Document Title	Standardization of Mirror Adjustment and Tinting 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	• "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

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 Contents

1 Purpose of this Document

This document describes a possible basic architecture of the mirror functionality.

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

Currently, this document is in a shape that allows discussions regarding possibly to-be-standardized component interfaces.

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

Section 4 deals with interfaces from and to the outside world of the mirror 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:

- Personalization parameters and interfaces
- Variant coding
- Diagnostic coding

2 Prerequisites

This section gives reference to related AUTOSAR documents as well as an introduction to the templates used in this document to describe ports, types and interfaces.

Other Work-Packages are – at the time of writing this document – busily crafting a formal approach to describe an AUTOSAR Software Component, therefore a rather informative approach is chosen in this document. Once the above mentioned formal Software Component template is available, the content of this document has to be transferred to the formal description. Nevertheless, we expect that we generally also need a design description similar to this document for all further software component groups in order to provide a basis for discussion with company experts.

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	`AccessRemoteKey`	`ReqAccessRemoteKey`	`request`

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] AUTOSAR Methodology
https://svn.autosar.org/repos/10Releases/
AUTOSAR_Methodology.ppt

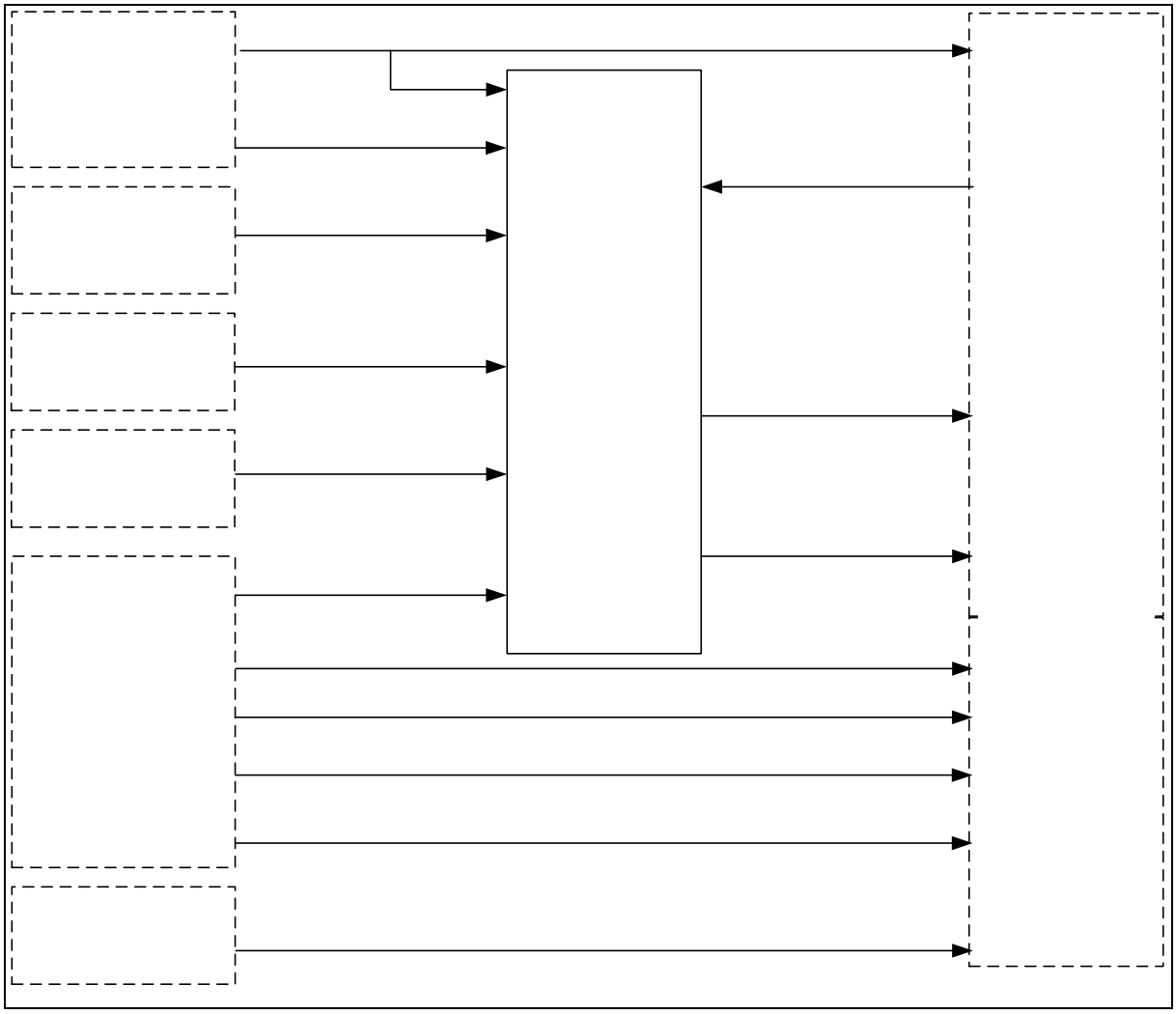
[3] Specification of the Virtual Functional Bus
https://svn.autosar.org/repos/10Releases/
AUTOSAR_VirtualFunctionBus.pdf

Application Interfaces (including Inter-domain & Intra-domain interfaces):
see Appendix

3 Mirror – Decomposition Overview

3.1 Mirror adjustment Decomposition

Components related to MirrorAdjustment are bundled in several Blocks.



[ManAdjustment]HMI

HMI

ReqAutoAd

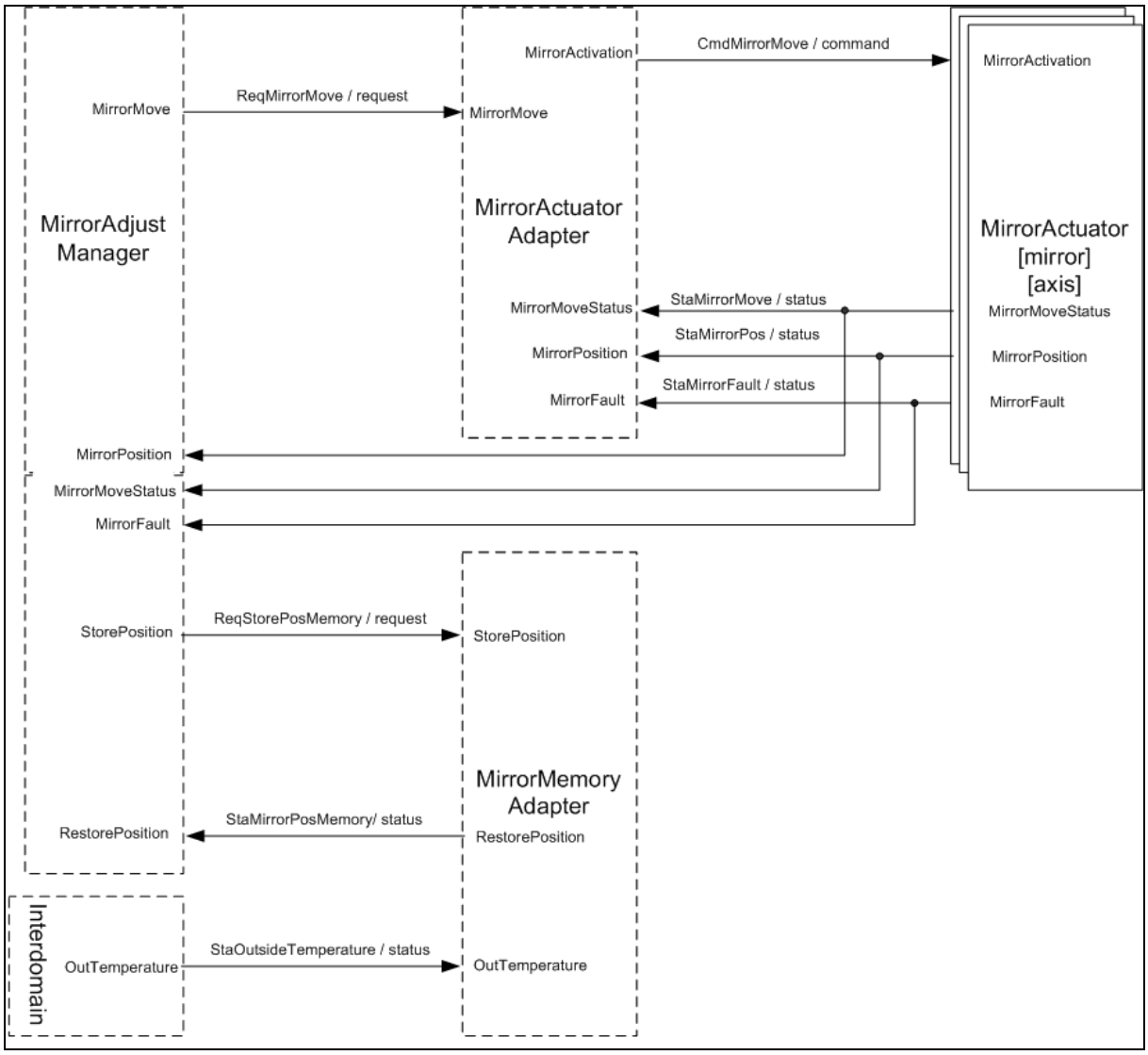


Fig 3.1-1 shows the SW-Cs mentioned above and the software components contained. Arrows indicate data flows.

Inside MirrorAdjustManager the core functionality is implemented.

HMI contains the Mirror Manual Adjustment, detecting the user's demands for manual adjustment and the Mirror Auto Adjustment, the panel with the memory buttons. It detects the user's demands to store and recall mirror positions.

The Profile manager uses information from the key transponder, keyless access and remote key to generate the current driver profile identification.

The Remote key indicates a user request from the remote key and this may trigger the Auto adjust user request adapter to perform a store or recall.

The Keyless access indicates a user request from keyless access and this may trigger the Auto adjust user request adapter to perform a store or recall.

Inter domain encapsulates SW-Cs that exchange cross-domain information such as selected gear position, operational mode or energy management. Through Inter Domain Interfaces inter-domain status information is exchanged.

Central locking status is received from the Central Locking component.

The Auto adjust user request adapter combines and prioritizes all the information from above and requests mirror movements and storing of memory positions.

Mirror actuator adapter knows about the electrical and mechanical constraints of each single axes of the mirror. It converts requests from the Mirror adjust manager to commands to each single axes (actuators). It compares the current position of the mirror axis with the desired position.

Mirror memory adapter is responsible for the actual memory position storage and recall. This may be local or global storage depending on the personalization control strategy. External temperature information is used to compensate storage and re-storage of mirror positions.

[mirror][axis]actuator is the software representation of a mechanical actuator. It is able to move one axis of the mirror to a certain direction. It also detects the position of the axis.

3.2 Mirror Tinting Decomposition

Components related to Mirror tinting are bundled in several Blocks.

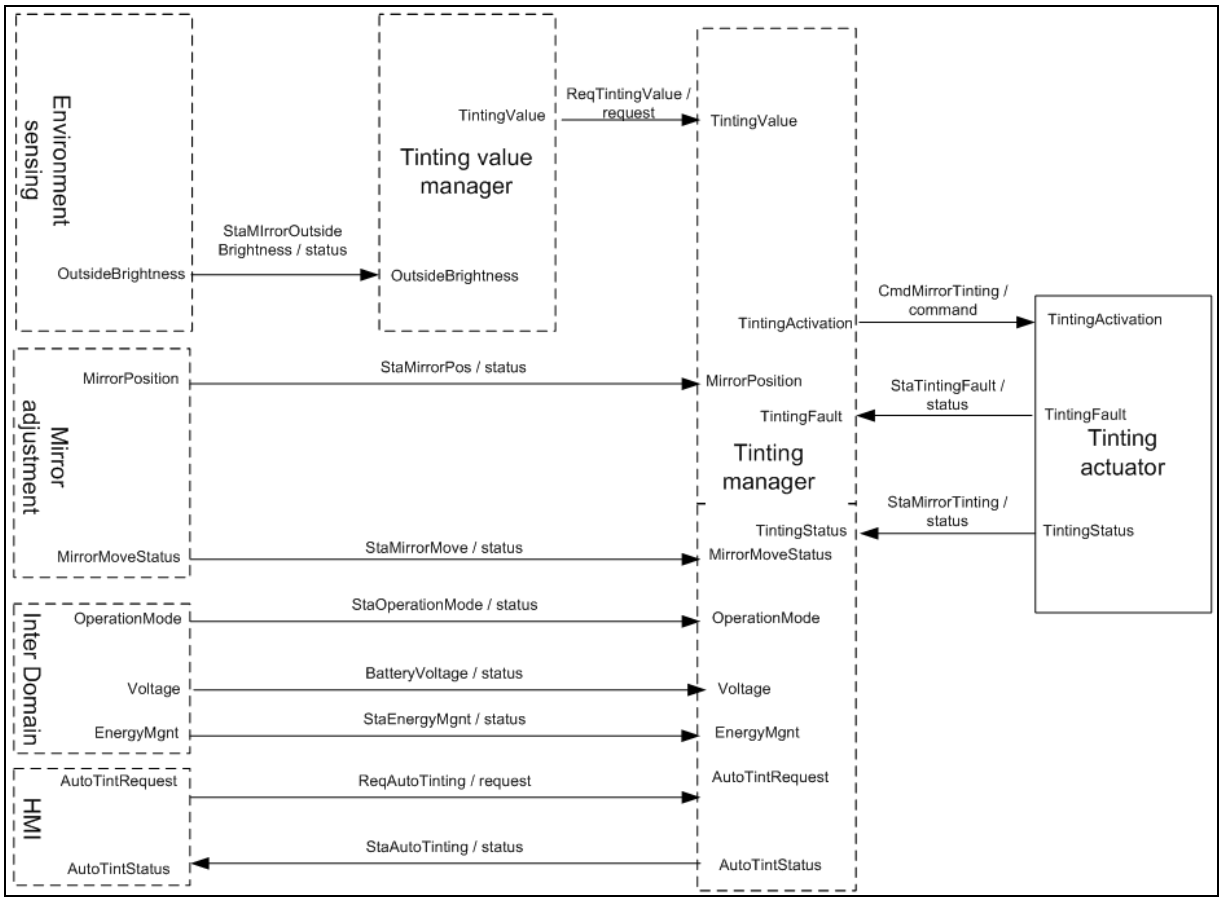


Fig 3.2-1 shows the SW-Cs mentioned above and the software components contained. Arrows indicate data flows.

Inside Tinting Manager the core functionality is implemented.

Based on the information of the Environment Sensing Component, the Tinting Value Manager Component calculates the desired tinting value using the outside brightness information.

The Inter domain, HIM and Mirror Adjustment Component provide information about the necessity of automatic tinting.

The Mirror Tinting Actuator executes the demands of the Tinting Manager and delivers Status Information back to the Manager

3.3 Design Rationale

This design standard is the AUTOSAR-architecture (VFB-view) for mirror adjustment and tinting. It provides a decomposition into SW-Cs and a list of standardized interfaces related to Mirror Adjustment and Tinting.
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. These SW-Cs will be obtained as a unit so that all the internal (and hence not AUTOSAR standardized) interfaces are controlled 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.

4 Interface Descriptions

This chapter gives an overview on 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 Interfaces

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

4.1.1 Ports referencing StaOperationMode

Used By	***Reason***
Mirror adjust manager	Some functions of mirror adjustment may depend on the status of a certain operational mode. (example use case: unfold outside mirrors with ignition on, or tinting with ignition on).
Mirror tinting manager	

4.1.2 Ports referencing VehicleSpeed

Used By	***Reason***
Mirror adjust manager	Used to unfold the mirrors at a specific speed or to inhibit memory-folding if speed is above a specific value.

4.1.3 Ports referencing BatteryVoltage

Used By	***Reason***
MirrorAdjust Manager Voltage	use cases: voltage compensation, disable mirror movement in case of over/under voltage
TintingManager	

4.1.4 Ports referencing StaEnergyMgnt

Used By	***Reason***
Mirror adjust manager	Use case: activation of one or more motors depends on the current battery capacity.

4.1.5 Ports referencing StaGear

Used By	***Reason***
Auto adjust user request	Use case: move down to outer mirror angle to provide the best

adapter , (Mirror adjust manager)	view to the curb when the reverse gear is selected.

4.1.6 Ports referencing StaCentralLocking

Used By	***Reason***
Mirror adjust manager	Use case: folding outer mirrors with locking

4.1.7 Ports referencing StaOutsideTemperature

Used By	***Reason***
Mirror memory adapter	Used to retrieve the external environment temperature for compensating the stored and restored mirror profile positions.

4.2 Overview on Intra Domain Interfaces

4.2.1 Ports referencing ReqManAdjustPanel

Used By	***Reason***
HMI	Used for the driver's request to manually move the mirrors to desired position.
Auto Adjust User Request Manager	
Mirror Adjust Manager	

PortInterface					
Name (of PortInterface)		ReqManAdjustPanel			
Communication Method		SenderReceiver			
Description		User request to move the axis of the mirror manually in a certain direction.			
isService		No (e.g. no BSW service)			
1	***Data Element Name***	request			
	Data Type	record			
	Record elements	***Element***	***Label***	***Value***	***Comment***
		mirror_ type (of type enum)	no_mirror	0	--
			driver_miror	1	--
			passen-ger_mirror	2	--
			interior_mirror	3	--
		axis_type (of type enum)	no_axis	0	--
			horiz_axis	1	--
			vert_axis	2	--
		movement_type (of type enum)	idle	0	--
			increase	1	--
			decrease	2	--
	Data/Event	Data			
	Remarks	Length (bit) = 2+2+2 Recommendations: Init value={no_mirror, no_axis, idle}; can invalidate=No.			

4.2.2 Ports referencing ReqAutoAdjustPanelMirror

Used By	*Reason*
HMI	Used for the driver's request to move the mirror to memorized position
Auto Adjust User Request Manager	

PortInterface					
Name (of PortInterface)		ReqAutoAdjustPanelMirror			
Communication Method		SenderReceiver			
Description		User request to move the mirror automatically or to store the mirror position to the given profile.			
isService		No (e.g. no BSW service)			
1	***Data Element Name***	request			
	Data Type	record			
	Record elements	***Element***	***Label***	***Value***	***Comment***
		mirror_profile (of type t_profile)	no_profile	0	--
			profile_1	1	--
			....	--	--
			profile_14	14	--
			profile_15	15	--
		operation (of type enum)	off	0	--
			recall	1	--
			store	2	--
	Data/Event	Data			
	Remarks	Length (bit) = 4+2 Recommendations: Init value={no_profile, off}; can invalidate=No.			

4.2.3 Ports referencing StaMirrorAction

Used By	*Reason, comment*
Mirror adjust manager	Used to retrieve from the mirror adjust manager the type of action in progress.
Auto Adjust User Request Manager	

PortInterface					
Name (of PortInterface)		StaMirrorAction			
Communication Method		SenderReceiver			
Description		Feedback from the mirror manager about the kind of action that is currently running (mirror moving or mirror stopped). It might be necessary to immediately stop automatic positioning if the driver hits any key (panic-stop).			
isService		No (e.g. no BSW service)			
1	***Data Element Name***	Status			
	Data Type	enum			
	Record elements	***Element***	***Label***	***Value***	***Comment***
		mirror_ type (of type enum)	no_mirror	0	--
			driver_miror	1	--
			passen-ger_mirror	2	--
			interior_mirror	3	--
		movement_type (of type enum)	idle	0	--
			increase	1	--
			decrease	2	--
	Data/Event	Data			

	Remarks	Length (bit) = 2+2 Recommendations: Init value={no_mirror, idle}, can invalidate=No.

4.2.4 Ports referencing ReqMirrorAutoMove

Used By	***Reason***
Mirror adjust manager	Used to send the automatic move request to the mirror adjustment manager.
Auto Adjust User Request Manager	

PortInterface				
Name (of PortInterface)		ReqMirrorAutoMove		
Communication Method		SenderReceiver		
Description		Request to move the mirror automatically to a certain position. This position could be determined by a user request from the mirror panel, an access from the remote key, a keyless access or if reverse gear was detected, to down outer mirror angle to provide the best view to the curb.		
isService		No (e.g. no BSW service)		
1	***Data Element Name***	Request		
	Data Type	enum		
	Record elements	***Element***	***Label***	***Value***	***Comment***
		mirror_ type (of type enum)	no_mirror	0	--
			driver_miror	1	--
			passenger_mirror	2	--
			interior_mirror	3	--
		mirror_profile (of type t_profile)	no_profile	0	--
			profile_1	1	--
			....		--
			profile_14	14	--
			profile_15	15	--
	Data/Event	Data			
	Remarks	Length (bit) = 2+4 Recommendations: Init value={no_mirror, no_profile }; can invalidate=No			

4.2.5 Ports referencing ReqStoreMirrorPos

Used By	***Reason***
Mirror adjust manager	Used to send the position store request to the mirror adjustment manager.
Auto Adjust User Request Manager	

PortInterface	
Name (of PortInterface)	ReqStoreMirrorPos
Communication Method	SenderReceiver
Description	Request to store the position of a certain mirror to the corresponding profile number.
isService	No (e.g. no BSW service)

1	***Data Element Name***	Request			
	Data Type	Enum			
	Record elements	***Element***	***Label***	***Value***	***Comment***
		mirror_ type (of type enum)	no_mirror	0	--
			driver_miror	1	--
			passen-ger_mirror	2	--
			interior_mirror	3	--
		mirror_profile (of type t_profile)	no_profile	0	--
			profile_1	1	--
			....		--
			profile_14	14	--
			profile_15	15	--
	Data/Event	Data			
	Remarks	Length (bit) = 2+4 Recommendations: Init value={no_mirror, no_profile}; can invali-date=No			

4.2.6 Ports referencing ReqMirrorMove

Used By	***Reason***
Mirror adjust manager	Used to send the movement request to the mirror actuator adapter.
Auto actuator adapter	

PortInterface					
Name (of PortInterface)	ReqMirrorMove				
Communication Method	SenderReceiver				
Description	Request to move the axis of the mentioned mirror to the indicated position.				
isService	No (e.g. no BSW service)				
1	***Data Element Name***	Request			
	Data Type	enum			
	Record elements	***Element***	***Label***	***Value***	***Comment***
		mirror_ type (of type enum)	no_mirror	0	--
			driver_miror	1	--
			passen-ger_mirror	2	--
			interior_mirror	3	--
		axis_type (of type enum)	no_axis	0	--
			horiz_axis	1	--
			vert_axis	2	--
		position (of type t_position)	***Offset***	0	
			Unit	%	
			Min Value	0	
			Max Value	100.0 100.1 special use (un-known, stop, don't care)	
			Resolution	0.1	
	Data/Event	Data			
	Remarks	Length (bit) = 2+2+10 Recommendations: Init value={no_mirror, no_axis,0}; can invali-date=No			

4.2.7 Ports referencing CmdMirrorMove

Used By	***Reason***
Mirror actuator adapter	Used to send the movement request to the mirror actuator.
Mirror actuator [mirror][axis]	

PortInterface					
Name (of PortInterface)		CmdMirrorMove			
Communication Method		SenderReceiver			
Description		Command to move the axis in a certain direction.			
isService		No (e.g. no BSW service)			
1	***Data Element Name***	request			
	Data Type	record			
	Record elements	***Element***	***Label***	***Value***	***Comment***
		mirror_ type (of type enum)	no_mirror	0	--
			driver_miror	1	--
			passen-ger_mirror	2	--
			interior_mirror	3	--
		axis_type (of type enum)	no_axis	0	--
			horiz_axis	1	--
			vert_axis	2	--
		movement_type (of type enum)	idle	0	--
			increase	1	--
			decrease	2	--
	Data/Event	Data			
	Remarks	Length (bit) = 2+2+2 Recommendations: Init value={no_mirror, no_axis, idle}; can invali-date=No.			

4.2.8 Ports referencing StaMirrorPos

Used By	***Reason***
Mirror adjust manager	Used to retrieve from the actuator the mirror's actual position.
Mirror actuator [mirror][axis]	

PortInterface					
Name (of PortInterface)		StaMirrorPos			
Communication Method		SenderReceiver			
Description		Actual mirror position			
isService		No (e.g. no BSW service)			
1	***Data Element Name***	Request			
	Data Type	enum			
	Record elements	***Element***	***Label***	***Value***	***Comment***
		mirror_ type (of type enum)	no_mirror	0	--
			driver_miror	1	--
			passen-ger_mirror	2	--
			interior_mirror	3	--
		axis_type (of type enum)	no_axis	0	--
			horiz_axis	1	--

			vert_axis	2	--
		position (of type t_position)	***Offset***	0	
			Unit	Percent	
			Min Value	0	
			Max Value	1000 1001 special use (unknown, stop, don't care)	
			Resolution	0.1	
	Data/Event	Data			
	Remarks	Length (bit) = 2+2+10 Recommendations: Init value={no_mirror, no_axis,0}; can invalidate=No			

4.2.9 Ports referencing StaMirrorMove

Used By	***Reason***
Mirror adjust manager	Used to retrieve from the actuator the mirror's movement status.
[mirror][axis] actuator	

PortInterface					
Name (of PortInterface)		StaMirrorMove			
Communication Method		SenderReceiver			
Description		Status of movement of a mirror on an axis in a certain direction.			
isService		No (e.g. no BSW service)			
1	***Data Element Name***	request			
	Data Type	record			
	Record elements	***Element***	***Label***	***Value***	***Comment***
		mirror_ type (of type enum)	no_mirror	0	--
			driver_miror	1	--
			passen-ger_mirror	2	--
			interior_mirror	3	--
		axis_type (of type enum)	no_axis	0	--
			horiz_axis	1	--
			vert_axis	2	--
		movement_type (of type enum)	idle	0	--
			increase	1	--
			decrease	2	--
	Data/Event	Data			
	Remarks	Length (bit) = 2+2+2 Recommendations: Init value={no_mirror, no_axis, idle}; can invalidate=No.			

4.2.10 Ports referencing ReqStorePosMemory

Used By	***Reason***
Mirror adjust manager	Used to store the position of the mirrors to the memory position 'profile'.
Mirror memory adapter	

PortInterface	
Name (of PortInterface)	ReqStorePosMemory
Communication Method	SenderReceiver

Description		Request to store the position of a certain mirror and axis to the memory position 'profile'.			
isService		No (e.g. no BSW service)			
1	***Data Element Name***	Request			
	Data Type	record			
	Record elements	***Element***	***Label***	***Value***	***Comment***
		mirror_ type (of type enum)	no_mirror	0	--
			driver_miror	1	--
			passenger_mirror	2	--
			interior_mirror	3	--
		axis_type (of type enum)	no_axis	0	--
			horiz_axis	1	--
			vert_axis	2	--
		mirror_profile (of type t_profile)	no_profile	0	--
			profile_1	1	--
			....		--
			profile_14	14	--
			profile_15	15	--
		position (of type t_position)	***Offset***	0	
			Unit	Percent	
			Min Value	0	
			Max Value	1000 1001 special use (unknown, stop, don't care)	
			Resolution	0.1	
	Data/Event	Data			
	Remarks	Length (bit) = 2+2+4+10 Recommendations: Init value={no_mirror, no_axis, no_profile, 0}; can invalidate=No			

4.2.11 Ports referencing StaMirrorPosMemory

Used By	***Reason***
Mirror adjust manager	Used to recover the mirrors memorize position from "profile".
Mirror memory adapter	

PortInterface					
Name (of PortInterface)		StaMirrorPosMemory			
Communication Method		SenderReceiver			
Description		Memory-position data from storage memory for the current profile of a certain mirror and axis.			
isService		No (e.g. no BSW service)			
1	***Data Element Name***	Status			
	Data Type	record			
	Record elements	***Element***	***Label***	***Value***	***Comment***
		mirror_ type (of type enum)	no_mirror	0	--
			driver_miror	1	--
			passen-ger_mirror	2	--
			interior_mirror	3	--
		axis_type (of type enum)	no_axis	0	--
			horiz_axis	1	--
			vert_axis	2	--

		mirror_profile (of type t_profile)	no_profile	0	--
			profile_1	1	--
			....		--
			profile_14	14	--
			profile_15	15	--
		position (of type t_position)	***Offset***	0	
			Unit	Percent	
			Min Value	0	
			Max Value	1000 1001 special use (unknown, stop, don't care)	
			Resolution	0.1	
	Data/Event	Data			
	Remarks	Length (bit) = 2+2+4+10 Recommendations: Init value={no_mirror, no_axis, no_profile, 0}; can invalidate=No			

4.2.12 Ports referencing StaMirrorFault

Used By	***Reason, comment***
Mirror Adust Manager	Operational/defective feedback of the mirror actuators
Mirror Actuation Adapter	
Mirror Actuator [mirror][axis]	

PortInterface				
Name (of PortInterface)		StaMirrorFault		
Communcation Method		SenderReceiver		
Description		Used as feedback from an actuator component to indicate that the mirror 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 mirror is broken and should not be actuated. Note:The interface is not used for detailed diagnostics.		
isService		No (e.g. no BSW service)		
1	**Data Element Name**	status		
	Data Type	enum		
	Symbolic Constants	**Label**	**Value**	**Comment**
		operational	0	--
		defective	1	e. g. mirror broken, not moving
	Data/Event	Data		
	Remarks	Length (bit) = 1 Recommendations: Init value=operational, can invalidate=No		

4.2.13 Ports referencing DiagMirrorMove

Used By	***Reason***
Mirror actuator [mirror][axis]	Command to move the axis in a certain direction for diagnostic purposes.

PortInterface	
Name (of PortInterface)	DiagMirrorMove
Communication Method	SenderReceiver
Description	Command to move the axis in a certain direction for diagnostic pur-

		poses.			
isService		No (e.g. no BSW service)			
1	***Data Element Name***	request			
	Data Type	record			
	Record elements	***Element***	***Label***	***Value***	***Comment***
		mirror_ type (of type enum)	no_mirror	0	--
			driver_miror	1	--
			passen-ger_mirror	2	--
			interior_mirror	3	--
		axis_type (of type enum)	no_axis	0	--
			horiz_axis	1	--
			vert_axis	2	--
		movement_type (of type enum)	idle	0	--
			increase	1	--
			decrease	2	--
		diagnostic_type (of type t_onoff)	off	0	diagnostic inactive
			on	1	diagnostic active
	Data/Event	Data			
	Remarks	Length (bit) = 2+2+2+1 Recommendations: Init value={no_mirror, no_axis, idle, off}; can in-validate=No.			

4.2.14 Ports referencing StaMirrorOutsideBrightness

Used By	***Reason***
Environment sensing	Reads the outside brightness from the mirror sensor.
Tinting value manager	

PortInterface			
Name (of PortInterface)		StaMirrorOutsideBrightness	
Communication Method		SenderReceiver	
Description		Incoming brightness from an source outside of the vehicle serves as input value for the calculation of the tinting value	
isService		No (e.g. no BSW service)	
1	***Data Element Name***	Status	
	Data Type	t_percentage	
	Physical Interpretation	***Offset***	0
		Unit	%
	Physical Values	***Min Value***	0
		Max Value	100
		Resolution	1
	Data/Event	Data	
	Remarks	Length (bit) = 7 (see data type) Recommendations: Init value=0, can invalidate=Yes	

4.2.15 Ports referencing ReqAutoTinting

Used By	***Reason***
HMI AutoTintRequest	Automatic tinting request

TintingManager AutoTintRequest	

PortInterface		
Name (of PortInterface)		ReqAutoTinting
Communication Method		SenderReceiver
Description		user request to set the mirrors tinting automatically
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

4.2.16 Ports referencing StaAutoTinting

Used By	***Reason***
HMI AutoTintRequest	Automatic tinting status to be sent to HMI
TintingManager AutoTintRequest	

PortInterface				
Name (of PortInterface)		StaAutoTinting		
Communication Method		SenderReceiver		
Description		Status of the automatic tinting.		
isService		No (e.g. no BSW service)		
1	***Data Element Name***	Status		
	Data Type	enum		
	Data/Event	Data		
	Symbolic Constants	***Label***	***Value***	***Comment***
		error	0	--
		on	1	--
		off	2	--
	Remarks	Length (bit) = 7 (see data type) Recommendations: Init value=0, can invalidate=No		

4.2.17 Ports referencing ReqTintingValue

Used By	***Reason***
Tinting value manager	Used to set the tinting value.
Tinting manager	

PortInterface			
Name (of PortInterface)		ReqTintingValue	
Communication Method		SenderReceiver	
Description		Deliveres the desired value for mirror tinting.	
isService		No (e.g. no BSW service)	
1	***Data Element Name***	Status	
	Data Type	t_percentage	
	Physical Interpretation	***Offset***	0
		Unit	%
	Physical Values	***Min Value***	0

		Max Value	100
		Resolution	1
	Data/Event	Data	
	Remarks	Length (bit) = 7 (see data type) Recommendations: Init value=0, can invalidate=Yes	

4.2.18 Ports referencing CmdMirrorTinting

Used By	Reason, comment
Tinting Actuator	Used to send the mirror tinting actuation command.
Tinting manager	

PortInterface					
Name (of PortInterface)		CmdMirrorTinting			
Communication Method		SenderReceiver			
Description		Command for tinting to the actuator component.			
isService		No (e.g. no BSW service)			
1	***Data Element Name***	Request			
	Data Type	record			
	Record elements	***Element***	***Label***	***Value***	***Comment***
		mirror_ type (of type enum)	no_mirror	0	--
			driver_miror	1	--
			passen-ger_mirror	2	--
			interior_mirror	3	--
		position (of type t_position)	***Offset***	0	
			Unit	Percent	
			Min Value	0	
			Max Value	1000 1001 special use (un-known, stop, don't care)	
			Resolution	0.1	
	Data/Event	Data			
	Remarks	Length (bit) = 4+10 Recommendations: Init value={no_mirror, 0}; can invalidate=No			

4.2.19 Ports referencing DiagMirrorTinting

Used By	Reason, comment
Tinting Actuator	Used to send the mirror tinting actuation command for diagnostic purposes.

PortInterface					
Name (of PortInterface)		DiagMirrorTinting			
Communication Method		SenderReceiver			
Description		Command for tinting to the actuator component for diagnostic purposes.			
isService		No (e.g. no BSW service)			
1	***Data Element Name***	Request			
	Data Type	record			
	Record elements	***Element***	***Label***	***Value***	***Comment***
		mirror_ type (of type enum)	no_mirror	0	--
			driver_miror	1	--

			passen-ger_mirror	2	--
			interior_mirror	3	--
		position (of type t_position)	***Offset***	0	
			Unit	Percent	
			Min Value	0	
			Max Value	1000 1001 special use (unknown, stop, don't care)	
			Resolution	0.1	
		Element	***Label***	***Value***	***Comment***
		diagnostic_type (of type t_onoff)	off	0	diagnostic inactive
			on	1	diagnostic active
	Data/Event	Data			
	Remarks	Length (bit) = 2+10+1 Recommendations: Init value={no_mirror, 0, off}; can invalidate=No			

4.2.20 Ports referencing StaMirrorTinting

Used By	***Reason***
Tinting actuator Tinting status	Status of tinting actuator
TintingManager AutoTintRequest	

PortInterface	
Name (of PortInterface)	StaMirrorTinting
Communication Method	SenderReceiver
Description	Current status of tinting as feedback in order to control the tinting command to the actuator component.
isService	No (e.g. no BSW service)
1 ***Data Element Name***	Status
Data Type	t_onoff
Data/Event	Data
Remarks	Length (bit) = 1 (see data type) Recommendations: Init value=0, can invalidate=No

4.2.21 Ports referencing StaTintingFault

Used By	***Reason, comment***
Tinting Manager	Operational/defective feedback of the mirror tinting actuator
Tinting Actuator	

PortInterface	
Name (of PortInterface)	StaMirrorFault
Communcation Method	SenderReceiver
Description	Used as feedback from an tinting actuator component to indicate that the mirror tinting 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 the mirror tinting is broken and should not be actuated. Note:The interface is not used for detailed diagnostics.
isService	No (e.g. no BSW service)
1 **Data Element Name**	status

	Data Type	enum		
	Symbolic Constants	**Label**	**Value**	**Comment**
		operational	0	--
		defective	1	e. g. mirror tinting not working
	Data/Event	Data		
	Remarks	Length (bit) = 1 Recommendations: Init value=operational, can invalidate=No		

4.3 Types Declarations

4.3.1 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	**Label**	**Value**	**Description**
	off	0	--
	on	1	--

4.3.2 t_percentage

Name	t_percentage	
Lower-Limit	0	
Upper-Limit	100	
Description	The type t_percentage represents a percentage value.	
Symbolic Constants	**Offset**	0
	Factor	1
	Unit	%

4.3.3 t_position

Name	t_position	
Lower-Limit	0	
Upper-Limit	100	
Description	The type t_position represents a percent value specially for the adjustment positions.	
Symbolic Constants	**Offset**	0
	Factor	0.1
	Unit	%

4.3.4 t_profile

Name	t_profile		
Lower-Limit	0		
Upper-Limit	15		
Description	The type t_profile represents an enumeration type for the selection of the mirror profile.		
Symbolic Constants	**Label**	**Value**	**Description**
	no_profile	0	--
	profile1	1	--

	profile2	2	--
	...	...	--
	profile14	14	--
	profile15	15	--

5 Specification of Ports

5.1 Auto Adjust User Request Manager

The Auto Adjust User Request Manager combines and prioritizes all the information from above and requests mirror movements and storing of memory positions.

5.1.1 Require Ports

Port Name	**PassengerMirrorManAdjustmentHMI**			
Reference to interface	**ReqManAdjustPanel**			
Direction	required (by SW-C Auto Adjust User Request Adapter)			
Description	Used to receive the user's manual movement request for the passanger's mirror.			
Data Elements	*Name*	*Init Value*	*can invalidate*	*Further attributes*
	request	**idle**	**no**	--

Please note that this port is representative for many other ports **[ManAdjustment]HMI** used to request a manual movement of a mirror denoted by [ManAdjustment].

Port Name	**PassengerMirrorAutoAdjustmentHMI**			
Reference to interface	**ReqAutoAdjustPanelMirror**			
Direction	required (by SW-C Auto Adjust User Request Adapter)			
Description	Used to receive the user's request to move the passenger's mirror to a memorized position.			
Data Elements	*Name*	*Init Value*	*can invalidate*	*Further attributes*
	request	**{no_profile, off}**	**no**	--

Port Name	**Driver Profile**			
Reference to interface	**StaProfilePerson**			
Direction	required (by SW-C Auto Adjust User Request Adapter)			
Description	Used to receive the driver's profile in order to auto adjust the mirror position according to the memorized position related to the driver's profile.			
Data Elements	*Name*	*Init Value*	*can invalidate*	*Further attributes*
	status	**no_profile**	**no**	--

Port Name	**AccessRemoteKey**
Reference to interface	**ReqAccessRemoteKey**
Direction	required (by SW-C Auto Adjust User Request Adapter)
Description	Used to receive the request from remote key to store current position in case of lock request and to adjust to a memorized position in case of an unlock request.

Data Elements	Name	Init Value	can invalidate	Further attributes
	request	**idle**	**no**	--

Port Name	AccessKeyless			
Reference to interface	ReqAccessKeyless			
Direction	required (by SW-C Auto Adjust User Request Adapter)			
Description	Used to receive the request from keyless access to store current position in case of lock request and to adjust to a memorized position in case of an unlock request.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	**{undefined; idle}**	**no**	--

Port Name	Gear			
Reference to interface	StaGear			
Direction	required (by SW-C Auto Adjust User Request Adapter)			
Description	Used to receive the status of the gearbox (in order to move down to outer mirror angle to provide the best view to the curb when the reverse gear is selected).			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status		**no**	--

Port Name	MirrorActionState			
Reference to interface	StaMirrorAction			
Direction	required (by SW-C Auto Adjust User Request Adapter)			
Description	Used to receive from the mirror adjust manager the type of action in progress.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	status	**idle**	**no**	--

5.1.2 Provide Ports

Port Name	AutoMove			
Reference to interface	ReqMirrorAutoMove			
Direction	provided (by SW-C Auto Adjust User Request Adapter)			
Description	Used to provide the automatic move request to the mirror adjustment manager.			
Data Elements	Name	Init Value	can invalidate	Further attributes
	request	**{no_mirror, no_profile}**	**no**	---

Port Name	StorePosition			
Reference to interface	ReqStoreMirrorPos			
Direction	provided (by SW-C Auto Adjust User Request Adapter)			
Description	Used to provide the position store request to the mirror adjustment manager.			

Data Elements	*Name*	*Init Value*	*can invalidate*	*Further attributes*
	request	**{no_mirror, no_axis, no_profile, 0};**	**no**	--

5.2 Mirror Adjust Manager

The Mirror Adjust Manger implements the core of the mirror movement functionality.

5.2.1 Require Ports

Port Name	**PassengerMirrorManAdjustmentHMI**			
Reference to interface	**ReqManAdjustPanel**			
Direction	required (by SW-C Mirror Adjust Manager)			
Description	Used to receive the user's manual movement request for the passanger's mirror.			
Data Elements	*Name*	*Init Value*	*can invalidate*	*Further attributes*
	request	**idle**	**no**	--

Please note that this port is representative for many other ports **[ManAdjustment]HMI** used to request a manual movement of a mirror denoted by [ManAdjustment].

Port Name	**AutoMove**			
Reference to interface	**ReqMirrorAutoMove**			
Direction	required (by SW-C Mirror Adjust Manager)			
Description	Used to receive the automatic move request from the auto adjust user request manager.			
Data Elements	*Name*	*Init Value*	*can invalidate*	*Further attributes*
	request	**{off}**	**no**	--

Port Name	**StorePosition**			
Reference to interface	**ReqStoreMirrorPos**			
Direction	required (by SW-C Mirror Adjust Manager)			
Description	Used to receive the position store request from the auto adjust user request manager.			
Data Elements	*Name*	*Init Value*	*can invalidate*	*Further attributes*
	request	**{no_mirror, no_axis, no_profile, 0};**	**no**	--

Port Name	**MirrorPosition**			
Reference to interface	**StaMirrorPos**			
Direction	required (by SW-C Mirror Adjust Manager)			
Description	Used to receive from the actuator the mirror's actual position.			
Data Elements	*Name*	*Init Value*	*can invalidate*	*Further attributes*

	status	**0**	**no**	--

Port Name	**MirrorMove Status**			
Reference to interface	**StaMirrorMove**			
Direction	required (by SW-C Mirror Adjust Manager)			
Description	Used to receive from the actuator the mirror's movement status.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	status	**off**	**no**	--

Port Name	**RestorePosition**			
Reference to interface	**StaMirrorPosMemory**			
Direction	required (by SW-C Mirror Adjust Manager)			
Description	Used to receive the mirror's memorized position from a specific profile.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	status	**{no_mirror, no_axis, no_profile, 0};**	**no**	--

Port Name	**OperationMode**			
Reference to interface	**StaOperationMode**			
Direction	required (by SW-C Mirror Adjust Manager)			
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 invali-date***	***Further attributes***
	status	**off**	no	--

Port Name	**EnergyMgnt**			
Reference to interface	**StaEnergyMgnt**			
Direction	required (by SW-C Mirror Adjust Manager)			
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 invali-date***	***Further attributes***
	status	**full_funct**	no	--

Port Name	**Speed**			
Reference to interface	**VehicleSpeed**			
Direction	required (by SW-C Mirror Adjust Manager)			
Description	Used to unfold the mirrors at a specific speed or to inhibit memory-folding if speed is above a specific value.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	value	**0**	no	--

Port Name	**Voltage**			
Reference to interface	**BatteryVoltage**			
Direction	required (by SW-C Mirror Adjust Manager)			
Description	Used for voltage compensation, disable mirror movement in case of over/under voltage.			
Data Elements	*Name*	*Init Value*	*can invali-date*	*Further attributes*
	value	**25,5**	no	--

Port Name	**CentralLocking**			
Reference to interface	**StaCentralLocking**			
Direction	required (by SW-C Mirror Adjust Manager)			
Description	Used for folding outer mirrors when locking.			
Data Elements	*Name*	*Init Value*	*can invali-date*	*Further attributes*
	status	**{undefined, no_trigger_ source}**	no	--

Port Name	**Mirror Fault**			
Reference to interface	**StaMirrorFault**			
Direction	required (by SW-C Mirror Adjust Manager)			
Description	Operational / defective feedback from an mirror actuator.			
Data Elements	**Name**	**Init Value**	**can invali-date**	**Further attributes**
	status	**operational**	no	--

5.2.2 Provide Ports

Port Name	**MirrorActionState**			
Reference to interface	**StaMirrorAction**			
Direction	provided (by SW-C Mirror Adjust Manager)			
Description	Used to provide to the auto adjustment user request adapter the type of action in progress.			
Data Elements	*Name*	*Init Value*	*can invali-date*	*Further attributes*
	status	**idle**	**no**	--

Port Name	**MirrorMove**			
Reference to interface	**ReqMirrorMove**			
Direction	provided (by SW-C Mirror Adjust Manager)			
Description	Used to provide the movement request to the mirror actuator adapter.			
Data Elements	*Name*	*Init Value*	*can invali-date*	*Further attributes*
	request	**0**	**no**	--

Port Name	**StorePosition**
Reference to interface	**ReqStorePosMemory**
Direction	provided (by SW-C Mirror Adjust Manager)
Description	Used to provide the store request for the position of the mirrors to the mem-

	ory position 'profile'.			
Data Elements	***Name***	***Init Value***	***can invalidate***	***Further attributes***
	request	**{no_mirror, no_axis, no_profile, 0};**	**no**	--

5.2.3 Ports to AUTOSAR-Services

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

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.

5.3 Mirror Actuator Adapter

The Mirror Actuator Adapter knows about the electrical and mechanical constraints of each single axes of the mirror. It converts requests from the Mirror Adjust Manager to commands to each single axes (actuators). It compares the current position of the mirror axis with the desired position.

5.3.1 Require Ports

Port Name	**MirrorMove**			
Reference to interface	**ReqMirrorMove**			
Direction	required (by SW-C Mirror Actuator Adapter)			
Description	Used to receive the movement request from the mirror adjust manager.			
Data Elements	***Name***	***Init Value***	***can invalidate***	***Further attributes***

	request	**0**	**no**	--

Port Name	**MirrorPosition**			
Reference to interface	**StaMirrorPos**			
Direction	required (by SW-C Mirror Actuator Adapter)			
Description	Used to receive from the actuator the mirror's actual position.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	status	**0**	**no**	--

Port Name	**MirrorMove Status**			
Reference to interface	**StaMirrorMove**			
Direction	required (by SW-C Mirror Actuator Adapter)			
Description	Used to receive from the actuator the mirror's movement status.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	status	**off**	**no**	--

Port Name	**Mirror Fault**			
Reference to interface	**StaMirrorFault**			
Direction	required (by SW-C Mirror Actuator Adapter)			
Description	Operational / defective feedback from an mirror actuator.			
Data Elements	**Name**	**Init Value**	**can invali-date**	**Further attributes**
	status	**operational**	no	--

5.3.2 Provide Ports

Port Name	**MirrorAction**			
Reference to interface	**CmdMirrorMove**			
Direction	provided (by SW-C Mirror Actuator Adapter)			
Description	Used to provide the movement request to the mirror actuator.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	command	**off**	**no**	--

5.4 Mirror Actuator

The Mirror Actuator is the software representation of a mechanical actuator. It is able to move one axis of the mirror to a certain direction. It also detects the position of the axis

5.4.1 Require Ports

Port Name	**MirrorAction**			
Reference to interface	**CmdMirrorMove**			
Direction	required (by SW-C Mirror Actuator)			
Description	Used to receive the movement request to the mirror actuator adapter.			
Data Elements	*Name*	*Init Value*	*can invali-date*	*Further attributes*
	command	**off**	**no**	--

5.4.2 Provide Ports

Port Name	**MirrorPosition**			
Reference to interface	**StaMirrorPos**			
Direction	provided (by SW-C Mirror Actuator)			
Description	Used to provide to the mirror actuator adapter and mirror adjust manager the mirror's actual position.			
Data Elements	*Name*	*Init Value*	*can invali-date*	*Further attributes*
	status	**0**	**no**	--

Port Name	**MirrorMove Status**			
Reference to interface	**StaMirrorMove**			
Direction	provided (by SW-C Mirror Actuator)			
Description	Used to provide to the mirror actuator adapter and mirror adjust manager the mirror's movement status.			
Data Elements	*Name*	*Init Value*	*can invali-date*	*Further attributes*
	status	**off**	**no**	--

Port Name	**Mirror Fault**			
Reference to interface	**StaMirrorFault**			
Direction	provided (by SW-C Mirror Actuator)			
Description	Operational / defective feedback for the the mirror actuator adapter and mirror adjust manager.			
Data Elements	*Name*	*Init Value*	*can invali-date*	*Further attributes*
	status	**operational**	no	--

5.4.3 Diagnostic Ports

The port defined below is supposed to be connected to a Central Diagnostic SW-C (above RTE !). Please note that the port “Mirror 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	**DiagMirrorMove**			
Direction	required (by SW-C Mirror actuator [mirror][axis])			
Description	Command to move the axis in a certain direction for diagnostic purposes.			
Data Elements	*Name*	*Init Value*	*can invalidate*	*Further attributes*
	request	**{no_mirror; no_axis; idle; off}**	no	--

Due to it's diagnostic capabilities each Mirror 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 Mirror actuator [mirror][axis])			
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 Mirror actuator [mirror][axis])			
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 Mirror actuator [mirror][axis])			
Description	Actuator doesn't consume power although it's 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 Mirror actuator [mirror][axis])			
Description	Actuator consumes excessive power (connected to batt.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 Mirror actuator [mirror][axis])

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 invalidate*	*Further attributes*
	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	--

5.5 Mirror Memory Adapter

The Mirror Memory Adapter is responsible for the actual memory position storage and recall. This may be local or global storage depending on the personalisation control strategy. External temperature information is used to compensate storage and restorage of mirror positions.

5.5.1 Require Ports

Port Name	**StorePosition**			
Reference to interface	**ReqStorePosMemory**			
Direction	required (by SW-C Mirror Memory Adapter)			
Description	Used to receive the store request for the position of the mirrors to the memory position 'profile'.			
Data Elements	*Name*	*Init Value*	*can invalidate*	*Further attributes*
	request	**{no_mirror, no_axis, no_profile, 0};**	**no**	--

Port Name	**OutTemperature**			
Reference to interface	**StaOutsideTemperature**			
Direction	required (by SW-C Mirror Memory Adapter)			
Description	Used to receive the external environment temperature for compensating the stored and restored mirror profile positions.			
Data Elements	*Name*	*Init Value*	*can invalidate*	*Further attributes*

	status	**0**	**no**	--

5.5.2 Provide Ports

Port Name	**RestorePosition**			
Reference to interface	**StaMirrorPosMemory**			
Direction	provided (by SW-C Mirror Memory Manager)			
Description	Used to provide the mirror's memorized position from a specific profile.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	status	**{no_mirror, no_axis, no_profile, 0};**	**no**	--

5.6 Tinting Value Manager

The Tinting Value Manager calculates the desired tinting value using the outside brightness information.

5.6.1 Require Ports

Port Name	**OutsideBrightness**			
Reference to interface	**StaMirrorOutsideBrightness**			
Direction	required (by SW-C Tinting Value Manager)			
Description	Used to receive the incoming brightness from a source outside of the vehicle.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	status	**0**	**yes**	--

5.6.2 Provide Ports

Port Name	**TintingValue**			
Reference to interface	**ReqTintingValue**			
Direction	provided (by SW-C Tinting Value Manager)			
Description	Used to provide the desired value for mirror tinting.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	request	**0**	**yes**	--

5.7 Tinting Manager

The Tinting Manager implements the core of the mirror tinting functionality.

5.7.1 Require Ports

Port Name	**TintingValue**			
Reference to interface	**ReqTintingValue**			
Direction	required (by SW-C Tinting Manager)			
Description	Used to receive the desired value for mirror tinting.			
Data Elements	*Name*	*Init Value*	*can invalidate*	*Further attributes*
	request	**0**	**no**	--

Port Name	**MirrorPosition**			
Reference to interface	**StaMirrorPos**			
Direction	required (by SW-C Tinting Manager)			
Description	Used to receive to the mirror's actual position.			
Data Elements	*Name*	*Init Value*	*can invalidate*	*Further attributes*
	status	**0**	**no**	--

Port Name	**MirrorMove Status**			
Reference to interface	**StaMirrorMove**			
Direction	required (by SW-C Tinting Manager)			
Description	Used to receive the mirror's movement status.			
Data Elements	*Name*	*Init Value*	*can invalidate*	*Further attributes*
	status	**off**	**no**	--

Port Name	**Tinting Status**			
Reference to interface	**StaMirrorTinting**			
Direction	required (by SW-C Tinting Manager)			
Description	Used to receive the current status of tinting as feedback in order to control the tinting command to the actuator component.			
Data Elements	*Name*	*Init Value*	*can invalidate*	*Further attributes*
	status	**deactivated**	**no**	--

Port Name	**OperationMode**			
Reference to interface	**StaOperationMode**			
Direction	required (by SW-C Tinting Manager)			
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	**EnergyMgnt**
Reference to interface	**StaEnergyMgnt**
Direction	required (by SW-C Tinting Manager)
Description	The information is required whether all, basic or only keep-alive compo-

	nents/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 invali-date***	***Further attributes***
	status	**full_funct**	no	--

Port Name	**Voltage**			
Reference to interface	**BatteryVoltage**			
Direction	required (by SW-C Tinting Manager)			
Description	Used for voltage compensation, disable mirror movement in case of over/under voltage.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	value	**25,5**	no	--

Port Name	**AutoTintRequest**			
Reference to interface	**ReqAutoTinting**			
Direction	required (by SW-C Tinting Manager)			
Description	Used to receive the user's request to set mirrors tinting automatically.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	request	**off**	no	--
Port Name	**Tinting Fault**			
Reference to interface	**StaTintingFault**			
Direction	required (by SW-C Tinting Manager)			
Description	Operational / defective feedback from an mirror actuator.			
Data Elements	**Name**	**Init Value**	**can invali-date**	**Further attributes**
	status	**operational**	no	--

5.7.2 Provide Ports

Port Name	**AutoTint Status**			
Reference to interface	**StaAutoTinting**			
Direction	provided (by SW-C Tinting Manager)			
Description	Used to provide the status of the automatic tinting.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	status	**0**	**no**	--

Port Name	**TintingActivation**			
Reference to interface	**CmdMirrorTinting**			
Direction	provided (by SW-C Tinting Manager)			
Description	Used to provide the mirror tinting activation command.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	command	**0**	**no**	--

5.8 Tinting Actuator

5.8.1 Require Ports

Port Name	**TintingActivation**			
Reference to interface	**CmdMirrorTinting**			
Direction	required (by SW-C Tinting Actuator)			
Description	Used to provide the mirror tinting activation command.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	command	**0**	**no**	--

5.8.2 Provide Ports

Port Name	**Tinting Status**			
Reference to interface	**StaMirrorTinting**			
Direction	provided (by SW-C Tinting Manager)			
Description	Used to provide the current status of tinting as feedback in order to control the tinting command to the actuator component.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	status	**deactivated**	**no**	--

Port Name	**Tinting Fault**			
Reference to interface	**StaTintingFault**			
Direction	provided (by SW-C Tinting Manager)			
Description	Operational / defective feedback for the tinting manager.			
Data Elements	**Name**	**Init Value**	**can invali-date**	**Further attributes**
	status	**operational**	no	--

5.8.3 Diagnostic Ports

The port defined below is supposed to be connected to a Central Diagnostic SW-C (above RTE !). Please note that the port “Mirror 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	**DiagMirrorTinting**			
Direction	required (by SW-C Tinting Actuator)			
Description	Used to send the mirror tinting actuation command for diagnostic purposes.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	request	**{no_mirror; 0; idle; off}**	no	--

Due to it's diagnostic capabilities each Mirror Tinting 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 Tinting Actuator)			
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 invali-date***	***Further attributes***
	status	**--**	no	--

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

Port Name	**OpenLoad**			
Reference to interface	**DiagnosticMonitor** (see AUTOSAR_Services.doc)			
Direction	client (by SW-C Tinting Actuator)			
Description	Actuator doesn't consume power although it's activated.			
Data Elements	***Name***	***Init Value***	***can invali-date***	***Further attributes***
	status	**--**	no	--

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

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

Port Name	**OpenLoadShortToGnd**
Reference to interface	**DiagnosticMonitor** (see AUTOSAR_Services.doc)
Direction	client (by SW-C Tinting Actuator)

Description	Error is either OpenLoad or ShortToGnd			
Data Elements	*Name*	*Init Value*	*can invali-date*	*Further attributes*
	status	**--**	no	--

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

5.9 Central Locking

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

5.9.1 Require Ports

There is NO information from Mirror required by Central Locking.

5.9.2 Provide Ports

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

StaProfilePerson
AccessKeyless
AccessRemoteKey
StaCentralLocking

6 APPENDIX

6.1 Inter Domain Interfaces

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

6.1.1 StaOperationMode

Interface				
Name (of Interface)		StaOperationMode		
Communcation Method		SenderReceiver		
Description		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. This interface should be used to control which functionalities / components are active in the vehicle. All states have to be implemented in all cars, but not necessarily used.		
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		

6.1.2 VehicleSpeed

Interface			
Name (of Interface)		VehicleSpeed	
Communcation 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		

6.1.3 BatteryVoltage

Interface				
Name (of Interface)		BatteryVoltage		
Communcation 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.		

6.1.4 StaEnergyMgmt

Interface				
Name (of Interface)		StaEnergyMgmt		
Communcation 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.		

6.1.5 StaGear

Interface	
Name (of Interface)	StaGear
Communcation Method	SenderReceiver
Description	This interface provides the selected gear. This status is provided by the gear box. The selected gear information has to be filtered to e. g. not see in the status the rear position while moving the gear shift lever from park position over rear position to drive position.
isService	No (e.g. no BSW service)
1 ***Data Element Name***	Status

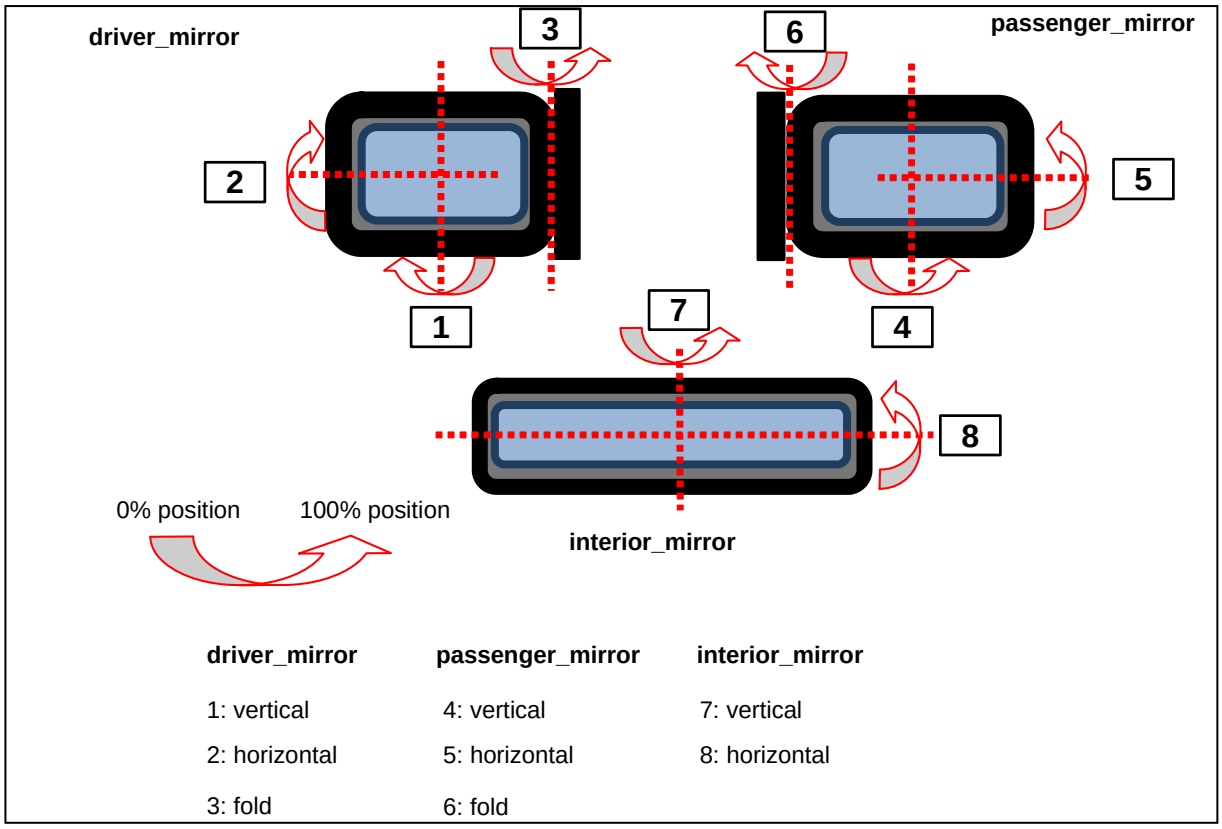
	Data Type	enum		
	Symbolic Constants	***Label***	***Value***	***Comment***
		parking_position	0	--
		neutral_position	1	--
		reverse_position	2	--
		drive_position	3	--
		gear_1	4	--
		gear_2	5	--
		gear_3	6	--
		gear_4	7	--
		gear_5	8	--
		gear_6	9	--
		gear_7	10	--
		gear_8	11	--
		reserved_2	12	--
		reserved_3	13	--
		reserved_4	14	--
		invalid	15	--
	Data/Event	Data		
	Remarks	Recommended init value: neutral-position		

6.1.6 StaOutsideTemperature

Interface			
Interface Name		StaOutsideTemperature	
Comm. Method		Sender Receiver	
Description		This interface provides the information about the current outside temperature of the car.	
isService		No	
1	***Data Element Name***	value	
	Data Type	numeric	
	Physical Interpretation	***Offset***	-50
		Unit	°C
	Physical Values	***Min Value***	-50
		Max Value	75
		Resolution	0.5
	Data/Event	Data	
	Remarks	Recommended init value: 0	

7 Glossary

Regarding mirrors and mirror axes, see below.



driver_mirror	passenger_mirror	interior_mirror
1: vertical	4: vertical	7: vertical
2: horizontal	5: horizontal	8: horizontal
3: fold	6: fold	