in USSDetectionsInfo	
6	Object ID reference detection level (A.4.2)	ObjectID in DetectionStatus	
7	Feature ID reference (A.4.3)	FeatureID in DetectionStatus	
8	Reflectivity (A.4.22)	Reflectivity in USSDetectionInfo	depends on bit7
9	Ellipse semimajor axis (A.4.34)	EllipseSemiMajorAxis in USSDetectionInfo	Relevant virtual sensor
10	Ellipse semiMajor axis error (A.4.35)	SemiMajor of USSEllipseAxesError(USSEllipseAxesError) in USSDetectionInfo	
11	Height ultrasonic (A.4.36)	HeightUSS in USSDetectionInfo	
11	Height ultrasonic error (A.4.37)	HeightUSSError in USSDetectionInfo	
12	Sensor origin position error (x,y,z) (A.1.22)	OriginPositionError in MountingInfo	
13	Sensor orientation error (yaw,pitch,roll) (A.1.24)	OriginOrientationError in MountingInfo	

Table 10.9: Capability Vector of USSDetectionService

10.4 Supportive Service

10.4.1 SensorPerformanceService Capability Vector

The table below includes the capability bit setting for the optional elements for SensorPerformanceService, which also refers to ISO 23150.The Bit setting to 1 means the presence of the optional element, while 0 means absent.

Bit	Reference Singal in ISO23150	Reference Element in SensorPerformanceService	Option
0	Interface ID (A.1.3)	InterfaceID	
1	Cycle counter (A.1.5.1)	CycleCounter in HeaderBasic	
2	Interface cycle time (A.1.6)	ItfCycleTime in HeaderBasic	
3	Interface cycle time variation (A.1.7)	ItfCycleTimeV in HeaderBasic	
4	Scope of interface (A.1.19)	SPScopeIntf ScopeofInterfaces in SensorPerformance	

△

Bit	Reference Singal in ISO23150	Reference Element in SensorPerformance-Service	Option
5	Real world object detection rates	OriginPositionError in MountingInfo	LSG Object level interface
6	Reference target detection rates	OriginOrientationError in MountingInfo	Detection and Feature Level interface LSG
7	Header sensor environmental information LSG	VanishingPointInfo	LSG
8	Vanishing point error (azimuth, elevation) (A.1.26)	VPError VanishingPointDirection2DSphericError in VanishingPointInfo	subset of bit 7
9	Resolution (horizontal, vertical) (A.5.3)	SResolution SensorResolution in SensorPerformanceStatus	
10	Beam divergence (horizontal, vertical) (A.5.4)	BeamDivergence (BeamDivergence) in SensorPerformanceStatus	
11	Range gain (A.5.5)	RangeGain in SensorPerformanceStatus	
12	Minimum classification rate (A.5.13)	MinimunClassRate in ObjectDetectionRateInfo	subset of bit 5
13	Maximum false positive rate (A.5.14)	MaxFalsePositiveRate in ObjectDetectionRateInfo	subset of bit 5
14	Positive predictive value (A.5.15)	PositivePredictiveValue in ObjectDetectionRateInfo	subset of bit 5
15	Radar cross section reference target (A.5.17)	RCSRefTarget in ReferTargetRateInfo	Relevant radar subset of bit 6
16	Reflectivity reference target (A.5.18)	ReflectivityRefTarget in ReferTargetRateInfo	Relevant lidar, ultrasonic subset of bit 6
17	Delta E colour difference (A.5.19)	DeltaEDiff in ReferTargetRateInfo	Relevant camera subset of bit 6
18	Relative speed range (min, max) (A.5.20)	RTSpeedRange RelativeSpeedRange in ReferTargetRateInfo	Relevant radar subset of bit 6
19	Sensor origin position error (x,y,z) (A.1.22)	OriginPositionError in MountingInfo	
20	Sensor orientation error (yaw,pitch,roll) (A.1.24)	OriginOrientationError in MountingInfo	

Table 10.10: Capability Vector of SensorPerformanceService

10.4.2 SensorHealthService Capability Vector

The table below includes the capability bit setting for the optional elements for SensorHealthService, which also refers to ISO 23150.The Bit setting to 1 means the presence of the optional element, while 0 means absent.

Bit	Reference Singal in ISO23150	Reference Element in SensorHealthService	Option
0	Interface ID (A.1.3)	InterfaceID	
1	Cycle counter (A.1.5.1)	CycleCounter in HeaderBasic	
2	Interface cycle time (A.1.6)	ItfCycleTime in HeaderBasic	
3	Interface cycle time variation (A.1.7)	ItfCycleTimeV in HeaderBasic	
4	Sensor origin position error (x,y,z) (A.1.22)	OriginPositionError in MountingInfo	
5	Sensor orientation error (yaw,pitch,roll) (A.1.24)	OriginOrientationError in MountingInfo	

▽

△

Bit	Reference Singal in ISO23150	Reference Element in SensorHealthService	Option
6	Sensor external disturbed (A.5.32)	ExternalDisturbed SensorExternalDisturbed in SensorHealthStatus	
7	Sensor transmit power reduced (A.5.33)	TPReduced SensorTransmitPowerReduced in SensorHealthStatus	
8	Status sensor heating (A.5.34)	HeatingStatus StatusSensorHeating in SensorHealthStatus	
9	Status sensor cleaning (A.5.35)	CleaningStatus StatusSensorCleaning in SensorHealthStatus	
10	Sensor time sync (A.5.36)	TSYStatus StatusTimSync in SensorHealthStatus	
11	Sensor time sync offset value (A.5.37)	SensorTimeSynccOffset in SensorHealthStatus	
12	Calibration	SensorCaliInfo	LSG
13	Correction origin position (x,y,z) (A.5.41)	CoorectionOriginPos in SensorCaliInfo	subset of bit 12
14	Correction origin position error (x,y,z) (A.5.42)	CoorectionOriginPosError in SensorCaliInfo	subset of bit 12
15	Correction position limit minimum (x,y,z) (A.5.43)	CoorectionPosLimitMin in SensorCaliInfo	subset of bit 12
16	Correction position limit maximum (x,y,z) (A.5.44)	DeltaEDiff in SensorCaliInfo	subset of bit 12
17	Correction orientation (yaw,pitch,roll) (A.5.45)	CoorectionPosLimitMax in SensorCaliInfo	subset of bit 12
18	Correction orientation error (yaw,pitch,roll) (A.5.46)	CoorectionOrientation in SensorCaliInfo	subset of bit 12
19	Correction angle limit minimum (yaw,pitch,roll) (A.5.47)	CoorectionOrientationError in SensorCaliInfo	subset of bit 12
20	Correction angle limit maximum (yaw,pitch,roll) (A.5.48)	CoorectionAngleLimitMin in SensorCaliInfo	subset of bit 12
21	Correction orientation (yaw,pitch,roll) (A.5.45)	CoorectionAngleLimitMax in SensorCaliInfo	subset of bit 12

Table 10.11: Capability Vector of SensorHealthService