

Document Title	Standardization of Interior Light Application Interfaces
Document Owner	AUTOSAR GbR
Document Responsibility	AUTOSAR GbR
Document Version	1.0.1
Document Status	Final
Part of Release	2.1
Revision	0014

Document Change History			
Date	Version	Changed by	Change Description
24.01.2007	1.0.1	AUTOSAR Administration	<ul style="list-style-type: none">• “Advice for users” revised• “Revision Information” added• Legal disclaimer revised
05.12.2006	1.0.0	AUTOSAR Administration	Initial Release

Page left intentionally blank

Disclaimer

Any use of these specifications requires membership within the AUTOSAR Development Partnership or an agreement with the AUTOSAR Development Partnership. The AUTOSAR Development Partnership will not be liable for any use of these specifications.

Following the completion of the development of the AUTOSAR specifications commercial exploitation licenses will be made available to end users by way of written License Agreement only.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Copyright © 2004-2006 AUTOSAR Development Partnership. All rights reserved.

Advice to users of AUTOSAR Specification Documents:

AUTOSAR Specification Documents may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the Specification Documents for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such Specification Documents, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

Table of Content

1	Purpose of this Document	7
2	Prerequisites	8
2.1	Naming Convention	8
2.2	Related Documents	9
3	Interior Light – Decomposition Overview	10
3.1	Interior Light Decomposition	10
3.1.1	InteriorLightManager.....	10
3.1.2	HMI.....	10
3.1.3	Doorlocks.....	11
3.1.4	CentralLocking / Remote Key / Keyless Access	11
3.1.5	Exterior Light / Environment Sensing.....	12
3.1.6	Anti-Theft-System.....	12
3.1.7	Inter-Domain Interfaces	12
3.1.8	ProfileManager	12
3.1.9	Actuators	12
3.2	List of interior light sources – groups of light sources	13
3.3	Design Rationale	15
3.4	Known Defects.....	16
4	Interface Descriptions.....	17
4.1	Overview on Inter Domain Ports	17
4.1.1	Ports referencing StaOperationMode	17
4.1.2	Ports referencing StaCrash	17
4.1.3	Ports referencing StaEnergyMgmt.....	17
4.1.4	Ports referencing VehicleSpeed	17
4.1.5	Ports referencing BatteryVoltage	17
4.2	Overview on Intra Domain Ports	18
4.2.1	Ports referencing ReqIntLight	18
4.2.2	Ports referencing ReqBrightness	18
4.2.3	Ports referencing StaOutsideBrightness.....	18
4.2.4	Ports referencing CmdGlobalDimLight	18
4.2.5	Ports referencing DisExtLight	19
4.2.6	Ports referencing DisParkingLights.....	19
4.2.7	Ports referencing ReqComingHomeLeavingHome	19
4.2.8	Ports referencing StaCentralLocking	19
4.2.9	Ports referencing StaDoor	20
4.2.10	Ports referencing ReqAccessMechanicalKey.....	20
4.2.11	Ports referencing ReqAccessKeyless	20
4.2.12	Ports referencing ReqAccessRemoteKey	20
4.2.13	Ports referencing ReqCarFinder	20
4.2.14	Ports referencing ReqVisualAlarm	21
4.2.15	Ports referencing CmdBrightness	21
4.2.16	Ports referencing StaLightFault.....	21
4.2.17	Ports referencing StaProfilePerson	21
4.2.18	Ports referencing DiagBrightness.....	21
4.3	Referenced Intra Domain Interfaces	22

4.3.1	ReqIntLight	22
4.3.2	ReqBrightness	22
4.3.3	StaOutsideBrightness	23
4.3.4	CmdGlobalDimLight.....	23
4.3.5	DisExtLight	24
4.3.6	DisParkingLights.....	24
4.3.7	ReqComingHomeLeavingHome	24
4.3.8	StaCentralLocking	25
4.3.9	StaDoor	25
4.3.10	ReqAccessMechanicalKey.....	26
4.3.11	ReqAccessKeyless	26
4.3.12	ReqAccessRemoteKey	27
4.3.13	ReqCarFinder.....	27
4.3.14	ReqVisualAlarm	27
4.3.15	CmdBrightness.....	28
4.3.16	StaLightFault.....	29
4.3.17	StaProfilePerson	29
4.3.18	DiagBrightness.....	30
4.4	Type Declarations.....	31
4.4.1	t_onoff.....	31
4.4.2	t_promille	31
5	Specification of Ports.....	32
5.1	Interior Light Manager.....	32
5.1.1	Require Ports.....	32
5.1.2	Provide Ports	38
5.1.3	Ports to AUTOSAR-Services	39
5.2	Interior Light Actuator(s)	39
5.2.1	Require Ports.....	40
5.2.2	Provide Ports	40
5.2.3	Diagnostic Ports.....	41
5.2.4	Cal-Ports.....	43
5.3	Interior Light Sensor(s)	43
5.3.1	Require Ports.....	43
5.3.2	Provide Ports	43
5.3.3	Diagnostic Ports.....	44
5.3.4	Cal-Ports.....	46
5.4	Exterior Light.....	46
5.4.1	Require Ports.....	46
5.4.2	Provide Ports	46
5.5	Central Locking.....	47
5.5.1	Require Ports.....	47
5.5.2	Provide Ports	47
5.6	other	47
5.6.1	Require Ports.....	47
5.6.2	Provide Ports	47
6	APPENDIX- Interface Descriptions	49
6.1	Inter Domain Interfaces	49
6.1.1	StaOperationMode.....	49

6.1.2	StaCrash.....	51
6.1.3	StaEnergyMgmt.....	51
6.1.4	VehicleSpeed.....	53
6.1.5	BatteryVoltage	53

1 Purpose of this Document

This document describes a basic architecture of the interior light functionality.

It decomposes the interior light functionality into several sensor and actuator software components of the interior light functionality, the interior light master component and their related interfaces.

Section 3 provides the hierarchical decomposition of the interior light functionality divided in interior light manager, sensor and actuator components.

Section 4 deals with interfaces from and to the outside world of the interior light functionality. It is split up into Inter- and Intra- domain interfaces.

Section 5 describes require & provide ports.

This document is not intended to provide a full description of the interfaces mentioned, it is rather an overview and a guide where to find the interfaces.

The following topics are intentionally not covered in this document as they are out of scope at this time:

- personalization parameters and interfaces
- variant coding
- diagnostic coding
- Provided ports referencing Inter-Domain Interfaces
- Ports & interfaces of Sensor/Core/Actuator SW-C's to the BSW (except diagnostic ports & cal-ports).

2 Prerequisites

2.1 Naming Convention

Guidelines for the naming convention:

- Type safety, i.e. interface definitions reflect usage/classification. Implausible connections are more difficult to make unintentionally.
- Easy finding of names (e.g. data element name is just one of „status“, „command“, „request“, „display“)
- Keep names simple whenever possible. Extend names only if required (e.g. multiple data elements).

	Port	Interface	Data element
Single occurrence	Content	Operation + Content	Operation
Example	LowBeamInd ParkingLightsInd	DisExtLight DisParkingLights	display display

	Port	Interface	Data element
Multiple occurrence	Content + (ExtendedContent)	Operation + Content	Operation + Content
Example for multiple ports	[IntLight] Activation	CmdBrightness	command

Semantic of keywords (e.g. “operation”) in the interface/ data element names:

- **Cmd**_(command) do/activate something (e. g. from Master to Actuator)
- **Req**_(request) demand to do/activate something (e. g. from Sensor to Master)
- **Sta**_(status) get functional status information
- **Hmi** user request (e.g. from driver via switch, touch screen,...)
- **Dis**_(display) feedback status for driver information display
- **Err**_(failure) operative/defective failure feedback (from actuator to master)

The “content” consists of one or more self-explaining catchwords.

2.2 Related Documents

- [1] Technical Overview
https://svn.autosar.org/repos/10Releases/AUTOSAR_TechnicalOverview.pdf
- [2] Methodology
https://svn.autosar.org/repos/10Releases/AUTOSAR_Methodology.ppt
- [3] Specification of the Virtual Functional Bus
https://svn.autosar.org/repos/10Releases/AUTOSAR_VirtualFunctionBus.pdf

3 Interior Light – Decomposition Overview

The InteriorLight software component group controls the interior light functionality of a car. InteriorLight receives driver wishes, senses the environment and controls the behavior of the interior light sources of a car.

3.1 Interior Light Decomposition

Components related to InteriorLight are bundled in several Blocks.

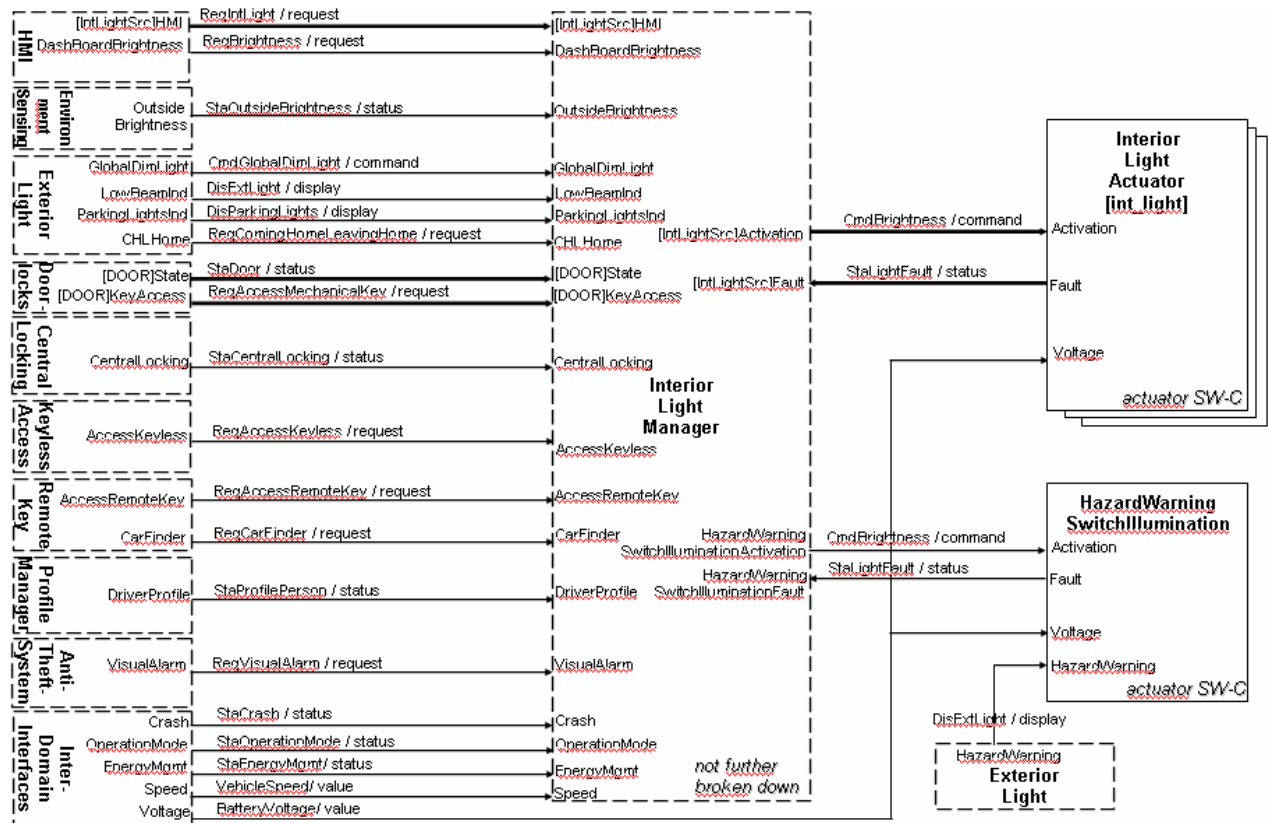


Fig 3.1-1 shows the decomposition of interior light architecture. Arrows indicate data flows.

Remark: This decomposition is used for explanation of standardized interfaces

3.1.1 InteriorLightManager

Inside InteriorLightManager the core functionality is implemented (not further broken down).

3.1.2 HMI

HMI contains all sensor components that detect the drivers demands related to interior light. Please note that “HMI” is used only for non-normative description.

For further decomposition see Fig 3.1-2, whereas usually several SW-C's InteriorLightSensor[int_light] exist.

3.1.3 Doorlocks

Out of Doorlocks the information is generated whether the corresponding door is open or closed as well as information on mechanical door lock(s). For further details see also separate documentation of Central Locking Control.

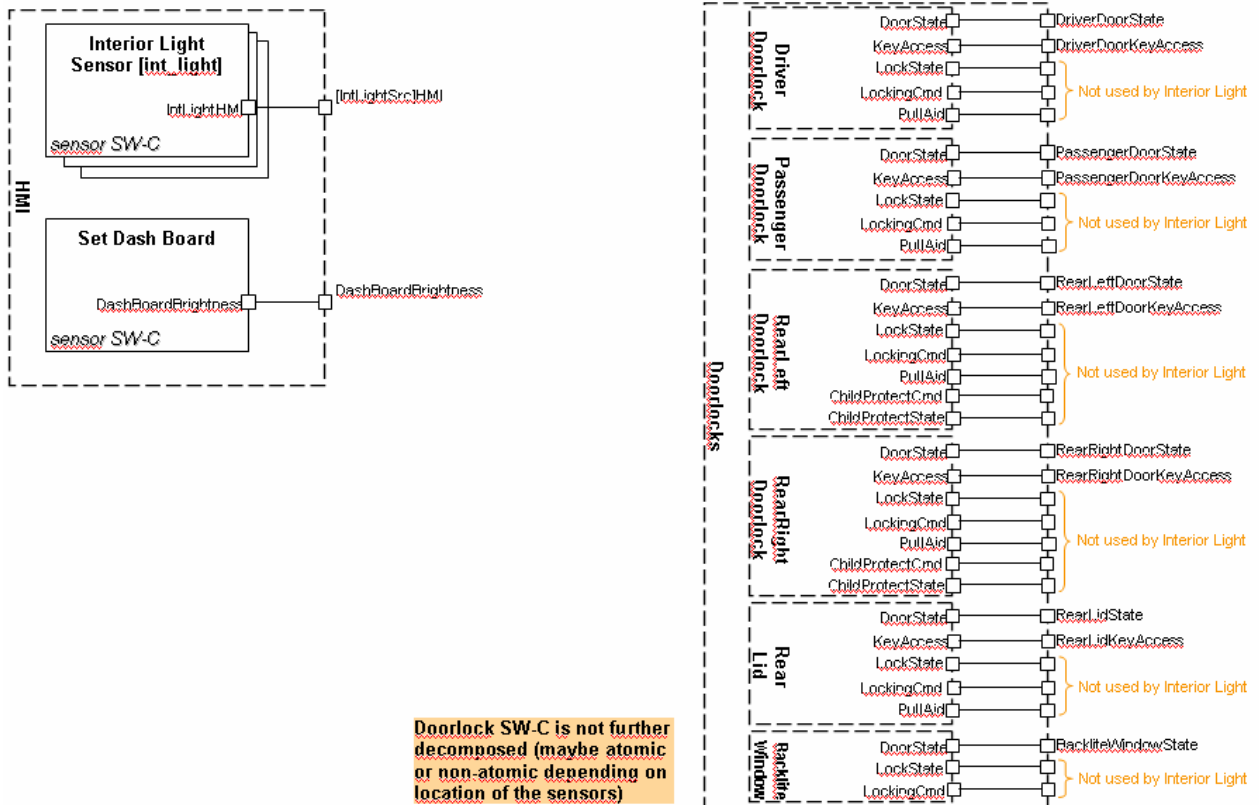


Fig 3.1-2 shows the further Decomposition of HMI & Doorlocks.

3.1.4 CentralLocking / Remote Key / Keyless Access

Driver requests to lock or unlock the car might also be provided by Remote Key or Keyless Access. Interior light uses only exterior requests (i.e. issued outside of the car) and not internal requests (such as emergency locking requests).

Note, that interior light may also use the actual door locking status for certain use cases (auto-relocking event starts interior light activity). Reflecting the lock state to the driver is not an intended feature of this standardized interior light.

Remote Key also delivers requests for “carfinder” features of the Interior Light.

For further details on the above mentioned functionalities see also separate documentation of Central Locking Control.

3.1.5 Exterior Light / Environment Sensing

Other blocks like `ExteriorLight` provide some information as well as `EnvironmentSensing` which provides information about the outside brightness.

For further details on the above mentioned functionalities see also separate documentation of Exterior Light.

3.1.6 Anti-Theft-System

One interfaces is provided also by this body and comfort functionalitiy. For further details on Anti-Theft-System see separate documentation.

3.1.7 Inter-Domain Interfaces

Through `Inter-Domain-Interfaces` inter-domain status information is exchanged.

3.1.8 ProfileManager

It is supposed only a single source coordinates profile Ids from certain personalization sources (remote key, keyless etc.). The `ProfileManager` is NOT in scope of this document.

3.1.9 Actuators

`InteriorLightActuator[int_light]` (including `HazardWarningSwitchIllumination`): All interior light actuators are described by the `LightSource[int_light]` components where `[int_light]` is a placeholder for individual light sources (e.g. `CabinLightFnCe` as Cabinlight “Front-Middle”) or a group of light sources (e.g. `reading_lights`). See next chapter for a list of possible effective names.

It might be decomposed into a light actuator type independent and a light actuator type dependent component, where light actuator types include conventional bulbs, LEDs or other decompositions, which allow the switching of lights on a more “generic” level.

The actuator also covers low level functionalities (e.g. measures to prevent flickering due to voltage variations) as well as over-/under voltage compensation.

Restrictions to Interior Light Actuator: Actuator for `IntLightGroup0..15` is allowed only if all light sources within the group are

- 1) not part of any other light group
- 2) not directly used by another light actuator

Same applies for groups like `CabinLights`, `TrunkCompartmentLights`, `FootwellLights` etc.

3.2 List of interior light sources – groups of light sources

The groups of light sources mentioned below (e.g. IntLightGroup0..15) may be configured by means of “Configuration-Bitfields” of type `t_LightGroupConfiguration` whereas each position corresponds to a certain interior light source (see typedef-documentation). This means that this information might be stored in NVRAM and accessible through calibration and/or diagnostic tools. Also possible is providing this information as static input to Interior Light Manager.

CabinLights

- CabinLightFnRi (front_right)
- CabinLightFnLe (front_left)
- CabinLightFnCe (front_center e.g. in the middle)
- CabinLightMidRi (middle_right)
- CabinLightMidLe (middle_left)
- CabinLightMidCe (middle_center)
- CabinLightReRi (rear_right)
- CabinLightReLe (rear_left)
- CabinLightReCe (rear_center)

TrunkCompartmentLights

- TrunkCompartmentLightCe (light_middle)
- TrunkCompartmentLightLoRi (lower_right)
- TrunkCompartmentLightLoLe (lower_left)
- TrunkCompartmentLightUpRi (upper_right)
- TrunkCompartmentLightUpLe (upper_left)

FootwellLights

- FootwellLightFnLe (front_left)
- FootwellLightFnCe (front_middle)
- FootwellLightFnRi (front_right)
- FootwellLightMidLe (middle_left)
- FootwellLightMidCe (middle_middle)
- FootwellLightMidRi (middle_right)
- FootwellLightReLe (rear_left)
- FootwellLightReCe (rear_middle)
- FootwellLightReRi (rear_right)

AshtrayLights

- AshtrayLightFnLe (front_left)
- AshtrayLightFnCe (front_middle)
- AshtrayLightFnRi (front_right)
- AshtrayLightMidLe (middle_left)
- AshtrayLightMidCe (middle_middle)
- AshtrayLightMidRi (middle_right)
- AshtrayLightReLe (rear_left)
- AshtrayLightReCe (rear_middle)
- AshtrayLightReRi (rear_right)

ReadingLights

- ReadingLightFnLe (front_left)
- ReadingLightFnCe (front_middle)
- ReadingLightFnRi (front_right)

ReadingLightReLe (rear_left)
ReadingLightReCe (rear_middle)
ReadingLightReRi (rear_right)
ReadingLight3rdRowLe (middle_left)
ReadingLight3rdRow Ce (middle_middle)
ReadingLight3rdRow Ri (middle_right)

CurbLights

CurbLightFnLe (front_left)
CurbLightFnRi (front_right)
CurbLightMidLe (middle_left)
CurbLightMidRi (middle_right)
CurbLightReLe (rear_left)
CurbLightReRi (rear_right)

VanityLights

VanityLightFnLe (front_left)
VanityLightFnRi (front_right)
VanityLightMidLe (middle_left)
VanityLightMidRi (middle_right)
VanityLightReLe (rear_left)
VanityLightReRi (rear_right)

CigaretteLighterIlluminationLights

CigaretteLighterIlluminationLightFnLe (front_left)
CigaretteLighterIlluminationLightFnCe (front_middle)
CigaretteLighterIlluminationLightFnRi (front_right)
CigaretteLighterIlluminationLightMidLe (middle_left)
CigaretteLighterIlluminationLightMidCe (middle_middle)
CigaretteLighterIlluminationLightMidRi (middle_right)
CigaretteLighterIlluminationLightReLe (rear_left)
CigaretteLighterIlluminationLightReCe (rear_middle)
CigaretteLighterIlluminationLightReRi (rear_right)
CigaretteLighterIlluminationLightTrLe (trunk_left)
CigaretteLighterIlluminationLightTrRi (trunk_right)

DoorSillIlluminationLights

DoorSillIlluminationLightFnLe (front_left)
DoorSillIlluminationLightFnRi (front_right)
DoorSillIlluminationLightMidLe (middle_left)
DoorSillIlluminationLightMidRi (middle_right)
DoorSillIlluminationLightReLe (rear_left)
DoorSillIlluminationLightReRi (rear_right)

DoorOpenerIlluminationLights

DoorOpenerIlluminationLightFnLe (front_left)
DoorOpenerIlluminationLightFnRi (front_right)
DoorOpenerIlluminationLightMidLe (middle_left)
DoorOpenerIlluminationLightMidRi (middle_right)
DoorOpenerIlluminationLightReLe (rear_left)
DoorOpenerIlluminationLightReRi (rear_right)

SafetyBeltLockIlluminationLights

SafetyBeltLockIlluminationLightFnLe (front_left)

SafetyBeltLockIlluminationLightFnCe (front_middle)
SafetyBeltLockIlluminationLightFnRi (front_right)
SafetyBeltLockIlluminationLightMidLe (middle_left)
SafetyBeltLockIlluminationLightMidCe (middle_middle)
SafetyBeltLockIlluminationLightMidRi (middle_right)
SafetyBeltLockIlluminationLightReLe (rear_left)
SafetyBeltLockIlluminationLightReCe (rear_middle)
SafetyBeltLockIlluminationLightReRi (rear_right)

GloveCompartmentLight
EngineCompartmentLight
IgnitionSwitchIllumination
HazardWarningSwitchIllumination
ExteriorLightSwitchIllumination

These groups of interior light interfaces are defined (virtual interior light interfaces):

IntLightGroup0,
IntLightGroup1,
...
IntLightGroup15,

Each group can consist out of any number of above mentioned interior light sources. This is configurable per implementation.

Examples are:

SwitchIlluminationGroupAccessory
SwitchIlluminationGroupIgnitionOn
SwitchIlluminationGroupExteriorLightOn
DoorAjarFrontLeftGroup
DoorAjarFrontRightGroup
DoorAjarRearLeftGroup
DoorAjarRearRightGroup

As a further example for the application of an "IntLightGroup" there is the „Battery-Saver Feature“ for deactivation of inadvertent loads of interior light (example glove box: lamp(s) might be hardwired activated by a switch. If left open unintentionally a separate actuator could be provided for deactivation of all these hardwired lamps after a certain timeout after ignition off).

3.3 Design Rationale

This design standard is the AUTOSAR-architecture (VFB-view) for interior lights. It provides a decomposition into SW-Cs and a list of standardized interfaces related to Interior Lights.

This decomposition is limited in granularity to sensor components, adapter components, the core functionality and actuator components.

The intent is for the decomposition to get not to atomic SWCs but to "purchasable" SW-Cs (as long as these are not sensor/actuator SW-Cs they are realized as AUTOSAR compositions for formal reasons). These SW-Cs will be obtained as a unit so that all the internal (and hence not AUTOSAR standardized) interfaces are con-

trolled by a single vendor, even if there are SW-Cs within the bought unit that reside on different ECUs. All interfaces between SW-Cs from different vendors should be standardized.

In addition, all ports are described, showing their AUTOSAR data qualities. Invalidation (where needed) is defined as in-band invalidation. Init values are specified where appropriate e.g. “off”, “idle”, “undefined”, “unknown” ... Where standardization is inappropriate, recommendations may be provided.

3.4 Known Defects

Data type definitions that are only used for one data element are currently not defined as a type but inserted directly into the data element description table. In future releases, these types will be declared explicitly.

4 Interface Descriptions

This chapter gives an overview on standardized ports and referenced interfaces. The intention for the subsequent description is to document common use cases for a better understanding of the standardized interfaces.

4.1 Overview on Inter Domain Ports

Please note that the referenced Inter Domain Interfaces are subject to change (for details see chapter 6.1)

4.1.1 Ports referencing StaOperationMode

<i>Used By</i>	<i>Reason, comment</i>
Interior Light Manager OperationMode	Activation/deactivation of interior light(s) depends on the status of certain operational modes. (Use case: switch off with ignition on)

4.1.2 Ports referencing StaCrash

<i>Used By</i>	<i>Reason, comment</i>
Interior Light Manager Crash	Activation of interior light(s) in case of an accident

4.1.3 Ports referencing StaEnergyMgmt

<i>Used By</i>	<i>Reason, comment</i>
Interior Light Manager EnergyMgmt	Restricted usage of interior lighting in case of low battery capacity

4.1.4 Ports referencing VehicleSpeed

<i>Used By</i>	<i>Reason, comment</i>
Interior Light Manager Speed	use case: switch off certain int.light sources above certain speed.

4.1.5 Ports referencing BatteryVoltage

<i>Used By</i>	<i>Reason, comment</i>
Interior Light Actuator Voltage	use case: over-/under voltage compensation
HazardWarningSwitchIllumination Voltage	

4.2 Overview on Intra Domain Ports

4.2.1 Ports referencing ReqIntLight

Used By	Reason, comment
Interior Light Manager [IntLightSrc]HMI (*)	user request for (de)activation of an interior light actuator or a virtual interior light group (switch on/off or automatic mode)
HMI [IntLightSrc]HMI (*)	

(*) whereas [IntLightSrc] specifies the requested light source/ group of light sources
 Comment: this interface is specific to interior lighting. User requests within exterior lighting are of type switch on/off.

4.2.2 Ports referencing ReqBrightness

Used By	Reason, comment
Interior Light Manager DashBoardBrightness	Sets the illumination brightness of the instrument cluster. Most often the driver interface is a potentiometer.
HMI DashBoardBrightness	

4.2.3 Ports referencing StaOutsideBrightness

Used By	Reason, comment
Interior Light Manager OutsideBrightness	Activation or brightness of the interior lights may depend on the outside brightness
Environment Sensing OutsideBrightness	

Comment: this interface is specific to interior/exterior lighting.

4.2.4 Ports referencing CmdGlobalDimLight

Used By	Reason, comment
Interior Light Manager GlobalDimLight	This interface gives the information of the actual outside brightness. E.g. exterior or interior light can use this information to adapt their own dimming calculation. HINT: might be calculated from StaOutsideBrightness in order to allow smooth transitions (e.g. ramp or visual effects of exterior or interior light) while switching e.g. from "night" to "twilight_1".
Exterior Light GlobalDimLight	

Comment: this interface is specific to interior/exterior lighting.

4.2.5 Ports referencing DisExtLight

Used By	Reason, comment
Interior Light Manager LowBeamInd	Activation of the interior lights may depend on the status of the low beam. (Use case: after switching ignition off, activate the interior lights only if the low beams currently or a short time before are switched on)
Exterior light LowBeamInd	

Used By	Reason, comment
HazardWarningSwitchIllumination HazardWarning	use case: illumination of hazard switch is often used for functional indication too.
Exterior light HazardWarning	

4.2.6 Ports referencing DisParkingLights

Used By	Reason, comment
Interior Light Manager ParkingLightsInd	Activation of the interior lights may depend on the status of the parking lights. (Use case: Instrument cluster is illuminated when parking lights are switched on by driver.)
Exterior light ParkingLightsInd	

Comment: this interface is specific to interior/exterior lighting. (off/left/right/both)

4.2.7 Ports referencing ReqComingHomeLeavingHome

Used By	Reason, comment
Interior Light Manager CHLHome	Activate interior light(s) in case of “coming/leaving home” request. This seems at a first glance redundant information, as also ReqAccess_* is available. Using also ReqChlh allows synchronization with exterior light to illuminate the car internally and externally at the same time. Additionally it allows coordinated reactions to the outside brightness (i.e. should avoid that CHLH functionality starts, but interior light does not start due to different activation levels of the outside brightness.
Exterior light CHLHome	

Comment: this interface is specific to interior/exterior lighting. (off/coming_home/leaving_home)

4.2.8 Ports referencing StaCentralLocking

Used By	Reason, comment
Interior Light Manager CentralLocking	use case: automatic re-lock of car (after user has –maybe unintentionally– unlocked the car but not opened any door)
Central Locking CentralLocking	

4.2.9 Ports referencing StaDoor

Used By	Reason, comment
Interior Light Manager [Door]State (*)	Activation of the interior lights if one of the doors was opened
Doorlocks [Door]State (*)	

(*) whereas [door] is one out of: DriverDoor, PassengerDoor, RearLeftDoor, Rear-RightDoor, RearLid, BackliteWindow

4.2.10 Ports referencing ReqAccessMechanicalKey

Used By	Reason, comment
Interior Light Manager [Door]KeyAccess (*)	Activation or deactivation of interior light if car is locked or unlocked by mechanical driver/passenger door lock. Mechanical key can also request comfort close and comfort open. The request for comfort open may not always be allowed, depending on the security status of the vehicle.
Doorlocks [Door]KeyAccess (*)	

(*) whereas [door] is one out of: DriverDoor, PassengerDoor, RearLeftDoor, Rear-RightDoor, RearLid: Note regarding rear doors: ReqAccessKeyDoor might be applicable for Sliding Doors too!

4.2.11 Ports referencing ReqAccessKeyless

Used By	Reason, comment
Interior Light Manager AccessKeyless	Activation or deactivation of interior light if car is locked or unlocked by keyless-entry system.
KeylessAccess AccessKeyless	

Comment: this interface is specific to interior lighting/ central locking.

4.2.12 Ports referencing ReqAccessRemoteKey

Used By	Reason, comment
Interior Light Manager AccessRemoteKey	Activation or deactivation of interior light if car is locked or unlocked by remote key.
RemoteKey AccessRemoteKey	

Comment: this interface is specific to interior lighting/ central locking.

4.2.13 Ports referencing ReqCarFinder

Used By	Reason, comment
Interior Light Manager CarFinder	Activation of the interior light(s) if the user presses a certain button on the remote key in order to easily find his car at night
RemoteKey CarFinder	

4.2.14 Ports referencing ReqVisualAlarm

Used By	Reason, comment
Interior Light Manager VisualAlarm	Activation of the interior lights in case of theft alarm
AntiTheftSystem VisualAlarm	

4.2.15 Ports referencing CmdBrightness

Used By	Reason, comment
Interior Light Manager [IntLightSrc]Activation (*)	Command to activate (with the specified brightness) or deactivate [IntLightSrc]. High resolution required (less than 1%).
Interior light Actuator(s) Activation	
HazardWarningSwitchIllumination Activation	

(*) whereas [IntLightSrc] specifies the requested light source/ group of light sources

4.2.16 Ports referencing StaLightFault

Used By	Reason, comment
Interior Light Manager [IntLightSrc]Fault (*)	Operational/defective feedback of the light actuators
Interior light Actuator(s) Fault	
HazardWarningSwitchIllumination Fault	

(*) whereas [IntLightSrc] specifies the requested light source/ group of light sources

4.2.17 Ports referencing StaProfilePerson

Used By	Reason, comment
Interior Light Manager DriverProfile	personalization profile detected by profile manager.
Profile_manager DriverProfile	

Comment: this interface is specific to personalization profile.

4.2.18 Ports referencing DiagBrightness

Used By	Reason, comment
Interior light Actuator(s) Activation	Command to activate (with the specified brightness) or release [IntLightSrc] for diagnostic purposes

(*) whereas [IntLightSrc] specifies the requested light source/ group of light sources

4.3 Referenced Intra Domain Interfaces

This chapter gives an overview on the referenced intra domain interfaces. For inter domain interfaces see APPENDIX.

4.3.1 ReqIntLight

<i>Interface</i>			
Name (of Interface)	ReqIntLight		
Communcation Method	SenderReceiver		
Description	Request to activate / deactivate an interior light actuator denoted by [int_light] or a virtual interior light group also denoted by [int_light]. Faulty switch requires (in-band) invalidation. Also there might be a separate value be required for initialization.		
isService	No (e.g. no BSW service)		
1 Data Element Name	request		
Data Type	t_ReqIntLight of type enum		
Symbolic Constants	Label	Value	Comment
	unknown	0	--
	off	1	Always off
	on	2	Always on
	auto	3	Automatically on e. g. by opening a door
	invalid	4	--
Data/Event	Data		
Remarks	Length (bit) = 2 (see data type) Recommendations: Init value=unknown, can invalidate=Yes This interface is not necessary for each interior light actuator, it is not intended for hardwired interior lights.		

4.3.2 ReqBrightness

<i>Interface</i>			
Name (of Interface)	ReqBrightness		
Communcation Method	SenderReceiver		
Description	Request the brightness of a light source or the control panel. The percentage value (independent of light actuator type and environment e.g. voltage of electrical system) depends on the functionality that the light source shall be switched on for. 100 percent means full power of the light actuator. Invalid is chosen for emergency illumination in case of jammed network.		
isService	No (e.g. no BSW service)		
1 Data Element Name	request		
Data Type	t_promille		
Physical Interpretation	Offset	0	
	Unit	0,1%	
Physical Values	Min Value	0	
	Max Value	1000	
	Resolution	1	
Data/Event	Data		
Remarks	Length (bit) = 10 (see data type) Recommendations: Init value=0, can invalidate=Yes		

4.3.3 StaOutsideBrightness

Interface			
Name (of Interface)	StaOutsideBrightness		
Communication Method	SenderReceiver		
Description	This is the outside brightness quantizes in some levels. It is e. g. used from exterior light master to generate a CmdGlobalDim value		
isService	No (e.g. no BSW service)		
1 Data Element Name	status		
Data Type	t_StaOutsideBrightness of type enum		
Symbolic Constants	Label	Value	Comment
	unknown	0	e.g. (temporary) no sensor information available
	night	1	--
	Twilight_1	2	--
	Twilight_2	3	--
	Twilight_3	4	--
	Twilight_4	5	--
	day	6	--
	invalid	7	e.g. sensor signalizes defect
Data/Event	Data		
Remarks	Length (bit) = 3 (see data type) Recommendations: Init value= unknown, can invalidate=Yes These values need to be calibrated against light levels for each car.		

4.3.4 CmdGlobalDimLight

Interface			
Name (of Interface)	CmdGlobalDimLight		
Communication Method	SenderReceiver		
Description	This is the command of light master to give a global exterior and interior light dimming information based on the outside brightness level. It is used e. g. to dim the exterior light depending on outside brightness. 0% means full darkness 100% means maximum brightness This interface gives the information of the actual outside brightness. E.g. exterior or interior light can use this information to adapt their own dimming calculation.		
isService	No (e.g. no BSW service)		
1 Data Element Name	command		
Data Type	t_promille		
Physical Interpretation	Offset	0	
	Unit	0,1%	
Physical Values	Min Value	0	
	Max Value	1000	
	Resolution	1	
Data/Event	Data		
Remarks	Length (bit) = 7 (see data type) Recommendations: Init value=0, can invalidate=Yes		

4.3.5 DisExtLight

Interface	
Name (of Interface)	DisExtLight
Communcation Method	SenderReceiver
Description	Provides the information whether a certain function is on or off. As long as the function is on (activated), the value on should be provided. This interface can be used e.g. to provide feedback to the driver via MMI. This is considered as a status signal, which carries continuously a certain value as long as its semantic condition is present.
isService	No (e.g. no BSW service)
1 Data Element Name	display
Data Type	t_onoff
Data/Event	Data
Remarks	Length (bit) = 1 (see data type) Recommendations: Init value=off , can invalidate=No

4.3.6 DisParkingLights

Interface																
Name (of Interface)	DisParkingLights															
Communcation Method	SenderReceiver															
Description	The interface is provided by a master component and provides feedback to e.g. the MMI whether the left parking lights, the right parking lights or the positioning lights are to be displayed as active. As long as this functionality is required the value should be on.															
isService	No (e.g. no BSW service)															
1 Data Element Name	display															
Data Type	t_DisParkingLights of type enum															
Symbolic Constants	<table border="1"> <thead> <tr> <th>Label</th> <th>Value</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>off</td> <td>0</td> <td>--</td> </tr> <tr> <td>left</td> <td>1</td> <td>--</td> </tr> <tr> <td>right</td> <td>2</td> <td>--</td> </tr> <tr> <td>both</td> <td>3</td> <td>Positioning light</td> </tr> </tbody> </table>	Label	Value	Comment	off	0	--	left	1	--	right	2	--	both	3	Positioning light
Label	Value	Comment														
off	0	--														
left	1	--														
right	2	--														
both	3	Positioning light														
Data/Event	Data															
Remarks	Length (bit) = 2 (see data type) Recommendations: Init value=off , can invalidate=No															

4.3.7 ReqComingHomeLeavingHome

Interface										
Name (of Interface)	ReqComingHomeLeavingHome									
Communcation Method	SenderReceiver									
Description	Demand to activate/ deactivate coming home/ leaving home functionalities. The request can be set e.g. manually by the user or via user settings in the MMI.									
isService	No (e.g. no BSW service)									
1 Data Element Name	request									
Data Type	t_ReqComingHomeLeavingHome of type enum									
Symbolic Constants	<table border="1"> <thead> <tr> <th>Label</th> <th>Value</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>off</td> <td>0</td> <td>--</td> </tr> <tr> <td>coming_home</td> <td>1</td> <td>--</td> </tr> </tbody> </table>	Label	Value	Comment	off	0	--	coming_home	1	--
Label	Value	Comment								
off	0	--								
coming_home	1	--								

	leaving_home	2	--
Data/Event	Data		
Remarks	Length (bit) = 2 (see data type) Recommendations: Init value=off , can invalidate=No		

4.3.8 StaCentralLocking

Interface				
Name (of Interface)	StaCentralLocking			
Communcation Method	SenderReceiver			
Description	Provides the information which status the central door locking currently has.			
isService	No (e.g. no BSW service)			
1 Data Element Name	status			
Data Type	record			
Record elements	Element	Label	Value	Comment
	locking_state (of Type enum)	undefined	0	undefined status
		open	1	At least one door or lid is open
		closed	2	all doors including lids closed, (at least one door unlocked)
		locked	3	all doors including lids locked, (at least one door unsafed)
		safed	4	Complete car safed (double locked)
	Element	Label	Value	Comment
	trigger_source (of Type enum)	no_trigger_source	0	--
		trigger_source1	1	User defined source1
		trigger_source2	2	User defined source2
...		...	--	
trigger_source31		31	User defined source31	
Data/Event	Data			
Remarks	Length (bit) = 3 (see data type) Recommendations: Init value= {undefined, no_trigger_source}, can invalidate=No The status undefined is valid if there are no feedback switches in the latch and the ECU lost the information of actual status			

4.3.9 StaDoor

Interface			
Name (of Interface)	StaDoor		
Communcation Method	SenderReceiver		
Description	Provides information whether a door or a lid is open or closed. As long as the status is valid, the value should be provided.		
isService	No (e.g. no BSW service)		
1 Data Element Name	status		
Data Type	t_StaDoor of type enum		
Symbolic Constants	Label	Value	Comment
	unknown	0	Initialisation phase
	open	1	--

	closed	2	--
	ajar	3	nearly closed
	invalid	4	implausible
Data/Event	Data		
Remarks	Length (bit) = 3 Recommendations: Init value=unknown, can invalidate =Yes		

4.3.10 ReqAccessMechanicalKey

Interface			
Name (of Interface)	ReqAccessMechanicalKey		
Communication Method	SenderReceiver		
Description	This interface requests unlock, lock, comfort close/ open from the mechanical key in the key cylinder. The values for comfort_open and for comfort_close are included intentionally, these states could be reached e.g. when the key is turned into the open position (or, vice versa the close position) for a longer period of time.		
isService	No (e.g. no BSW service)		
1 Data Element Name	request		
Data Type	t_ReqAccessMechanicalKey of type enum		
Symbolic Constants	Label	Value	Comment
	idle	0	no access request
	unlock	1	--
	lock	2	--
	comfort_open	3	--
	comfort_close	4	--
Data/Event	Data		
Remarks	Length (bit) = 3 Recommendations: Init value=idle, can invalidate=No		

4.3.11 ReqAccessKeyless

Interface			
Name (of Interface)	ReqAccessKeyless		
Communication Method	SenderReceiver		
Description	Access request (from detected source) to lock / unlock the latch from keyless entry system.		
isService	No (e.g. no BSW service)		
1 Data Element Name	request		
Data Type	record		
Record elements	Element	Type	Comment
	detection_source (of type enum)	{undefined, driver_door, passenger_door, right_rear_door, left_rear_door, rear_lid}	Provides information about the location of the person (with valid key) approaching the vehicle
	requested_action (of type enum)	{Idle, lock, unlock, safe, comfort_open, comfort_close}	Provides information about the requested action from the corresponding position (detection source)

Data/Event	Data
Remarks	Length (bit) = 3+3 Recommendations: Init value={undefined; idle}, can invalidate=No There is no request for "selected_unlock" possible because this feature is implemented by means of detection source and system configuration (e.g. personalization).

4.3.12 ReqAccessRemoteKey

Interface			
Name (of Interface)	ReqAccessRemoteKey		
Communication Method	SenderReceiver		
Description	Access request generated e.g. by remote key receiver unit. The content is derived from the user request by remote key.		
isService	No (e.g. no BSW service)		
1 Data Element Name	request		
Data Type	t_ReqAccessRemoteKey of type enum		
Symbolic Constants	Label	Value	Comment
	idle	0	no user request
	unlock	1	--
	selected_unlock	2	e. g. one door unlock
	lock	3	--
	safe	4	double_lock
	Comfort_open	5	e. g. open all windows
	Comfort_close	6	--
Data/Event	Data		
Remarks	Length (bit) = 3 Recommendations: Init value=idle, can invalidate=No There is no request for "unsafe" included here because "unsafe" is not considered as an exclusive / individual state.		

4.3.13 ReqCarFinder

Interface	
Name (of Interface)	ReqCarFinder
Communication Method	SenderReceiver
Description	Demand to activate/ deactivate carfinder functionality. The request can be set e.g. manually by the user or via user settings in the MMI.
isService	No (e.g. no BSW service)
1 Data Element Name	request
Data Type	t_onoff
Data/Event	Data
Remarks	Length (bit) = 1 (see data type) Recommendations: Init value=off , can invalidate=No

4.3.14 ReqVisualAlarm

Interface	
Name (of Interface)	ReqVisualAlarm
Communication Method	SenderReceiver
Description	Request from anti theft system master to e.g. blink master to start requested blink sequence.

isService	No (e.g. no BSW service)		
1 Data Element Name	request		
Data Type	t_ReqVisualAlarm of type enum		
Symbolic Constants	Label	Value	Comment
	no_alarm	0	--
	alarm_type_1	1	--
	alarm_type_2	2	--
	alarm_type_3	3	--
	alarm_type_4	4	--
	alarm_type_5	5	--
	alarm_type_6	6	--
alarm_type_7	7	--	
Data/Event	Data		
Remarks	Length (bit) = 3 (see data type) Recommendations: Init value=no_alarm, can invalidate=No		

4.3.15 CmdBrightness

Interface				
Name (of Interface)	CmdBrightness			
Communcation Method	SenderReceiver			
Description	Request the brightness of a light source or the control panel. The percentage value (independent of light actuator type and environment e.g. voltage of electrical system) depends on the functionality that the light source shall be switched on for. 100 percent means full power of the light actuator. Invalidation (in-band) might be choosen for emergency illumination in case of jammed network.			
isService	No (e.g. no BSW service)			
1 Data Element Name	command			
Data Type	record			
Record elements	Element	Type	Comment	
			brightness	t_promille
			Physical Interpretation	
			Offset	0
			Unit	0.1 %
			Physical Values	
			Min Value	0
	Max Value	1000		
	Resolution	1		
	Element	Type	Label	Value
ramp	enum	no_ramp	0	
		ramp1	1	
		ramp2	2	
		
		ramp7	7	
Data/Event	Data			

Remarks	<p>Length (bit) = 13 (see data type) Recommendations: Init value={0;0}, can invalidate=Yes (Inband-invalidation for complete PortInterface is signaled by Invalid-value of brightness, e.g. if brightness is invalid then the complete record information has to be regarded as invalid).</p> <p>ramp defines a certain “changerate” with which the given percentage value shall be reached (no_ramp means the brightness value shall be effective immediately). A ramp might be chosen in order to avoid that the master component is executed at that rate (e.g. needs very high CPU load) to implement a dim curve. The actual ramp times are stored in NVRAM and should be tuneable via diagnostics.</p>
----------------	---

4.3.16 StaLightFault

Interface			
Name (of Interface)	StaLightFault		
Communcation Method	SenderReceiver		
Description	<p>Used as feedback from an actuator component to indicate that the light source is broken or not. Note that no details of the fault are transmitted, but just an operational / defective fault indication. This enables a control component to decide whether a broken light has to be replaced by another light source.</p> <p>Note: The interface Status[light] is not used for detailed diagnostics.</p> <p>If the light source contains more than one e. g. bulb, it shall be defined how many actuators have to be broken to reach the state defective. The decision to give the status defective is done by the actuator component.</p>		
isService	No (e.g. no BSW service)		
1 Data Element Name	status		
Data Type	t_StaLightFault of type enum		
Symbolic Constants	Label	Value	Comment
	operational	0	--
	defective	1	e. g. bulb, LED array broken
Data/Event	Data		
Remarks	<p>Length (bit) = 1 Recommendations: Init value=operational, can invalidate=No</p>		

4.3.17 StaProfilePerson

Interface			
Name (of Interface)	StaProfilePerson		
Communcation Method	SenderReceiver		
Description	<p>Status of current selected personalization profile from profile manager. It is a common profile selectable from transponder, remote key, keyless access, HMI,...)</p>		
isService	No (e.g. no BSW service)		
1 Data Element Name	id		
Data Type	t_StaProfilePerson of type enum		
Symbolic Constants	Label	Value	Comment
	no_profile	0	no defined profile
	profile1	1	--
	profile2	2	--

Data/Event	profile3	3	--
	--
	--
	profile14	14	--
	profile15	15	--
Remarks	Data Length (bit) = 4 Recommendations: Init value=no_profile, can invalidate=No		

4.3.18 DiagBrightness

Interface				
Name (of Interface)	DiagBrightness			
Communication Method	SenderReceiver			
Description	Request the brightness of a light source for diagnostic purposes. The percentage value (independent of light actuator type and environment e.g. voltage of electrical system) depends on the functionality that the light source shall be switched on for. 100 percent means full power of the light actuator.			
isService	No (e.g. no BSW service)			
1 Data Element Name	command			
Data Type	record			
Record elements	Element	Type	Comment	
	brightness	t_promille	Physical Interpretation	
			Offset	0
			Unit	0.1 %
			Physical Values	
			Min Value	0
			Max Value	1000
		Resolution	1	
	Element	Type	Label	comment
	diagnostic	t_onoff	off	diagnostic inactive
on			diagnostic active	
Data/Event	Data			
Remarks	Length (bit) = 8 (see data type) Recommendations: Init value={0;off}, can invalidate=No			

4.4 Type Declarations

4.4.1 t_onoff

Name	t_onoff		
Lower-Limit	0		
Upper-Limit	1		
Description	The type t_onoff is considered as a status signal, which carries continuously a certain value as long as its semantic condition is present.		
Symbolic Constants	Element Name	Value	Description
	off	0	--
	on	1	--

4.4.2 t_promille

Name	t_promille	
Base Type	unsigned integer	
Invalid Value	1001	
Description	The type t_promille represents a promille value.	
Physical Interpretation	Offset	0
	Factor	1
	Unit	Promille
Physical Values	Min Value	0
	Max Value	1000
	Resolution	1

5 Specification of Ports

5.1 Interior Light Manager

Interior Light Manager as Core SW-C provides the application functionality as well as failure handling e.g. intelligent generation of replacement value. Please note that system restrictions should not be mixed up with application functionality. There might be a separate SW-C handling system restrictions like Actuator-Adapter-SW-C.

5.1.1 Require Ports

Port Name	CabinLightFnCeHMI			
Reference to interface	ReqIntLight			
Direction	required (by SW-C InteriorLightManager)			
Description	User request to activate / deactivate cabin_light_front_middle			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	unknown	yes	---

Please note that this port is representative for many other ports **Req[IntLightSrc]** used to request to activate / deactivate an interior light actuator denoted by [IntLightSrc] or a virtual interior light group also denoted by [IntLightSrc]. This interface is not necessary for each interior light actuator, it is not intended for hardwired interior lights.

Port Name	DashBoardBrightness			
Reference to interface	ReqBrightness			
Direction	required (by SW-C InteriorLightManager)			
Description	Interior Light Manager requires information how to set dashboard brightness			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	0	yes	---

Port Name	OutsideBrightness			
Reference to interface	StaOutsideBrightness			
Direction	required (by SW-C InteriorLightManager)			
Description	Outside brightness quantized in some levels. These values need to be calibrated against light levels for each car. Activation or brightness of the interior lights may depend on the outside brightness.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	unknown	yes	---

Port Name	GlobalDimLight			
Reference to interface	CmdGlobalDimLight			
Direction	required (by SW-C InteriorLightManager)			
Description	<p>Command of exterior light master to give a global exterior and interior light dimming information based on the outside brightness level. It is used e. g. to dim the exterior/interior light depending on outside brightness by means of smooth transitions (e.g. ramp resulting from switching from "night" to "twilight_1")</p> <p>0% means full darkness 100% means maximum brightness.</p>			
Data Elements	Name	Init Value	can invalidate	Further attributes
	command	0	yes	---

Port Name	LowBeamInd			
Reference to interface	DisExtLight			
Direction	required (by SW-C InteriorLightManager)			
Description	<p>Information provided by exterior light as feedback to e.g. the MMI whether the lowbeam lights are to be displayed as active. Used by InteriorLightManager as activation of the interior lights may depend on the status of the low beam. (Use case: after switching ignition off, activate the interior lights only if the low beams currently or a short time before are switched on)</p>			
Data Elements	Name	Init Value	can invalidate	Further attributes
	display	off	no	---

Port Name	ParkingLightsInd			
Reference to interface	DisParkingLights			
Direction	required (by SW-C InteriorLightManager)			
Description	<p>Information provided by exterior light as feedback to e.g. the MMI whether the left parking lights are to be displayed as active. Used by InteriorLightManager as activation of the interior lights may depend on the status of the parking lights. (Use case: Instrument cluster is illuminated when parking lights are switched on by driver.)</p>			
Data Elements	Name	Init Value	can invalidate	Further attributes
	display	off	no	---

Port Name	CHLHome			
Reference to interface	ReqComingHomeLeavingHome			
Direction	required (by SW-C InteriorLightManager)			
Description	<p>Demand to activate/ deactivate coming home/ leaving home functionalities. Interior light(s) might be activated in case of "coming/leaving home" request.</p>			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	off	no	---

Port Name	DriverDoorState			
Reference to interface	StaDoor			
Direction	required (by SW-C InteriorLightManager)			
Description	Information whether the driver door is open or closed.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	unknown	yes	---

Port Name	PassengerDoorState			
Reference to interface	StaDoor			
Direction	required (by SW-C InteriorLightManager)			
Description	Information whether the passenger door is open or closed.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	unknown	yes	---

Port Name	RearLeftDoorState			
Reference to interface	StaDoor			
Direction	required (by SW-C InteriorLightManager)			
Description	Information whether the rear door is open or closed.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	unknown	yes	---

Port Name	RearRightDoorState			
Reference to interface	StaDoor			
Direction	required (by SW-C InteriorLightManager)			
Description	Information whether the rear door is open or closed.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	unknown	yes	---

Port Name	RearLidState			
Reference to interface	StaDoor			
Direction	required (by SW-C InteriorLightManager)			
Description	Information whether the rear lid is open or closed.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	unknown	yes	---

Port Name	BackliteWindowState			
Reference to interface	StaDoor			
Direction	required (by SW-C InteriorLightManager)			
Description	information whether the backlite window is open or closed.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	unknown	yes	---

Port Name	CentralLocking			
Reference to interface	StaCentralLocking			
Direction	required (by SW-C InteriorLightManager)			
Description	automatic re-lock of car may require interior light activity			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	undefined	no	---

Port Name	DriverDoorKeyAccess			
Reference to interface	ReqAccessMechanicalKey			
Direction	required (by SW-C InteriorLightManager)			
Description	Requests unlock, lock, comfort close/ open from the mechanical key in the key cylinder of driver door. Unlocking and locking of doors can trigger interior light activities.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	idle	no	---

Port Name	PassengerDoorKeyAccess			
Reference to interface	ReqAccessMechanicalKey			
Direction	required (by SW-C InteriorLightManager)			
Description	Requests unlock, lock, comfort close/ open from the mechanical key in the key cylinder of passenger door. Unlocking and locking of doors can trigger interior light activities.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	idle	no	---

Port Name	RearLeftDoorKeyAccess			
Reference to interface	ReqAccessMechanicalKey			
Direction	required (by SW-C InteriorLightManager)			
Description	Requests unlock, lock, comfort close/ open from the mechanical key in the key cylinder of rear left sliding door. Unlocking and locking of doors can trigger interior light activities.			
Data Elements	Name	Init Value	can in-	Further attributes

			validate	
	request	idle	no	---

Port Name	RearRightDoorKeyAccess			
Reference to interface	ReqAccessMechanicalKey			
Direction	required (by SW-C InteriorLightManager)			
Description	Requests unlock, lock, comfort close/ open from the mechanical key in the key cylinder of rear right sliding door. Unlocking and locking of doors can trigger interior light activities.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	idle	no	---

Port Name	RearLidKeyAccess			
Reference to interface	ReqAccessMechanicalKey			
Direction	required (by SW-C InteriorLightManager)			
Description	Requests unlock, lock, comfort close/ open from the mechanical key in the key cylinder of rear lid. Unlocking and locking can trigger interior light activities.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	idle	no	---

Port Name	AccessKeyless			
Reference to interface	ReqAccessKeyless			
Direction	required (by SW-C InteriorLightManager)			
Description	Access request (from detected source) to lock / unlock the latch from keyless entry system may trigger interior light activities.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	{undefined; idle}	no	---

Port Name	AccessRemoteKey			
Reference to interface	ReqAccessRemoteKey			
Direction	required (by SW-C InteriorLightManager)			
Description	Access request generated e.g. by remote key receiver unit may trigger interior light activities. The content is derived from the user request by remote key.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	idle	no	---

Port Name	CarFinder			
------------------	-----------	--	--	--

Reference to interface	ReqCarFinder			
Direction	required (by SW-C InteriorLightManager)			
Description	Demand to activate/ deactivate carfinder functionality. May activate interior light(s) in order to easily find the car at night.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	off	no	---

Port Name	DriverProfile			
Reference to interface	StaProfilePerson			
Direction	required (by SW-C InteriorLightManager)			
Description	Status of current selected personalization profile from profile manager. It is a common profile selectable from transponder, remote key, keyless access, HMI,...)			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	no_profile	no	---

Port Name	VisualAlarm			
Reference to interface	ReqVisualAlarm			
Direction	required (by SW-C InteriorLightManager)			
Description	Request alarm blinking e.g. in case of an unauthorized car access detected by the alarm system. May also trigger interior light activities.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	no_alarm	no	---

Port Name	Crash			
Reference to interface	StaCrash			
Direction	required (by SW-C InteriorLightManager)			
Description	The current crash status is required (values like pre-crash, post-crash, ...). May also trigger interior light activities.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	NoEvent	no	---

Port Name	OperationMode			
Reference to interface	StaOperationMode			
Direction	required (by SW-C InteriorLightManager)			
Description	The car operation state is required. It can be seen as an abstraction and extension of the normal ignition switch position (also called "clamp position") which is traditionally used to guide functional operation of the vehicle.			
Data Elements	Name	Init Value	can invalidate	Further attributes

	status	off	no	---
--	--------	-----	----	-----

Port Name	EnergyMgmt			
Reference to interface	StaEnergyMgmt			
Direction	required (by SW-C InteriorLightManager)			
Description	The information is required whether all, basic or only keep-alive components/functionalities are allowed to run. As long as the status is valid, the corresponding value as defined by the table below should be provided.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	full_funct	no	---

Port Name	Speed			
Reference to interface	VehicleSpeed			
Direction	required (by SW-C InteriorLightManager)			
Description	certain int.light sources are switched off above certain speed.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	value	0	no	---

Port Name	CabinLightFMFault			
Reference to interface	StaLightFault			
Direction	required (by SW-C InteriorLightManager)			
Description	InteriorLightManager checks operational / defective feedback from an actuator.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	operational	no	---

5.1.2 Provide Ports

Port Name	CabinLightFMActivation			
Reference to interface	CmdBrightness			
Direction	provided (by SW-C InteriorLightManager)			
Description	InteriorLightManager requests the brightness of cabin_light_front_middle. The percentage value (independent of light actuator type and environment e.g. voltage of electrical system) depends on the functionality that the light source shall be switched on for. 100 percent means full power of the light actuator.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	command	{0;0}	yes	---

IMPORTANT NOTE: Interior Light Manager has many additional provide ports `CmdBrightness[int_light]` corresponding to **CmdBrightnessCabinLightFM**. Thereby Interior Light Manager request the brightness of a light actuator denoted by `[int_light]`. `[int_light]` may be any out the list of interior light sources & groups of light sources.

5.1.3 Ports to AUTOSAR-Services

For ports to ECU State-Management as well as COM-Services see according AUTOSAR documentation.

Regarding Cal-Ports there are only implementation specific parameters. Please note that “pure ROM parameters” (like PWM frequency) are not listed here.

Not in scope of this document: Diagnostic events of other SW-C's than Sensor-/Actuator SW-C.

HINT - Relationship between Diagnostics and internal state of SW-Cs: The interaction of SW-Cs with the Diagnostic Communication Manager (DCM) takes place exclusively through the corresponding AUTOSAR DCM Service. AUTOSAR Services are accessible, like any other communication, through interfaces and ports and thus Communication Modes apply. Through the application of Communication Modes to determined RTEEvents (like receiving a data element or an invocation to a function of a client-server interface), it is possible to avoid determined activities under special circumstances. As an example, given a diagnostic service provided by a SW-C and given the SW-C shall not allow any diagnostic call under a special state, we only need to define a `ModeDisablingDependency` for the corresponding RTEEvent (in this case `OperationInvokedEvent`) and the undesired state. Is there a call being made for this function, when the SW-C is in the undesired state, the RTEEvent (`OperationInvokedEvent`) will be discarded and thus the function of the SW-C won't be called. This means that the state of the SW-Cs can be considered by the diagnostics, behavior is up to the application developer. For more information, please check the RTE specification and the AUTOSAR Services Document (DCM Service) of System Team.

5.2 Interior Light Actuator(s)

All Actuator SW-Cs process commands from Core SW-C due to environmental restrictions (e.g. over voltage compensation/shutdown if feasible; otherwise complex device driver). Also hardware dependencies are considered (e.g. is PWM possible for corresponding actuator). Status feedback is generated (only operational/defective; no detailed diagnosis).

Actuator SW-Cs do not consider system restrictions (e.g. restrictions due to mechanical reasons regarding activation of several actuators). This should be task of a separate Actuator-Adapter SW-C.

IMPORTANT NOTE: There is just one Actuator SW-C described representatively for a bunch of Actuator SW-C's corresponding to the list of interior light sources & groups of light sources.

5.2.1 Require Ports

Port Name	Activation			
Reference to interface	CmdBrightness			
Direction	required (by SW-C CabinLightFrontMiddleActuator)			
Description	The brightness of cabin_light_front_middle is requested by InteriorLightManager. The percentage value (independent of light actuator type and environment e.g. voltage of electrical system) depends on the functionality that the light source shall be switched on for. 100 percent means full power of the light actuator.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	command	{0;0}	yes	---

Port Name	Voltage			
Reference to interface	BatteryVoltage			
Direction	required (by SW-C CabinLightFrontMiddleActuator)			
Description	used for over-/under voltage compensation			
Data Elements	Name	Init Value	can invalidate	Further attributes
	value	UNKNOWN	no	---

5.2.2 Provide Ports

IMPORTANT NOTE: Ports of Actuator SW-C to the BSW are NOT in scope of this document.

Port Name	Fault			
Reference to interface	StaLightFault			
Direction	provided (by SW-C CabinLightFrontMiddleActuator)			
Description	operational / defective feedback from an actuator.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	operational	no	---

5.2.3 Diagnostic Ports

The port defined below is supposed to be connected to a Central Diagnostic SW-C (above RTE !). Please note that the port "Fault" might also be connected to this Central Diagnostic SW-C in order to read the Operational/Defective Feedback.

Port Name	DiagnosticRequest			
Reference to interface	DiagBrightness			
Direction	required (by SW-C CabinLightFrontMiddleActuator)			
Description	Diagnostic request to manipulate a LightActuator accordingly in order to unconditionally activate LightActuator with a certain brightness or release the LightActuator.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	{0;off}	no	---

Due to it's diagnostic capabilities each Interior Light Actuator provides one or several of the following (client-)ports to be connected to one port of DEM (corresponding to a certain local diagnostic ID).

Port Name	GenericFault			
Reference to interface	DiagnosticMonitor (see AUTOSAR_Services.doc)			
Direction	client (by SW-C CabinLightFrontMiddleActuator)			
Description	Generic fault condition which can't be further detailed. As soon as the error reason can be specified, use one of the ports defined below.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	---	no	---

Port Name	ActuatorFault			
Reference to interface	DiagnosticMonitor (see AUTOSAR_Services.doc)			
Direction	client (by SW-C CabinLightFrontMiddleActuator)			
Description	Diagnostic event for defective actuator			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	---	no	---

Port Name	OpenLoad			
Reference to interface	DiagnosticMonitor (see AUTOSAR_Services.doc)			
Direction	client (by SW-C CabinLightFrontMiddleActuator)			
Description	Actuator doesn't consume power although it's activated e.g. usable for "all bulbs broken".			
Data Elements	Name	Init Value	can invalidate	Further attributes

	status	---	no	---
--	--------	-----	----	-----

Port Name	UnderLoad			
Reference to interface	DiagnosticMonitor (see AUTOSAR_Services.doc)			
Direction	client (by SW-C CabinLightFrontMiddleActuator)			
Description	Actuator doesn't consume enough power although it's activated e.g. usable for "one/some bulbs broken". Further differentiation at implementation level if needed.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	---	no	---

Port Name	OverLoad			
Reference to interface	DiagnosticMonitor (see AUTOSAR_Services.doc)			
Direction	client (by SW-C CabinLightFrontMiddleActuator)			
Description	Actuator consumes too much power when activated			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	---	no	---

Port Name	ShortToGnd			
Reference to interface	DiagnosticMonitor (see AUTOSAR_Services.doc)			
Direction	client (by SW-C CabinLightFrontMiddleActuator)			
Description	Actuator consumes excessive power (connected to battery voltage)			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	---	no	---

Port Name	ShortToBatt			
Reference to interface	DiagnosticMonitor (see AUTOSAR_Services.doc)			
Direction	client (by SW-C CabinLightFrontMiddleActuator)			
Description	Actuator unexpectedly consumes power (not connected to battery voltage)			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	---	no	---

Port Name	OpenLoadShortToGnd			
Reference to interface	DiagnosticMonitor (see AUTOSAR_Services.doc)			
Direction	client (by SW-C CabinLightFrontMiddleActuator)			
Description	Error is either OpenLoad or ShortToGnd			
Data Elements	Name	Init Value	can in-	Further attrib-

			<i>validate</i>	<i>utes</i>
	status	---	no	---

Port Name	OpenLoadShortToBatt			
Reference to interface	DiagnosticMonitor (see AUTOSAR_Services.doc)			
Direction	client (by SW-C CabinLightFrontMiddleActuator)			
Description	Error is either OpenLoad or ShortToBatt			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	---	no	---

5.2.4 Cal-Ports

For each Interior Light Actuator certain parameters are accessible via “cal-ports” (further details are supposed to be specified by RTE-Team).

Examples are:

- value for “full brightness”: light actuator is set to e.g. 90% of full brightness due to possibility to compensate low voltage.
- ramp characteristics.

Please note that “pure ROM parameters” (like PWM frequency) are not listed here.

5.3 Interior Light Sensor(s)

There is one Sensor SW-C (**HMI**) which ensures (as far as needed):

- filtering (debouncing) e.g. calculation of stable output values e.g. by debouncing
- conditioning e.g. calculation of average value
- plausibility check as basis for failure detection (implausible?)
- failure detection (but no error-handling/ error-compensation) e.g. valid/not valid

5.3.1 Require Ports

IMPORTANT NOTE: Ports of Sensor SW-C to the BSW are NOT in scope of this document.

5.3.2 Provide Ports

Port Name	CabinLightFnCeHMI			
Reference to interface	ReqIntLight			
Direction	provided (by SW-C HMI)			
Description	User request to activate / deactivate cabin_light_front_middle			
Data Elements	Name	Init Value	can invalidate	Further attributes

	request	unknown	yes	---
--	---------	---------	-----	-----

Please note that this port is representative for many other ports **[IntLightSrc]HMI** (see also Fig 3.1-2) used to request to activate / deactivate an interior light actuator denoted by [IntLightSrc] or a virtual interior light group also denoted by [IntLightSrc]. This port is not necessary for each interior light actuator, it is not intended for hard-wired interior lights.

Port Name	DashBoardBrightness			
Reference to interface	ReqBrightness			
Direction	provided (by SW-C HMI)			
Description	Information how to set dashboard brightness.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	0	yes	---

5.3.3 Diagnostic Ports

The ports defined below are supposed to be connected to a Central Diagnostic SW-C (above RTE !).

Port Name	DiagnosticRequest			
Reference to interface	DiagSensor			
Direction	required (by SW-C CabinLightFrontMiddleActuator)			
Description	Diagnostic request to manipulate a Interior Light Sensor SW-C accordingly in order unconditionally activate HMI output with a certain value or unconditionally release HMI SW-C afterwards.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	{0;off}	no	---

Port Name	DiagnosticRequest			
Reference to interface	DiagBrightness			
Direction	required (by SW-C Set Dash Board)			
Description	Diagnostic request to manipulate Set Dash Board in order to unconditionally activate the output of this SW-C with a certain brightness or release diagnostic request.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	{0;off}	no	---

HINT: not yet standardized: Diagnostic requests for reading HMI SW-C's input-values.

Due to its diagnostic capabilities each Interior Light Sensor provides one or several of the following (client-)ports to be connected to one port of DEM (corresponding to a certain local diagnostic ID).

Port Name	GenericFault			
Reference to interface	DiagnosticMonitor (see AUTOSAR_Services.doc)			
Direction	client			
Description	SW-C detects invalid input condition which can't be further detailed. As soon as the error reason can be specified, use one of the ports defined below.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	---	no	---

Port Name	SensorFault			
Reference to interface	DiagnosticMonitor (see AUTOSAR_Services.doc)			
Direction	client			
Description	Diagnostic event for defective sensor (e.g. pushbutton activated extensively)			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	---	no	---

Port Name	OpenLoad			
Reference to interface	DiagnosticMonitor (see AUTOSAR_Services.doc)			
Direction	client			
Description	Diagnostic event for open circuit/ broken line			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	---	no	---

Port Name	ShortToGnd			
Reference to interface	DiagnosticMonitor (see AUTOSAR_Services.doc)			
Direction	client			
Description	Diagnostic event for input permanently connected to ground			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	---	no	---

Port Name	ShortToBatt			
Reference to interface	DiagnosticMonitor (see AUTOSAR_Services.doc)			
Direction	client			
Description	Diagnostic event for input permanently connected to battery voltage			

<i>Data Elements</i>	<i>Name</i>	<i>Init Value</i>	<i>can invalidate</i>	<i>Further attributes</i>
	status	---	no	---

Port Name	OpenLoadShortToGnd			
Reference to interface	DiagnosticMonitor (see AUTOSAR_Services.doc)			
Direction	client			
Description	Error is either OpenLoad or ShortToGnd			
<i>Data Elements</i>	<i>Name</i>	<i>Init Value</i>	<i>can invalidate</i>	<i>Further attributes</i>
	status	---	no	---

Port Name	OpenLoadShortToBatt			
Reference to interface	DiagnosticMonitor (see AUTOSAR_Services.doc)			
Direction	client			
Description	Error is either OpenLoad or ShortToBatt			
<i>Data Elements</i>	<i>Name</i>	<i>Init Value</i>	<i>can invalidate</i>	<i>Further attributes</i>
	status	---	no	---

5.3.4 Cal-Ports

There should be a parameter for debounce time.

Please note that “pure ROM parameters” (like PWM frequency) are not listed here.

5.4 Exterior Light

This chapter covers only the relevant interfaces between Interior Light and Exterior-Light. For details see decomposition of ExteriorLight; for a complete list of all provide & require ports see ExteriorLight - Architectural Overview.

5.4.1 Require Ports

There is NO information from Interior Light required by ExteriorLight.

5.4.2 Provide Ports

The following information is provided by Exterior Light by means of ports as supposed:

- OutsideBrightness
- GlobalDimLight
- LowBeamInd

ParkingLightsInd
CHLHome

5.5 Central Locking

This chapter covers only the relevant interfaces between Interior Light and Central Locking. For details see decomposition of Central Locking; for a complete list of all provide & require ports see Central Locking - Architectural Overview.

5.5.1 Require Ports

There is NO information from Interior Light required by Central Locking.

5.5.2 Provide Ports

The following information is provided by Exterior Light by means of ports as supposed:

- DriverDoorState
- PassengerDoorState
- RearLeftDoorState
- RearRightDoorState
- RearLidState
- BackliteWindowState
- DriverDoorKeyAccess
- PassengerDoorKeyAccess
- AccessKeyless
- AccessRemoteKey
- CarFinder

5.6 other

This chapter covers only the relevant interfaces between Interior Light and Anti-Theft-System, Profile_manager & Inter-Domain Information. For details see corresponding decompositions and Architectural Overviews.

5.6.1 Require Ports

There is NO information from Interior Light required by other applications / domains.

5.6.2 Provide Ports

The following information is provided by Anti-Theft-System by means of ports as supposed: VisualAlarm

The following information is provided by Profile_Manager by means of ports as supposed: DriverProfile

The following inter-domain-information is provided by means of ports as supposed:

- Crash
- OperationMode
- EnergyMgmt
- Speed
- Voltage

6 APPENDIX- Interface Descriptions

This chapter gives an overview on the referenced interfaces.

6.1 Inter Domain Interfaces

Please note that Inter Domain Interfaces are listed for visualisation purpose and subject to be harmonized in later revisions.

6.1.1 StaOperationMode

<i>Interface</i>				
Name (of Interface)		StaOperationMode		
Communcation Method		SenderReceiver		
Description		<p>This interface provides the car operation states. It can be seen as an abstraction and extension of the normal ignition switch position (also called "clamp position") which is traditionally used to guide functional operation of the vehicle. It also includes information about what state the engine is in, i.e. running or not.</p> <p>This interface should be used to control which functionalities / components are active in the vehicle.</p> <p>All states have to be implemented in all cars, but not necessarily used.</p>		
isService		No (e.g. no BSW service)		
1	Data Element Name	status		
	Data Type	enum		
	Symbolic Constants	Label	Value	Comment
		prepare_idle	10	Transient state: might be used to prepare for going to idle state.
idle		0	No valid user identification detected; car is waiting for valid identification. Most systems should be in power-save mode. Only functionalities that do not require a valid identification may be active / operational (e.g. hazard warning, burglar alarm, Id detection mechanisms, parking light).	
prepare_identification	6	Transient state: components / functionalities involved in identification should be operational (may include overall system consistency checks to ensure that the system was not tampered during idle state etc.). State might be used to prepare for id_approved if start-up time is needed.		

	id_approved	1	Valid user identification detected in or around the car. Certain body functionalities may be active / operational (e.g. locking / unlocking of doors, (de-)activation of burglar alarm).
	post_accessory	9	Transient state: might be used to prepare for going to idle state (e.g. by storing persistent data to persistent storage).
	accessory	2	Valid user identification detected in or around the car. Certain accessory functionalities in the car may be active / operational (e.g. radio).
	post_ignition_on	8	Transient state: might be used to prepare for going to idle state (e.g. by storing persistent data to persistent storage).
	ignition_on	3	Ignition is on. Most power consumers may be at least temporarily active / operational; engine is not running.
	engine_running	5	Ignition is on and combustion engine is running.
	pre_crank	7	Transient state: cranking is about to begin. State might be used to prepare for cranking (high power consumers might be switched off) and possible low power situation (e.g. by storing persistent data to persistent storage).
	crank	4	Start of engine. Power consumers might be switched off.
	reserved_1	11	--
	reserved_2	12	--
	reserved_3	13	--
	reserved_4	14	--
	invalid	15	--
Data/Event	Data		

Remarks	<p>Length (bit) = 4; Recommendations: Init value= idle</p> <p>Hierarchy: <i>Super-state:</i> idle <i>State:</i> prepare_idle <i>State:</i> idle <i>Super-state:</i> identification <i>State:</i> prepare_identification <i>State:</i> id_approved <i>State:</i> post_accessory <i>Super-state:</i> accessory <i>State:</i> accessory <i>State:</i> post_ignition_on <i>Super-state:</i> ignition <i>State:</i> ignition_on <i>State:</i> engine_running <i>Super-state:</i> crank <i>State:</i> pre_crank <i>State:</i> crank</p> <p><i>Base modes</i> (idle, id_approved, accessory, ignition_on, crank, engine_running):</p> <ul style="list-style-type: none"> • Elements are fixed, no additional elements are possible • It must be implemented. <p><i>Special modes</i> (prepare_identification, pre_crank, post_ignition_on, post_accessory, prepare_idle, ...):</p> <ul style="list-style-type: none"> • Additional elements are possible. <p>The implementation is voluntary.</p>
----------------	--

6.1.2 StaCrash

Interface			
Name (of Interface)	StaCrash		
Communication Method	SenderReceiver		
Description	This interface provides the current crash status (values as pre-crash, post-crash, ...)..		
isService	No (e.g. no BSW service)		
1 Data Element Name	status		
Data Type	enum		
Symbolic Constants	Label	Value	Comment
		0	--
	pre_crash	1	--
	post_crash	2	--
	...	3	--
		4	--
Data/Event	Data		
Remarks			

6.1.3 StaEnergyMgmt

Interface	
Name (of Interface)	StaEnergyMgmt
Communication Method	SenderReceiver
Description	This interface provides the information whether all, basic or only

	keep-alive components/functionalities are allowed to run. As long as the status is valid, the corresponding value as defined by the table below should be provided.		
isService	No (e.g. no BSW service)		
1 Data Element Name	status		
Data Type	enum		
Symbolic Constants	Label	Value	Comment
	no_funct_possible	0	only for completeness
	safety_funct_only	1	only safety functionalities / components active (functionalities considered as relevant for passenger safety)
	hibernate_level_1	2	level 1 for deactivation of functionalities / components during long-term parking (to save battery capacity)
	hibernate_level_2	3	level 2 for deactivation of functionalities / components during long-term parking (to save battery capacity)
	hibernate_level_3	4	level 3 for deactivation of functionalities / components during long-term parking (to save battery capacity)
	lim_comf_level_1	5	level 1 for deactivation of functionalities / components in case of improper load-balance of battery / generator during normal use of the vehicle
	lim_comf_level_2	6	level 2 for deactivation of functionalities / components in case of improper load-balance of battery / generator during normal use of the vehicle
	lim_comf_level_3	7	level 3 for deactivation of functionalities / components in case of improper load-balance of battery / generator during normal use of the vehicle
	lim_comf_level_4	8	level 4 for deactivation of functionalities / components in case of improper load-balance of battery / generator during normal use of the vehicle
	full_funct	9	default (no functionalities / components deactivated due to energy management reasons)
	consume_energy	10	valid when battery is unbalanced or when more load has to be applied onto generator
	reserved_1	11	--
	reserved_2	12	--
	reserved_3	13	--
	reserved_4	14	--
Data/Event	Data		

Remarks	Length (bit) = 4; Recommendations: Init value= full_funct NOTE: this interface is to reflect the "load dump" stages in case of improper battery/generator balance. The handling of over- or under-voltage is not covered within this interface.
----------------	--

6.1.4 VehicleSpeed

Interface		
Name (of Interface)	VehicleSpeed	
Communication Method	SenderReceiver	
Description	This interface provides the information about the current speed of the car. Note, that this is an absolute value, i.e. a negative physical speed shall appear as a positive value.	
isService	No (e.g. no BSW service)	
1 Data Element Name	value	
Data Type	numeric	
Physical Interpretation	Offset	0
	Unit	km/h
Physical Values	Min Value	0
	Max Value	409,4
	Resolution	0.1
Data/Event	Data	
Remarks	Length (bit) = 12; Recommendations: Init value= 0	

6.1.5 BatteryVoltage

Interface		
Name (of Interface)	BatteryVoltage	
Communication Method	SenderReceiver	
Description	This interface provides the actual voltage level as measured at the battery.	
isService	No (e.g. no BSW service)	
1 Data Element Name	value	
Data Type	numeric	
Physical Interpretation	Offset	0
	Unit	V
Physical Values	Min Value	0
	Max Value	25.2
	Resolution	0.1
Data/Event	Data	
Remarks	Length (bit) = 8; Recommendations: Init value= UNKNOWN Proposal for "special values": "voltage measured is 25.3V or bigger" (253). "invalid" (254) a defect in the measuring system is detected "unknown" (255) for initialisation before the first available measurement.	