Document Title	Safety Use Case Example
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Version	641

Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.3.1

Document Change History

Date	Release	Changed by	Change Description
2017-12-08	4.3.1	AUTOSAR Release Management	• Editorial changes
2016-11-30	4.3.0	AUTOSAR Release Management	• Editorial changes
2015-07-31	4.2.2	AUTOSAR Release Management	• Initial Release

Disclaimer

Table of Contents

1 Introduction

This document shows major analysis steps of an exemplary system using AUTOSAR from a functional safety point of view. The example used within the following document bases upon the AUTOSAR guided tour example Front Light Management. Additional constraints were added whenever needed for analysis or concept discussion.
The present report intends to

1. Build up a relevant use case in an AUTOSAR environment for a functional safety analysis.
2. Provide an example to discuss and verify safety related concepts within AUTOSAR.
3. Identify improvement potential with respect to functional safety aspects in the current AUTOSAR specifications and methodology.
4. Propose improvements or additional concepts in AUTOSAR, required for safety architectures.
5. Create input for the concept: “Safety Related Extension for Methodology and Templates”.
6. Provide a guideline for safety analyses on top of the AUTOSAR methodology.

The example is prepared in context of the ISO 26262 requirements, but is focused on the AUTOSAR relevant parts. Although it could be seen as a basic guideline for safety analysis on top of AUTOSAR methodology, it just touches the main topics roughly. Further details (e.g. a detailed list of software safety requirements or examples for safety analysis measures) could be added in a next development step.

This example covers aspects like

- The functional safety concept
- The technical safety concept on system level
- The technical safety concept on ECU level
- Safety aspects on AUTOSAR basic software level.

Note: Implementation constraints do not necessarily match with an existing implementation.

2 Item Description

The item chosen for this example primarily equals the AUTOSAR guided tour example Front Light Management. Whenever additional definitions were needed, information supporting clarification of safety related topics was added.

The current example's scope focuses on a very limited functional part of the front light, namely the low beam feature. All other light functionalities e.g. parking orientation light, fog lights, etc. are excluded. In exception, daytime running lights are named as a possible fallback solution and therefore are integrated in the following figures. However, details of controlling day time running lights are not part of this example.

All vehicle level parts contributing to a Front Light Management will be considered here as "the system" (see Figure 1). This includes the Front Light Management ECU as well as relevant sensors, actuators, display and powering parts facilitating the Front Light Management.

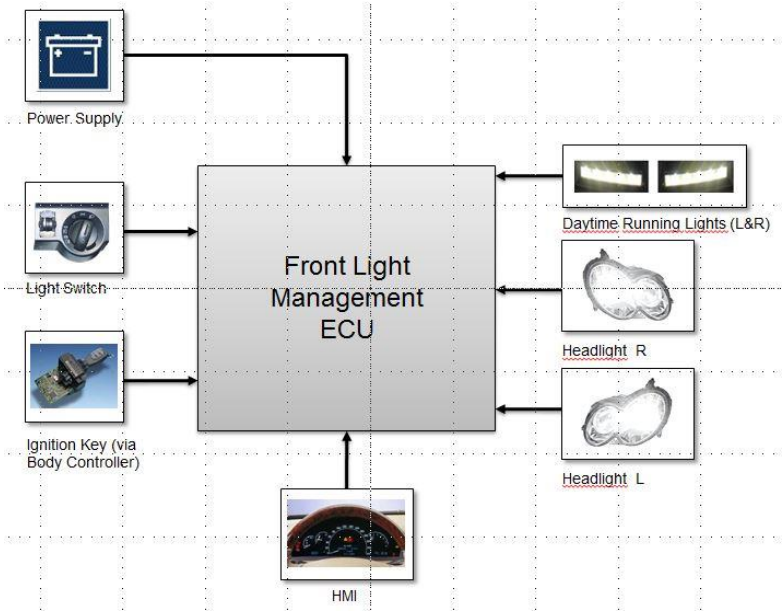

Figure 1: Front Light Management and System Overview

2.1 Functional Behavior

The general feature of a low beam function is to illuminate the roadway in the dark. In addition the low beam informs other road users about a vehicle approaching. Activation/Deactivation conditions are summarized in the following table.

Function	Operating Elements	Turn-on Conditions	Turn-off Conditions
Low Beam	Light switch (LS)	CL15 ON **AND** Light switch ON	Light switch OFF **OR** CL15 OFF

Table 1: Operating Elements and Functional Behavior

The low beam can be turned on by a light switch while clamp 15 (CL15) is activated via Ignition Key. Any malfunction of low beam lights shall be indicated to the driver.

As additional function, the daytime running light is available as part of the Front Light Management system. Furthermore, all relevant normative regulations available for the low beam function are to be applied.

Based on this nominal function the following functions and associated functional requirements are derived:

1. detection of low beam light request
    a. The Front Light Manager shall evaluate the Ignition Key position.
    b. The Front Light Manager shall read the LS switch position
2. evaluation of low beam light request
    a. The Front Light Manager shall evaluate the LS switch status.
    b. Only if the LS switch status changes from OFF to ON the Front Light Manager shall create a switch event (ON).
    c. If the LS switch status changes from ON to OFF the Front Light Manager shall create a switch event (OFF).
3. control of low beam lights
    a. The Front Light Manager shall activate the low beam light, if the Ignition Key position is ON and a light switch event is detected,
    b. The Front Light Manager shall deactivate the low beam light if the Ignition Key position is OFF or a switch event (OFF) is detected.
4. monitoring of low beam light function
    a. The Front Light Manager shall supervise the low beam lights
    b. The Front Light Manager shall indicate failures of the low beam lights, e.g. current failure or bulb failure.
5. activation of daytime running lights
    a. The Front Light Manager shall activate the daytime running lights in case of a low beam lights failure.

2.2 Preliminary Architecture

The following figure shows the assumed system architecture including system elements like

- Front Light Management ECU
- Light Switch (LS)
- Ignition Key (via Body Controller)
- Power Supply
- Headlights, left and right
- Daytime Running Lights, left and right
- HMI

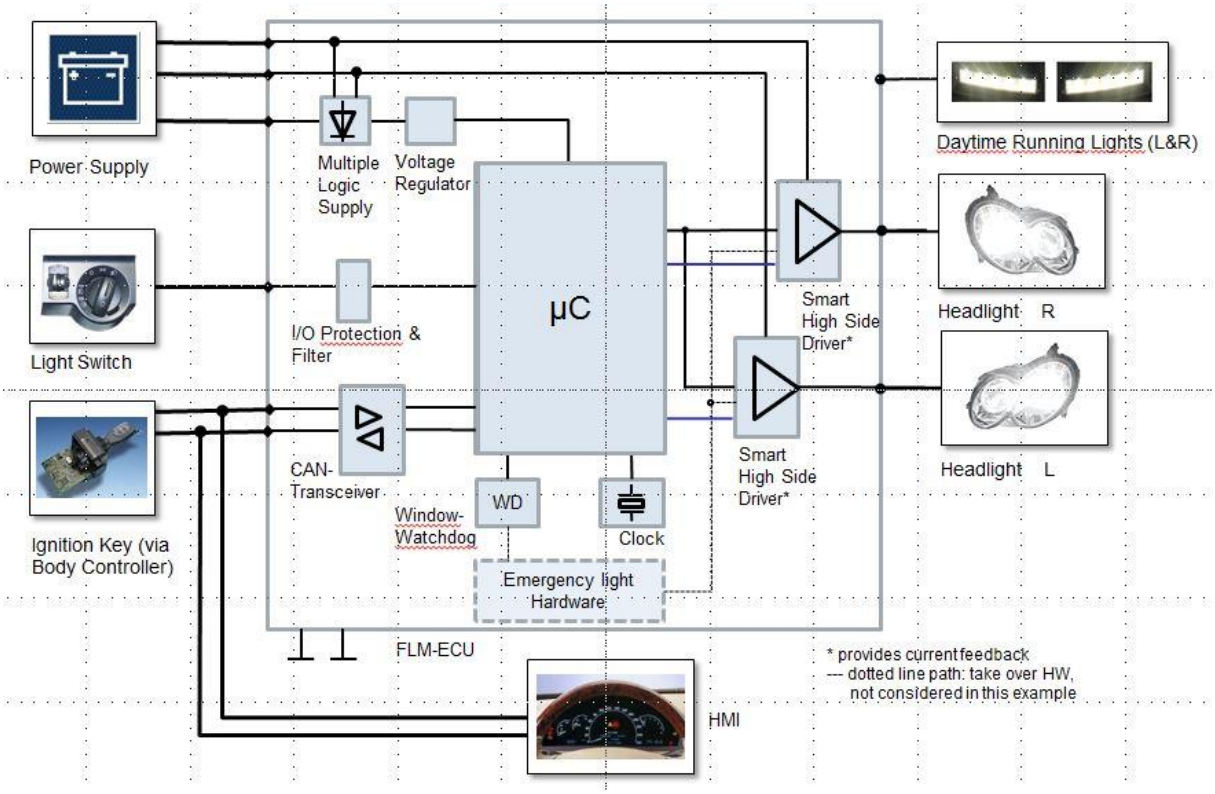

Figure 2: Preliminary Architecture Front Light Management

The technical interfacing of system elements with Front Light Management ECU is assumed as shown in Figure 2 and Table 2.

System Element	Interface to FLM ECU
Light switch position (LS)	DIO
Ignition Key position (CL15) (via Body Controller)	CAN interface
HMI	CAN interface
Head light control left	PWM
Head light control right	PWM
Daytime running lights (L&R)	PWM
Power Supply	Analog

Table 2: Interfaces of FLM ECU

A communication focused view of the technical interfacing is shown in Figure 3.

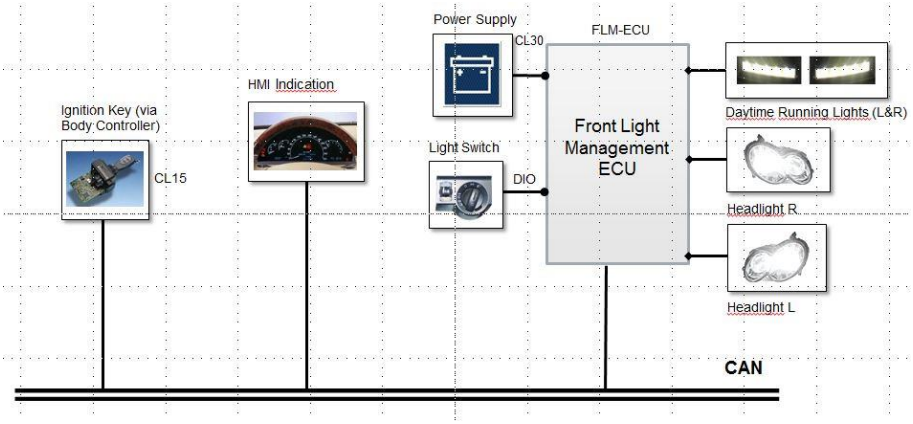

Figure 3: Preliminary Architecture Front Light Management (communication focused view)

2.3 Assumptions and Limitations of Exemplary Safety Analysis

As starting point for the example the following configuration of the system is assumed:

1. Implementation of Front Light Management software on **one ECU**
2. Activation of light via switch which provides **one output** that is a digital I/O.
   Note: Common types of this kind of light switch provide analogue or up to 4 digital outputs with a logic table to validate the status. The intention of this example is to show the data flow through the software architecture. Hardware diagnostics of sensors is not in focus.
3. **Emergency light** functionality (in case of e.g. µC operation failures) is provided by hardware and not intended to use AUTOSAR software parts.
4. All **memory** (volatile and non-volatile) is **protected** against reversible transient faults. It is assumed that mechanisms like ECC are available.
5. Hardware means for **memory partitioning** are available (e.g. MPU)
6. The Front Light Management software is integrated on a system with BSW modules which do not comply with an ASIL rating according to ISO26262.
7. The analysis of failure modes of the microcontroller is performed and safety measures are defined and implemented. This analysis is based on data provided by the supplier e.g. a safety manual and the requirements of ISO26262.

Additionally, subsequent constraints are defined to keep the focus of the example to specific AUTOSAR software safety questions:

1. Assuming that the ECU is working as required:
   a. The ECU is waking up; running and going to sleep correctly (The example does not focus on mode management or state management.).
   b. The necessary communication network is available, up and running correctly (The example does not focus on COM management).
   c. The necessary BSW modules are triggered as required.
2. Failures in the board wiring system, battery or power supply are not considered. Even though the battery is external to the Front Light Management (FLM), its failures will directly affect the FLM and any other vehicle's electronics systems. However, failure in the 12V supply will affect

many functions in the car (breaks, engine etc.) so we assume a vehicle wide solution to handle these failures.

3. Failure of power supply of low beam lights is not considered. Typically there is a separate power supply for left and right low beam.
4. A startup test of the bulbs is assumed to be in place. The design and implementation of it is not part of the example.
5. The light switch signal (LS) is already filtered/de-bounced. Furthermore it is assumed that the signal is provided in an adequate quality to match requirements of ISO 26262 (e.g. required failure rate, process requirements for hardware development see ISO 26262, part 5) (to simplify the example).

3 Safety Concept on Vehicle Level

The following chapter outlines the safety analysis on vehicle level and its result typically created by the OEM and partially provided to a supplier.

Note: This structure of safety analysis cannot be mapped 1 by 1 to the structure in ISO 26262 [1], since the norm does not include the different possible scopes of development, explicitly.

3.1 Outcome of Hazard Analysis and Risk Assessment

For this example, it is supposed the following hazard and attributes were identified as output of a hazard and risk assessment:

Hazard **H1:** *Total loss of low beam (ASIL B)*

This hazard potentially could cause the driver to lose control of the vehicle, leave the road and collide with environmental parts.

Exceptions and Boundary Conditions to H1:

- The loss of low beam is only to be seen as a risk in case of bad viewing conditions (night, fog, etc.).
- The loss of low beam on a curvy, unlighted rural road is evaluated as a most critical situation.
- The loss of only one low beam is not considered to be directly leading to a hazardous situation. However, it is a latent fault and will be included in the concept advisement.

ASIL: The ASIL B rating is based on the severity rate the exposure rate and the controllability identified during hazard analysis and risk assessment.

Safety Goal SG01: Prevent total loss of low beam

Safe State: Low beam activated

Fault Tolerance Time: 500 ms
(Loss of low beam during night drive with activated lights shall be considered.)

Note: Additional hazards could be assumed. However, to limit our example it is assumed to be the only relevant hazard. For the following discussion of AUTOSAR features it seems not to be helpful to include additional risks at this point in time.

3.2 Relevant Failure Modes

The safety goal SG01 can be violated through one or more of the following malfunctions (MF):

- MF01: Failure of the detection of the turn-on/ turn-off conditions of lights
- MF02: Failure of the evaluation and implementation of the light request function which is used to turn lights on
- MF03: Failure of the activated lights

3.3 Functional Safety Concept

The outcome of the function description, the safety goal as well as the top level safety requirements, is a functional safety architecture that will be mapped to one particular vehicle architecture.

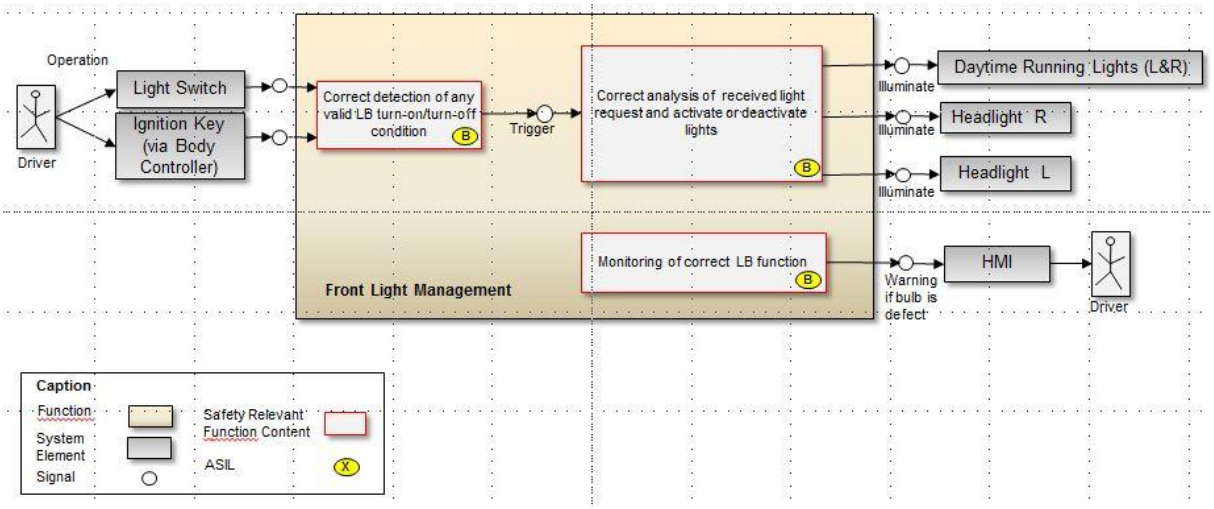

Figure 4: Functional Safety Architecture

3.3.1 FunSafReq01-01:

FLM shall detect any valid turn on condition of low beam correctly. (ASIL B)
Relates to MF01: No LB after a “valid LB lights ON” request.

3.3.2 FunSafReq01-02:

FLM shall verify the validity for any LB lights request received and activate or deactivate the low beam accordingly. (ASIL B)
Relates to MF02: Switching lights off without receiving a “valid LB lights OFF” request

3.3.3 FunSafReq01-03:

FLM shall detect failures of the activated low beam and signal malfunctions to the driver. (ASIL B)
Relates to MF03: Failure of the activated lights

Note:
Valid LB lights ON request => “CL15 is ON & light switch changes from OFF to ON”
Valid LB lights OFF request => “light switch changes from ON to OFF”

3.4 Safety Requirements on Vehicle Level

As part of the vehicle level safety analysis a warning and degradation concept is defined according to the available system architecture. The Technical Safety Requirements on vehicle level are derived from Functional Safety Requirements and assumptions of preliminary architecture (see 2.2) and marked with SysSafReq IDs.

Note: The process of generating Technical Safety Requirements is not part of this document and therefore not described in the following. These vehicle level requirements are used for demonstration of the several requirements levels. They are neither completely analyzed nor mapped from a real implementation.

3.4.1 Warning and Degradation Concept

For the low beam lights the following 2-step degradation concept shall be implemented:

Normal Mode:

- See nominal function within item description (see 2).

Degraded Mode Step 1 - loss of **one** low beam:

- The second low beam still fulfills its function.
- A warning shall be provided to the driver, indicating partial loss of low beam, while low beam is requested but not successfully activated (fault detected).

Degraded Mode Step 2 - loss of **both** low beams:

- Day time running lights shall be activated, whenever low beam is requested but not successfully activated (fault detected).
- A warning shall be provided to the driver, indicating loss of low beam and activation of daytime running lights, while low beam is requested.

Note: This kind of usage of day time running light as fallback is just used as an example to demonstrate a degraded function. This solution is probably not suitable under homologation aspects.

Note: Additional driver assistance functions could be activated in this mode to assist the driver, but this is not part of this example.

3.4.2 Technical Safety Requirements (on Vehicle Level)

The following technical safety requirements are derived within this exemplary analysis based on determination in 3.3:

3.4.2.1 SysSafReq01

The CAN connected Body Controller shall signal Ignition Key clamp 15 status via CAN bus message CL15_01 (CAN-message: CL15_01, CAN signal: CL15ON (Boolean, ‘1’ if clamp 15 is set to on, ‘0’ if clamp 15 is set to off)). (ASIL B)

Note: CAN message details are defined in CAN DB (frequency, inhibit time, signal type) as part of the nominal function.

3.4.2.2 SysSafReq02

The light switch shall signal the state of the switch via digital HW line HW_LB_OFF (0=0V if lights on are requested, 1=5V if lights off are requested). (ASIL B)

3.4.2.3 SysSafReq03

Failures within the light switch shall lead to the digital HW line HW_LB_OFF set to 0. (ASIL B)

3.4.2.4 SysSafReq04

FLM ECU shall ensure that the limits (like voltage and PWM) to power the LB bulbs are kept, while FLM ECU detects that the condition for powering the bulbs is fulfilled (as defined as part of nominal function). (ASIL B)

3.4.2.5 SysSafReq05

While CL15ON==1, FLM ECU shall switch the light off only if HW_LB_OFF ==1 condition is true continuously for 20ms (CAN message: CL15_01; CAN signal: CL15ON Boolean, '1' if clamp 15 is set to on, '0' if clamp 15 is set to off). (ASIL B)

Note: The timing condition to wait for a stable condition for some milliseconds is included to de-bounce the signal value. The particular value of 20ms is taken as experts' experience. Another value could be used instead, as well.

3.4.2.6 SysSafReq06

FLM ECU shall detect circuit failures (bulbs, fuses, wiring-oc/sc) and signal them via CAN (CAN message: LightStatus_01, CAN signal: LBFailure (2 bit, 01 in case left LB fails, 10 in case right LB fails, 11 in case both LB fail)). (ASIL B)

3.4.2.7 SysSafReq07

FLM ECU shall activate both daytime running lights (DRL) if a failure of both LB bulbs is detected continuously for 200ms. (ASIL B)

Note: The FTT budget is allocated the following way: 200ms failure detection time + conservative 200ms time interval until halogen bulb reaches full intensity (failure reaction) + 100ms buffer = 500ms.

3.4.2.8 SysSafReq08

FLM ECU shall use independent circuits to power left and right bulbs such that no single fault can cause a total loss of low beam. (ASIL B)

3.4.2.9 SysSafReq09

HMI shall display bulb outage information according to the signal LBFailure received via CAN (CAN message: LightStatus_01, CAN signal: LBFailure (2 bit, 01 in case left LB fails, 10 in case right LB fails, 11 in case both LB fail)). (ASIL A)

Note: For this measure the FTT is not relevant since it is a measure against latent failures. A relevant time here can be calculated (diagnosis interval). The measure against a double bulb failure (where FTT is relevant) is the activation of the daytime running light. As a measure against latent faults the ASIL is reduced according to ISO 26262-4:2011(E) 6.4.4.4.

3.4.2.10 SysSafReq10

The data transmission via CAN between sender and receiver must be ensured for CAN message: CL15_01 CAN signal: CL15ON Boolean, (‘1’ if clamp 15 is set to on, ‘0’ if clamp 15 is set to off). (ASIL B)

Note: For this measure all relevant failure modes known for data exchange (see ISO26262 part 6) should be considered.

3.4.2.11 SysSafReq11

The data transmission via CAN between sender and receiver must be ensured for CAN message: LightStatus_01, CAN signal: LBFailure (2 bit, 01 in case left LB fails, 10 in case right LB fails, 11 in case both LB fail). (ASIL A)
Comment: For this measure all relevant failure modes known for data exchange (see ISO 26262-6) should be considered.

3.4.2.12 SysSafReq12

FLM ECU shall activate low beam lights if a communication fault regarding message CL15_01 is detected continuously for 200ms.

Note: The FTT budget is allocated the following way: 200ms failure detection time + conservative 200ms time interval until halogen bulb reaches full intensity (failure reaction) + 100ms buffer = 500ms.

3.4.2.13 SysSafReq13

FLM ECU shall

- activate both daytime running lights (DRL) if a communication fault regarding message LightStatus_01 is detected continuously for 200ms and
- provide a text message to the driver like ‘Light system defect’.

3.4.3 Allocation of (Functional) System Safety Requirements

In the next step all system safety requirements need to be allocated to an architectural system element (see Figure 5).

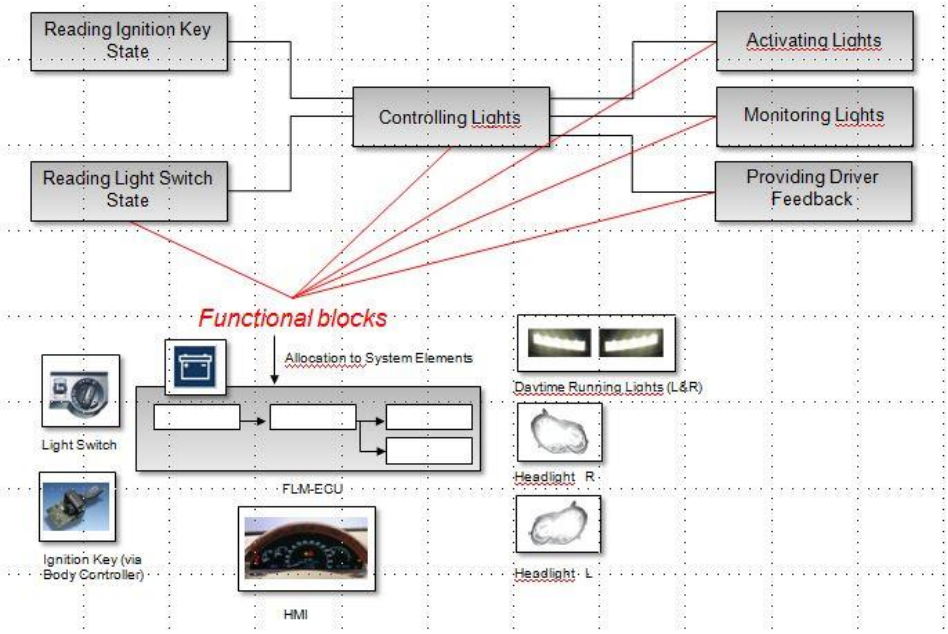

Figure 5: Allocation of Functional Blocks to Preliminary Architecture

3.4.4 Summary of Technical System Safety Requirements (Vehicle Level)

The overall mapping of System Safety Requirements to Functional Safety Requirements (vehicle level) as well as the allocation to system elements is summarized in **Table 3**.

ID	System Safety Requirement	Supports Functional Safety Req. IDs	ASIL	Item
SysSaf Req01	The Body Control Unit shall signal clamp 15 status via CAN bus message CL15_01 (CAN-message: CL15_01, CAN-Signal: CL15ON (Boolean, '1' if clamp 15 is set to on, '0' if clamp 15 is set to off)). *Note: CAN message details are defined in CAN DB (frequency, inhibit time, signal type) as part of the nominal function.*	FunSafReq01-01	ASIL B	BC-ECU
SysSaf Req02	The light switch shall signal the state of the switch via digital HW line HW_LB_OFF (0=0V if lights on are requested, 1=5V if lights off are requested).	FunSafReq01-01	ASIL B	Light Switch
SysSaf Req03	Failures within the light switch shall lead to the digital HW line HW_LB_OFF set to 0.	FunSafReq01-02	ASIL B	Light Switch
SysSaf Req04	FLM ECU shall ensure that the limits (voltage, PWM) to power the LB bulbs are kept, while FLM ECU detects that condition for powering the bulbs is fulfilled (as defined as part of nominal function).	FunSafReq01-02	ASIL B	FLM ECU
SysSaf Req05	While CL15ON==1 FLM ECU shall switch the light off only if HW_LB_OFF ==1 condition is true continuously for 20ms (CAN-message: CL15_01 CAN-Signal: CL15ON Boolean, '1' if clamp 15 is set to on '0' if clamp 15 is set to off). *Note: The timing condition to wait for a stable condition for some milliseconds is included to de-bounce the signal value. The particular value of 20ms is taken as experts' experience. Another value could be used instead, as well.*	FunSafReq01-02	ASIL B	FLM ECU

ID	System Safety Requirement	Supports Functional Safety Req. IDs	ASIL	Item
SysSaf Req06	FLM ECU shall detect circuit failures (bulbs, fuses, wiring-oc/sc) and signal them via CAN (CAN-message: LightStatus_01, CAN-Signal: LBFailure (2 bit, 01 in case left LB fails, 10 in case right LB fails, 11 in case both LB fail.	FunSafReq01-03	ASIL A	FLM ECU
SysSaf Req07	FLM ECU shall activate both daytime running lights (DRL) if a failure of both LB bulbs is detected continuously for 200ms. *Note: The FTT budget is allocated the following way: 200ms failure detection time + conservative 200ms time interval until halogen bulb reaches full intensity (failure reaction) + 100ms buffer = 500ms.*	FunSafReq01-03	ASIL B	FLM ECU
SysSaf Req08	FLM ECU shall use independent circuits to power left and right bulbs such that no single fault can cause a total loss of low beam.	FunSafReq01-02	ASIL B	FLM ECU
SysSaf Req09	HMI shall display bulb outage information according to the signal LBFailure received via CAN (CAN-message: LightStatus_01, CAN-Signal: LBFailure (2 bit, 01 in case left LB fails, 10 in case right LB fails, 11 in case both LB fail)). *Note: For this measure the FTT is not relevant since it is a measure against latent failures. A relevant time here can be calculated (diagnosis interval). The measure against a double bulb failure (where FTT is relevant) is the activation of the daytime running light. As a measure against latent faults the ASIL is reduced according to ISO 26262-4:2011(E) 6.4.4.4.*	FunSafReq01-03	ASIL A	HMI
SysSaf Req10	The data transmission of CL15_01.CL15ON between sender and receiver must be ensured. *Comment: For this measure all relevant failure modes known for data exchange (see ISO 26262-6) should be considered*.	FunSafReq01-02	ASIL B	FLM ECU BC-ECU
SysSaf Req12	FLM ECU shall activate low beam lights if a communication fault regarding message CL15_01 is detected continuously for 200ms. *Note: The FTT budget is allocated the following way: 200ms failure detection time + conservative 200ms time interval until halogen bulb reaches full intensity (failure reaction) + 100ms buffer = 500ms.*	FunSafReq01-02	ASIL B	FLM ECU
SysSaf Req11	The data transmission of LightStatus_01.LBFailure between sender and receiver must be ensured. *Comment: For this measure all relevant failure modes known for data exchange (see ISO 26262-6) should be considered.*	FunSafReq01-03	ASIL A	HMI FLM ECU
SysSaf Req13	FLM ECU shall • activate both daytime running lights (DRL) if a communication fault regarding message LightStatus_01 is detected continuously for 200ms and • provide a text message to the driver like 'Light system defect'.	FunSafReq01-03	ASIL A	HMI FLM ECU

Table 3: Summary of Technical System Safety Requirements

4 Technical Safety Concept on FLM-ECU Level

The following chapter outlines the safety analysis on ECU level and its result typically created by a supplier.

4.1 Assumptions and Limitations on ECU Level

The safety concept on FLM-ECU level bases on the following assumptions:

1. According to AUTOSAR methodology, BSW is running cyclically.
2. HW signals are read cyclically and are available as client/server signals in RAM of the I/O hardware abstraction (HWA).
3. Incoming CAN messages generate an interrupt.
4. The ECU is waking up, running and going to sleep correctly (no focus here on mode management or state management).
5. The necessary communication network is available, up and running correctly, (no focus here on COM management).
6. The necessary BSW modules are triggered as required.
7. RTE is used as medium for transmission of data and messages and does not perform any scaling (transformation of data).
8. Daytime running lights are technically in the same way connected like head lights but on a separately dedicated SPI.
   Note: In this case a cross-wiring could help for diagnostics of broken wiring harness, if left headlight cable controls right side daytime running light and vice versa.

4.2 Safety Goals to be Fulfilled

The relevant safety goals are defined in 3.1.

4.3 Relevant System Safety Requirements

The relevant system safety requirements are listed in 3.4.4.

4.4 Overview of Concept on ECU Level

This block diagram shows the elements on ECU level involved in fulfilling the safety goal including the ASIL allocation to input, processing, output, and warning elements.

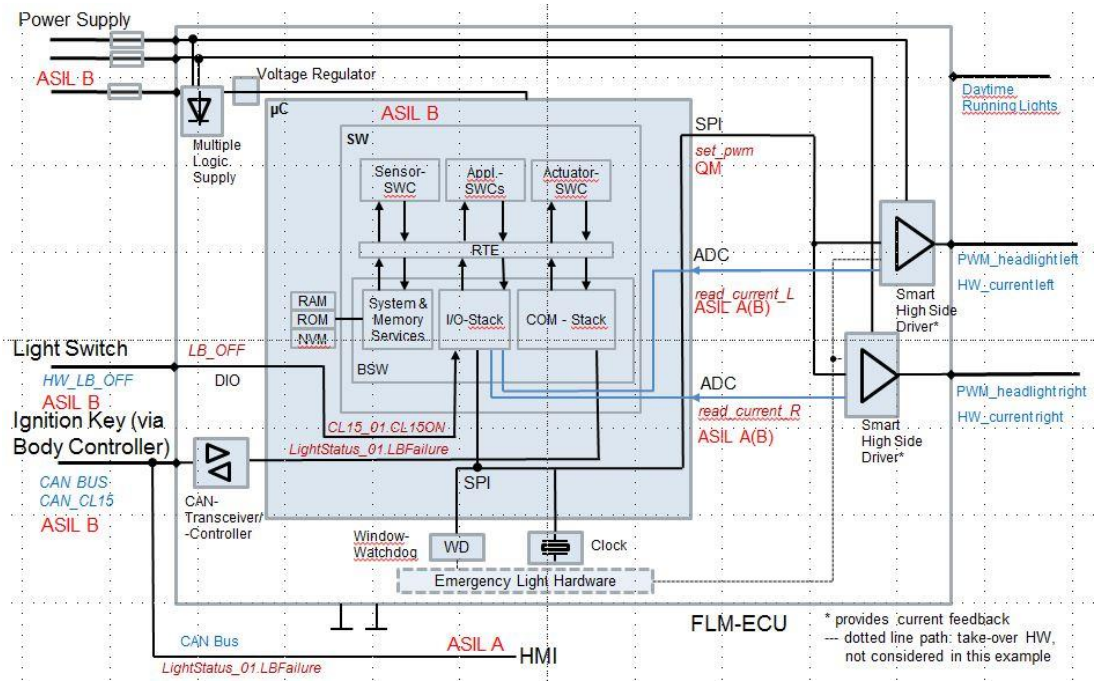

Figure 6: Block Diagram on ECU Level

The functionality “Headlight” (ASIL B) is decomposed into two separate headlights (2x ASIL A(B)) to provide redundancy as shown in Figure 6.

The assumption in the current example is: “a loss of set_pwm does not directly lead to “*Total loss of low beam (H1)”* Therefore the common SPI “bus” for controlling the headlights is QM rated.
“Set_pwm” is used to create a rectangular pulse wave whose pulse width is modulated to control headlights via smart high side driver (see Figure 6 rightmost side)

Rationale:

1. Startup self-diagnosis detects faults of the SPI connection with sufficient coverage with feedback to the driver (during stand still, before any hazardous driving situation).
2. The “read back” path read_current_R and read_current_L provides feedback whether a light request was transferred successfully or not.
3. If the “read back” path shows a difference to the activation path request, an appropriate, safe reaction (in this case Day Time Running Lights set to ON if LS ON was detected), has to be implemented, independent of the set_pwm command.
4. So, the “activation path SPI” can be implemented as QM path.
5. Furthermore, the LB activation path controls the activation, and the feedback path (ASIL B) informs before any dangerous (darkness) situation about the success. A single corrupted set_pwm message (each LB gets a separate request) would affect only one of both head lights

4.5 Requirements on ECU Level

The following list summarizes safety requirements on ECU level. Based on these software requirements additional requirements on basic software modules have to be derived. This step is not provided in detail within this document.
The granularity of the derived SW requirements will stay very limited to keep the example understandable. Safety related software components shall be developed according to the highest ASIL rating of assigned requirements.

ECU Safety Requirement ID	ECU Safety Requirement	Functional Block	Domain	ASIL	ECU Functionality	Related AUTOSAR Component	System Safety Requirement
	The CAN message CAN BUS CAN_CL15 shall be received correctly.	Reading	*ECU*	B			SysSafReq05
ECU01	The correct reading of the (physical) CAN BUS CAN_CL15 shall be ensured.	Reading	HW	QM(B)	Reading Ign.-Key State	CAN-Transceiver/-Controller	SysSafReq05
ECU02	The correct transformation of CAN BUS CAN_CL15 to the logical CL15_01 message shall be ensured.	Reading	HW + SW driver	QM(B)	Reading Ign.-Key State	CAN-Transceiver/-Controller	SysSafReq05
ECU03	The correct routing of the CL15_01 message through the AUTOSAR BSW/RTE shall be ensured. The signal CL15_01.CL15ON is to be extracted and provided to the Application-SWCs properly.	Reading	SW	QM(B)	Reading Ign.-Key State	COM Stack (Can Driver, CaniF, Com, PduR) RTE	SysSafReq05
ECU04	The ECU shall detect any potential communication faults affecting the signal CL15ON that could lead to a violation of the safety goal.	Diagnostics	HW SW	B(B)	Controlling Lights	RTE, Application-SWC, Com Stack (Can Driver, CaniF, Com, PduR) RTE	SysSafReq05
	The ECU shall determine the status of the light switch input HW_LB_OFF correctly.	Reading	*ECU*	B			SysSafReq05
ECU06	The correct reading of the HW_LB_OFF input shall be ensured.	Reading	SW	QM(B)	Reading Light Request	I/O Stack (Dio & Port)	SysSafReq05
ECU07	The correct configuration of the HW_LB_OFF input port and pin shall be ensured.	-	SW	QM(B)	Reading Light Request	I/O Stack (Dio & Port)	SysSafReq05
ECU08	The correct transformation of the HW_LB_OFF input to the logical LB_OFF signal shall be ensured.	Reading	HW SW	QM(B)	I/O Filtering (Not part of example, see assumptions)	I/O Stack (I/O HW abstraction)	SysSafReq05
ECU09	The correct routing of LB_OFF through the AUTOSAR BSW/RTE shall be ensured.	Reading	SW	QM(B)	Reading Light Request	I/O Stack (I/O HW abstraction), RTE	SysSafReq05
ECU10	The ECU shall detect potential faults affecting LB_OFF that could lead to a violation of the safety goal.	Diagnostics	SW	B(B)	Controlling Lights	Application-SWC, RTE, Sensor-SWC, I/O Stack (I/O HW abstraction)	SysSafReq05 SysSafReq06
	The Application SWCs shall evaluate the light request and initiate the powering of the bulbs as specified.	Processing		B			SysSafReq04
ECU12	The Application-SWC shall determine the LB_OFF and CL15ON status as specified.	Processing	SW	B	Reading Light Request	Application-SWC	SysSafReq05 SysSafReq04
ECU13	The Application-SWC shall evaluate the light request conditions based on LB_OFF and CL15ON and their timing as specified.	Processing	SW	B	Controlling Lights	Application-SWC	SysSafReq04 SysSafReq05
ECU14	The Application-SWC shall • set or reset the light on command (Lights_ON) based on the LB_OFF and CL15ON evaluation results or if any fault is detected • set the light on command, if a communication fault of CL15_01 message is detected continuously for more than 200ms or • set the light on command, if a fault on LB_OFF is detected continuously for more than 200ms.	Processing	SW	B	Controlling Lights	Application-SWC	SysSafReq05, SysSafReq12

ECU Safety Requirement ID	ECU Safety Requirement	Functional Block	Domain	ASIL	ECU Functionality	Related AUTOSAR Component	System Safety Requirement
ECU15	The Application-SWC shall activate both daytime running lights (DRL_ON) if a failure of both LB bulbs is detected continuously for 200ms (read_current_L, read_current_R)	Processing	SW	B	Controlling Lights	Application-SWC	SysSafReq07, SysSafReq12
ECU30	When the bulbs are powered, the Actuator-SWC shall read back current and provide the status of the bulbs (read_current_L, read_current_R).	Monitoring	SW	A(B) A(B)	Monitoring Lights	Application-SWC	SysSafReq06
	The correct powering of the bulbs according to the specification shall be ensured.	Controlling		B			SysSafReq04
ECU16	The correct powering of the bulbs according to the light request and the specification are to be signaled via set_pwm command.	Controlling	SW	QM(B)	Activating Lights	Actuator-SWC	SysSafReq04
ECU17	The correct routing of the set_pwm request to the µC SPI output shall be ensured.	Controlling	SW	QM(B)	Activating Lights	Actuator-SWC, RTE, I/O HW abstraction, SPI driver	SysSafReq04
ECU18	The correct routing of the µC set_pwm request to the high side drivers and the correct transformation of the set_pwm request to the physical outputs PWM_headlight left, PWM_headlight right shall be ensured.	Controlling	HW	QM(B)	Activating Lights	-	SysSafReq04
ECU19	The ECU shall provide 2 independent HW paths for powering the bulbs, one for the left bulb (PWM_headlight left), and one for the right bulb (PWM_headlight right).	Controlling	HW	A(B) A(B)	Activating Lights	-	SysSafReq08
	The ECU shall supervise the status of the bulbs and inform the driver in case of a failure.	Monitoring		B			
ECU20	When the bulbs are powered, the Application-SWC shall evaluate the status of the bulbs (read_current_L, read_current_R).	Monitoring	SW	A(B) A(B)	Monitoring Lights	Application-SWC	SysSafReq06
ECU21	Detected faults shall be signaled via CAN BUS LBFailure.	Controlling Driver Feedback	SW	A	Providing Driver Feedback	Application-SWC, RTE, Com Stack	SysSafReq06
ECU22	The ECU shall provide 2 independent channels for measuring the current HW_current left, HW_current right of each of the bulbs.	Monitoring	HW	A(B) A(B)	Monitoring Lights	-	SysSafReq08
ECU23	The ECU shall detect faults of the bulb health measurement path (µC HW, ECU HW, vehicle wiring harness) in correspondence of the fault tolerant time (FTT).	Diagnostics	HW SW	B	Monitoring Lights	I/O HW abstraction, SPI driver	SysSafReq06
ECU24	The correct routing of bulb health measurement values read_current_L, read_current_R through the AUTOSAR BSW/RTE shall be ensured.	Monitoring	SW	QM(B)	Monitoring Lights	I/O Stack (I/O HW abstraction), driver, RTE	SysSafReq06
ECU25	The ADC shall convert the measured current to read_current_L, read_current_R	Monitoring	HW SW	A(B) A(B)	Monitoring Lights	ADC driver	SysSafReq06
	SWCs internal above RTE and communication						
ECU26	The correct data exchange (timing and content) between the SW-components shall be ensured.	Processing	SW	B	Controlling Lights	Application-SWC	
ECU27	The transmission of CL15_01.CL15ON between sender and receiver must be ensured.	Reading	SW	B	Reading Ign.-Key State	Application-SWC	SysSafReq10
ECU28	The transmission of LightStatus_01.LBFailure between sender and receiver must be ensured.	Controlling Driver Feedback	SW	A	Providing Driver Feedback	Application-SWC	SysSafReq11
ECU29	The correct transformation of the logical PWM-l-signal to the SPI BUS message shall be ensured.	Controlling	SW	QM(B)	Activating Lights	Application-SWC	SysSafReq04
ECU30	When the bulbs are powered, the Actuator-SWC shall read back current and provide the status of the bulbs (read_current_L, read_current_R).	Monitoring	SW	A(B) A(B)	Monitoring Lights	Application-SWC	SysSafReq06

Table 4: Requirements on ECU Level

4.6 ECU Functionality

The following chapter is structured according to the functional blocks (see Figure 5), e.g. Reading Light Request, Reading Ignition Key State, etc. It describes signal interfacing at the boundary of the ECU, outlining how the input signals are transformed from the physical hardware input to the logical software input and from the logical software output to the hardware output.

4.6.1 Reading Light Switch State

Physical input: HW_LB_OFF
(On HW level, it is currently realized as single digital input.)
Logical input: LB_OFF
Involved SW modules: DIO Driver/Handler, RTE, SwitchEvent (Sensor-SWC).

4.6.2 Reading Ignition Key State (via Body Controller)

Physical input: CAN BUS CAN_CL15
Logical input: CL15_01 (ASIL B)
Involved SW modules: CAN driver and interface, PDU Router, AUTOSAR COM, RTE, Light Request (Application SWC).

4.6.3 Activating Lights (physical)

4.6.3.1 Low Beam Lights

Physical outputs: PWM_headlight left, PWM_headlight right (QM)
(The high side drivers create the PWM signal according to the SPI request.)
Logical outputs: Set_pwm
Involved SW modules: HeadLight (Actuator-SWC control path), RTE, SPI-Driver/Handler

4.6.3.2 Daytime Running Lights

Physical outputs: Daytime running lights
Logical outputs: Not part of the example
Involved SW modules: Not part of the example

4.6.4 Monitoring Lights

Physical inputs: HW_current left, HW_current right (ASIL A(B)).
Logical inputs: Read_current_L, read_current_R
Involved SW modules: HeadLight (Actuator-SWC read back path), RTE, ADC-Driver/Handler.

4.6.5 Providing Driver Feedback

Physical output: CAN bus (ASIL A)
Note: This ASIL is allocated to the signal content only and not to the whole CAN bus or bus interface.
Logical output: LBFailure
Involved SW modules: CAN driver and interface, PDU Router, AUTOSAR COM, RTE, FLM (Application-SWC).

4.6.6 Controlling Lights (logical)

The logical part of controlling lights is covered by SW only. For details see SW architecture in Figure 8. Logical input is provided according to the overview in 4.6.1 - 4.6.5.

Physical input: none
Physical output: none
Involved SW modules: Switch-Event (Sensor-SWC), LightRequest (Application-SWC), FLM (Application-SWC), Headlight (Actuator-SW-C) RTE

5 SW Architecture and SW Safety Requirements

Two objectives shall be supported within this chapter: First it shall provide an overview of the abstract software system, including its main software components (SWCs) and the involved AUTOSAR BSW modules in a hierarchical structure (Figure 7, Figure 8). This will include static aspects such as interfaces and data paths of all main software components as well as dynamic aspects such as process sequences and timing behavior, documented within a communication diagram (Figure 9).
Second, it will support the identification of the potential failure modes for each defined safety requirement on ECU level and how they can be detected by adequate safety measures. Finally, in case of already existing safety measures in AUTOSAR a reference will be provided.

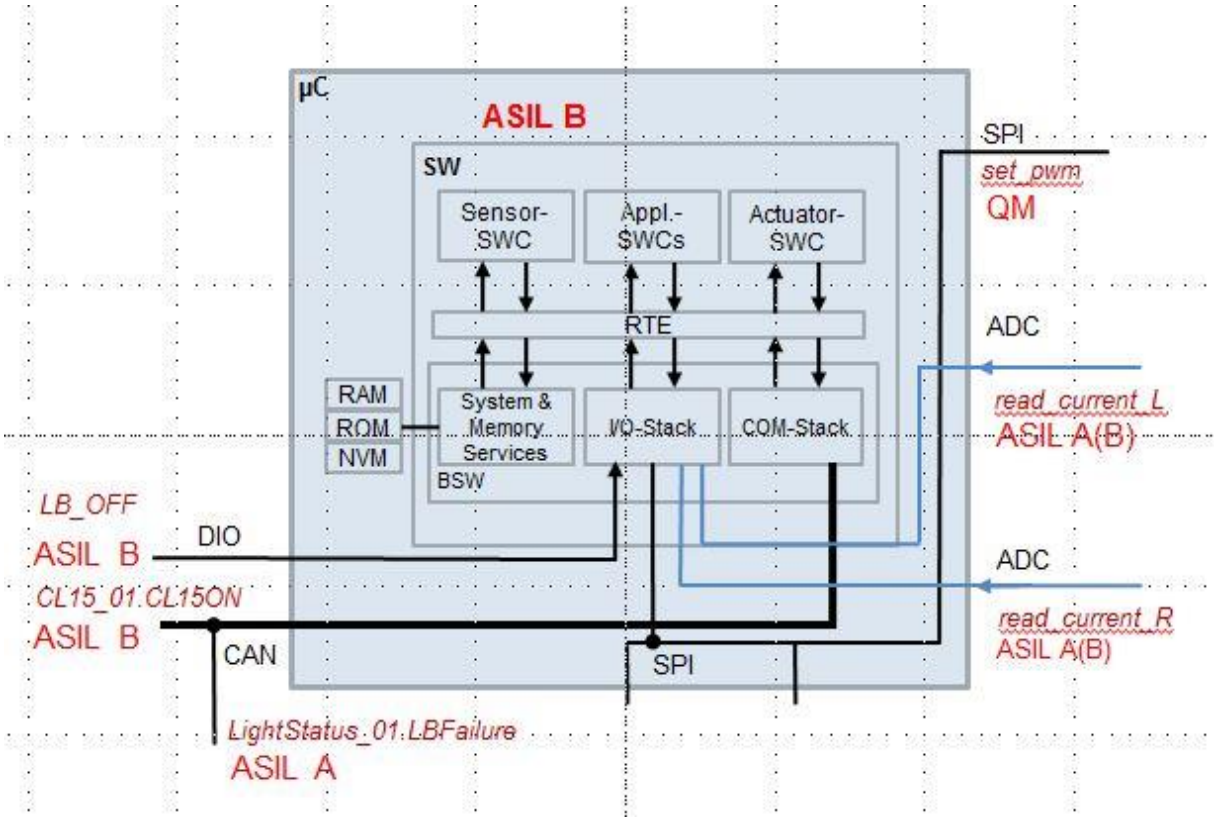

Figure 7: Block Diagram on SW Architecture Level

5.1 Software Architecture

The following picture shows the example of related software components integrated into a standard AUTOSAR architecture (see [2]).

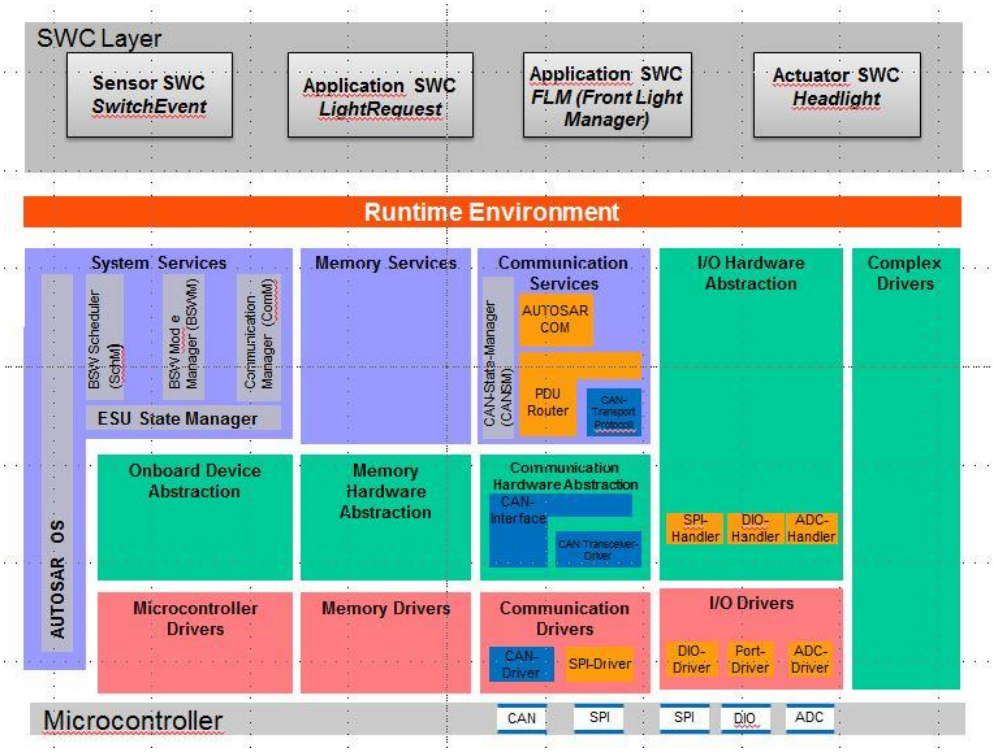

Figure 8: SW Components and Involved AUTOSAR BSW Modules

FLM shall be integrated in an existing ECU with AUTOSAR SW architecture, using an AUTOSAR BSW-Stack.
It shall consist of 4 SW components:

- Sensor SWC (SwitchEvent)
- Application SWC (LightRequest)
- Application SWC (FLM)
- Actuator SWC (Headlight)

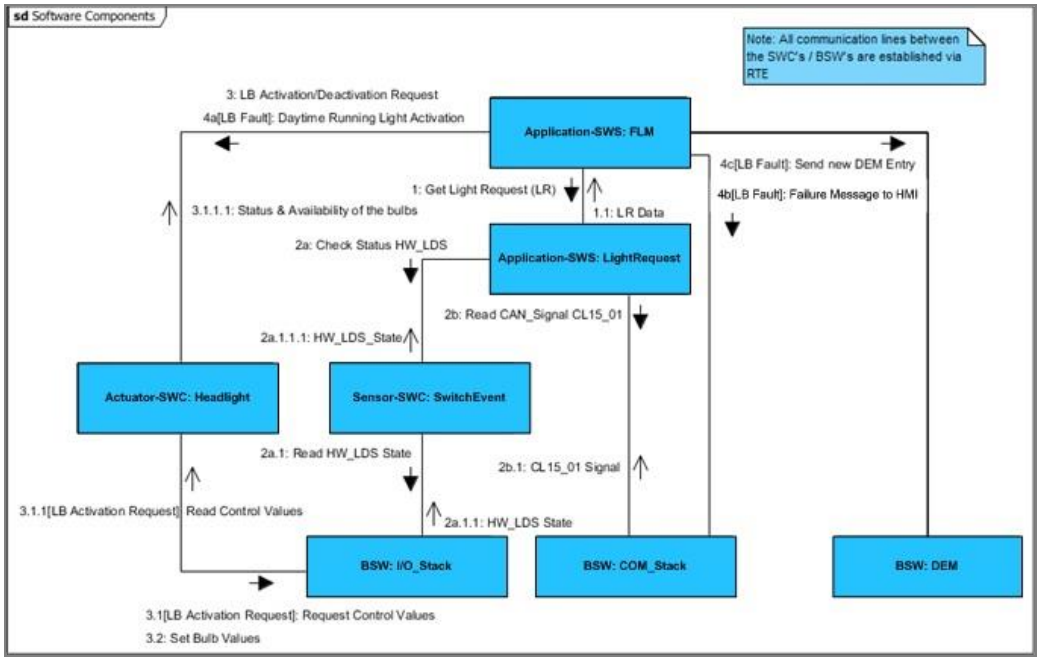

Figure 9: SWC Communication-Diagram via RTE

5.1.1 Software Components

The following software components are defined to provide the headlight function

1. *SwitchEvent*
   a. Sensor SWC:
   b. event triggered, on request of LightRequest *(Check Status HW_LB_OFF; HW_LB_OFF_State)*
   c. check_switch(): read HW_LB_OFF *(Read HW_LB_OFF)*
   d. stores state of HW_LB_OFF *(HW_LB_OFF_State)*

2. *LightRequest*
   a. Application SWC
   b. LightRequest: checks status of HW_LB_OFF (check_switch) *(Check Status HW_LB_OFF; HW_LB_OFF_State)*
   c. reads relevant CAN signals of message CL15_01 *(Read CAN_Signal CL15_01; CAN_Signal CL15_01)*
   d. sends on demand a light request according to the nominal function table to "FLM" *(LR Data)*

3. *FLM (Front Light Manager)*
   a. Application SWC
   b. monitors the status and availability of the bulbs (read-back path) *(Status & Availability of the bulbs)*
   c. cyclic request for getting light request information from "LightRequest", read "Light Request" data *(Get Light Request)*
   d. sends a signal to HMI (Head Unit) in case of failure of bulbs *([LB Fault] Failure Message to HMI)*
   e. when the conditions are fulfilled, low beam bulb activation request is sent to „Headlight"*(LB Activation/Deactivation Request)*
   f. sends daytime running light activation request to „Headlight" in case both low beam bulbs are defect *([LB Fault] Daytime Running Light Activation)*
   g. is responsible for entries in DEM *([LB Fault] Send new DEM Entry)*

4. *Headlight*
   a. Actuator SWC
   b. is able to provide the current light status(from 4.d.) and the availability of the bulbs on request *(Status & Availability of the bulbs)*
   c. controls the front lights
   d. checks the read-back path cyclically (physical details of the lamps are available) only if FLM already requested headlight activation *([LB Activation Request] Request Control Values; ([LB Activation Request] Read Control Values)*
   e. ensures limitation of bulb control (voltage, PWM) *(Set Bulb Values)*

5.1.2 RTE Runtime Environment

The RTE is the realization of the interface of the AUTOSAR Virtual Function Bus (VFB). The RTE provides the infrastructure services that enable communication between AUTOSAR software-components as well as acting as the means by which AUTOSAR software-components access basic software modules including the OS and communication service.

5.1.3 AUTOSAR BSW View

The following picture and description is focused on communication and signal flow from and to related software components.

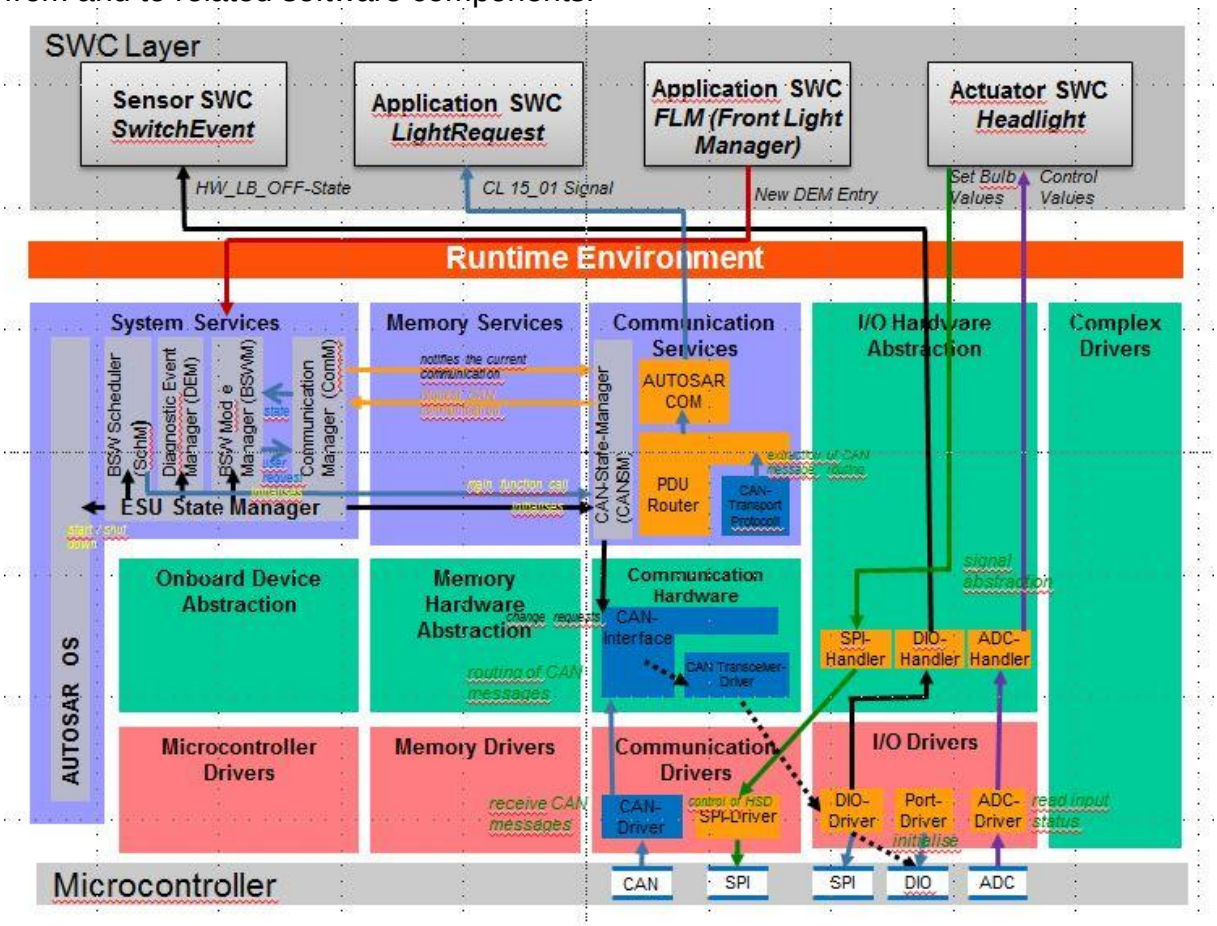

Figure 10: Communication Channels via AUTOSAR BSW Modules

5.1.3.1 Basic Software Components

The following basic software modules were identified as relevant with respect to the architecture and function of the example:

- COM Manager
- AUTOSAR COM
- CAN State Manager
- PDU Router
- CAN Interface
- CAN Driver
- CAN Transceiver Driver
- SPI Driver /Handler
- DIO and Port Driver /Handler
- ADC-Driver /Handler
- ECU State Manager
- BSW Manager
- AUTOSAR OS

5.1.4 General Overview of BSW Function

The main functionality of AUTOSAR BSW modules which is used in this example is provided as a rough overview within the following subchapters.
Comment: BSW function is not described in detail and not complete. For further information please use the AUTOSAR specification.

5.1.4.1 COM Manager

The ComM module controls basic software modules relating to communication and not software components or runnable entities.
The ComM module collects the bus communication access requests from communication requestors and coordinates the bus communication access requests.

- The ComM module receives any communication request from SW-C
- The ComM module uses the API of the CAN State Manager module to request communication modes of CAN Network.

5.1.4.2 AUTOSAR COM

AUTOSAR COM is located between RTE and PDU-Router. It provides different features that affect the communication (e.g. packing/unpacking PDUs, data interface for RTE etc.)

- Monitors the received signals (signals timeout) filter mechanisms for incoming signals.

5.1.4.3 CAN State Manager

The CAN State Manager module uses the API of the CAN Interface module to control the operating modes of the CAN Controllers and CAN Transceivers assigned to the CAN Networks.
The CAN State Manager module notifies the current communication mode of its CAN Networks to the Communication Manager (ComM) module.

5.1.4.4 PDU Router

The PDU Router module provides services for routing of I-PDUs using the following module types: Communication interface modules (e.g. Com, LinIf, CanIf, CanNm, FrIf and FrNm).

- Provides routing of PDUs between different abstract communication controllers and upper layers; Scale of the PDU Router is ECU specific (down to no size if e.g. only one communication controller exists); Provides transport protocol routing on-the-fly.

5.1.4.5 CAN Interface

The CAN Interface module is located between CAN Driver and upper communication service layer. It represents the interface to the services of the CAN Driver and the upper communication modules of the AUTOSAR COM stack.

5.1.4.6 CAN Driver

The CAN Driver performs the hardware access and offers a hardware independent API to the upper layers. The only upper layer with access to CAN Driver is the CAN Interface.

- Initiates transmissions.
- Calls the functions of the CAN Interface module for notifying events.
- Controls the behavior and state of the CAN Controllers.

5.1.4.7 CAN Transceiver Driver

The CAN Transceiver driver abstracts the CAN Transceiver hardware.

- It offers a hardware independent interface to the higher layers.
- It abstracts from the ECU layout by using APIs of MCAL layer to access the CAN Transceiver hardware.

5.1.4.8 SPI Driver / SPI Handler

The SPI Handler/Driver provides services for reading from and writing to devices connected via SPI busses.

- It provides access to SPI communication to several users (e.g. EEPROM, Watchdog, I/O ASICs).
- It also provides the required mechanism to configure the on chip SPI peripheral.

5.1.4.9 DIO and Port Driver / Handler

The DIO Driver provides services for reading and writing to/from (DIO Channels (Pins), DIO Ports, DIO Channel Groups).
The DIO Driver works on pins and ports which are configured by the PORT driver for this purpose.
For this reason, there is no configuration and initialization of this port structure in the DIO Driver.

5.1.4.10 ADC Driver

- Provides measurement values (current reading) to the SW.

5.1.4.11 ECU State Manager

The ECU Manager module is a basic software module that manages common aspects of ECU states. Specifically, the ECU Manager module:

- Initializes and de-initializes the OS, the SchM and the BswM as well as some basic software driver modules.
- Configures the ECU for SLEEP and SHUTDOWN when requested.
- Manages all wakeup events on the ECU.
- Coordinates multi cores of an ECU.

5.1.4.12 BSW Manager

Its responsibility is to arbitrate mode requests from application layer SW-Cs or other BSW modules based on simple rules, and perform actions based on the arbitration result.

- Organizes mode handling and mode related interaction of SW-Cs and the BSW modules.
- Monitors changes of the ECU state machine.

5.1.4.13 Autosar OS

- Manages the scheduling of tasks and events.
- Manages the data flow between different tasks.
- Provides features for monitoring and error handling.
- Controls protection features (timing protection/memory protection).

5.2 Failure Modes

ISO 26262 [1] distinguishes between two different causes for failures:

a. random hardware failures that can occur unpredictably during the lifetime of a hardware element and that follow a probability distribution, and
b. systematic failures which are related in a deterministic way to a certain cause, that can only be eliminated by a change of the design or of the manufacturing process, operational procedures, documentation or other relevant factors.

Note: The focus in this chapter will be on systematic software failures because of the general scope of AUTOSAR.

5.2.1 HW Failure Modes

HW failure modes are not part of SW safety analysis. Therefore they are not further detailed here in this AUTOSAR context.

For details see ISO 26262-5:2011(E) ANNEX B and D [1].

5.2.2 SW Failure Modes

For reference of software failure modes see ISO 26262-6:2011(E) Annex D [1]. To handle the results and reduce duplicate entries in the analysis results, the failure modes are clustered into 4 categories – “data integrity, initialization & configuration data”, “data exchange”, “timing & control flow”, “data processing” (see Table 5).

Generic Failure Modes	Potential Cause(s) of Failure				
	Data Integrity		Data Exchange	Timing & Control Flow	Data Processing
	Memory	Initialization & Configuration Data			
Corruption of memory content Aspect "DataIntegrity: *simple memory corruption* Aspect "Init&ConfigData" *already applied initialization can be corrupted on e.g. the peripheral or the configuration "source" data can be already corrupted*	x	x			
Memory partitioning fault	x				
Memory access fault	x				
Incorrect addressing of transmitted data			x		
Blocking access to a communication channel			x		
Corruption of transmitted data *not data in RAM but "transmitted data" disturbed on the wire or while transmission between SW elements (via RTE)*			x		
Delay of transmitted data			x		
Inconsistency of transmitted data			x		
Incorrect transmitted data value (above or below range)			x		
Incorrect sequence of transmitted data			x		
Insertion of transmitted data			x		
Masquerade of transmitted data			x		
Missing transmitted data / Loss of transmitted data			x		
Repetition of transmitted data			x		
Erroneous data processing (SW logic failure)					x
Incomplete execution due to unexpected termination				x	
Execution timing incorrect, executes too slow or too fast or too early or late, incorrect frequency				x	
No execution				x	
Non-terminating calculation				x	

Table 5: Overview of Failure Mode Clustering

5.2.2.1 Data Integrity

Data Integrity summarizes all failure modes related to corrupted **memory** or **initialization and configuration data**.

Data Integrity - Memory covers all failure modes considering corruption of software data at one particular memory address (register, global or static variables, and program code).

Data Integrity - Initialization & Configuration Data also covers specific configuration data that is not applied at all or misapplied to e.g. a peripheral.

5.2.2.2 Data Exchange

Data Exchange covers all failure modes considering data transmission between sender and receiver (Intra-ECU: SWC, BSW, CDD (Inter-ECU: SWC))

Note: The initial physical (CAN/FlexRay) Bus based meaning is extended onto the VFB of AUTOSAR. Therefore, it covers in this view also intra-ECU communication, meaning SW-C to SW-C messages via RTE.

5.2.2.3 Timing & Control Flow

Timing and Control Flow cover all failure modes considering execution timing and sequence order.

5.2.2.4 Data Processing

Data Processing covers all failure modes considering logical software failures like e.g. wrong implementation of any algorithms, filters etc.

5.3 Software Aspects and Potential Failure Modes

Currently no pure software safety requirements are derived from ECU-level safety requirements within this document.
Here only software safety mechanisms for certain failure modes are described.

The following failure mode assignment is based on the clustering described in 5.2.2. For all listed ECU level requirements potential failure modes are identified and some related, often implemented safety measures are shown. This overview shall illustrate the relation of failure modes and safety measures as well as usage of safety measures in a safety concept. Furthermore a mapping to specified AUTOSAR safety measures is included. For details how to use these AUTOSAR mechanisms see [3]. Those points with no mapping to existing AUTOSAR mechanisms or with identified limitations could be used to identify topics for further possible improvement of AUTOSAR specification (see 6).

Note: If the following tables include a n/a entry this means that the selected aspect is not relevant in regard to the analyzed requirement.

Note: The following analysis result is structured according to the related input signal. Therefore, the IDs of linked ECU level requirements are not following a strict order.

5.3.1 Analysis of ECU02
The correct transformation of CAN BUS CAN_CL15 to the logical CL15_01 message shall be ensured.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
o					n/a	n/a	n/a
	x				Wrong/missing initialization or unwanted re-configuration of HW will lead to unpredictable CAN HW behavior	1. ECC(random HW faults, no protection against SW failures) 2. CRC (detect modification of the source data or even of the applied configuration in the Peripheral) 3. Duplication of the source configuration data(compare it before it will be applied) 4. HW-supported protection mechanisms (MPU, MMU, register protection): protection against re-initialization 5. WdgM (maybe to check whether initialization took place only once)	MPU NOT inside BSW(BSW partitioning not specified, especially OS and MCAL require full rights, partitioned BSW implementations are proprietary) CRC-Lib
		o			n/a	n/a	n/a
			x		Interrupt mode: unexpected long/often locked interrupts or wrong interrupt prioritization can lead to “ignored” CAN “data available” triggers Polling mode: The poll frequency/timing can be affected by wrong OS scheduling/task prioritization configuration, unexpected preemption etc.	internal watchdog a) direct reset b) “top level interrupt” which cannot be blocked by “locked interrupts” (machine dependent) external watchdog direct reset	WdgM
				o	n/a	n/a	n/a

Table 6: Analysis of ECU02

5.3.2 Analysis of ECU03

The correct routing of the CL15_01 message through the AUTOSAR BSW/RTE shall be ensured. The signal CL15_01.CL15ON is to be extracted and provided to the application-SWCs properly.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	possible Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
x					Several COM stack & RTE modules interfere with each other and filter/translate/modify common data. This implies storing the data (registers, stack etc.) which makes them (illegally) accessible also from outside the COM stack.	1. ECC(random HW faults, no SW feature) 2. MPU, MMU (protect against unwanted writing) 3. E2E protection(detect modification of data) Note: routing here up to the SW-C would mean to apply E2E protection wrapper not COM callouts (if BSW has lower ASIL than required for the protected data)	E2E MPU NOT inside BSW(BSW partitioning not specified, especially OS and MCAL require full rights, partitioned BSW implementations are proprietary)
	x				Wrong/missing initialization or unwanted re-configuration of CanDrv, CanIf, PduR, COM	1. WdgM (maybe to check whether initialization took place only once) 2. CRC inside configuration source data to check their integrity (not enforced by AUTOSAR, feedback mechanism missing) 3. ECC(random HW faults, no protection against SW failures)	WdgM CRC lib
		x			Corruption of the message could take place.	1. Checksum for the Message/Signal (as part of the data but not the CAN CRC), 2. Message repetition, 3. Message counter, 4. E2E protection	CRC via COM stack (only if BSW is developed for same ASIL), E2E protection via E2E lib, Data repetition via COM stack (only if BSW is developed for same ASIL)
			x		Wrong/missing scheduling of CanDrv, PduR, Com, RTE	1. WdgM (WdgTrigger interfaces weak, every code part in BSW can re-trigger unintended 2. OS-Timing protection (ComStack integrated "data deadline monitoring" with limitation) 3. Proprietary OS "safety" extensions that grant freedom from interference also on BSW.	WdgM OS-Timing protection (Weakness1: a "trusted" BSW can overrun/disable such mechanisms by accident Weakness2: OS-timing protection potentially not fully sufficient from ISO 26262 [1] perspective)

Potential Cause(s) of Failure					Rational/ Failure Mode Example	possible Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
				x	Extraction of the signal from the bit stream in BSW can fail RTE data conversion/ filtering is not used here	Faults can be detected by E2E protection	

Table 7: Analysis of ECU03

5.3.3 Analysis of ECU27
The transmission of CL15_01.CL15ON between sender and receiver must be ensured (ASIL B).

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
o					n/a	n/a	n/a
	o				n/a	n/a	n/a
		x				1. Checksum for the Message/Signal (as part of the data but not the CAN CRC), 2. Message repetition, 3. Message counter, 4. E2E protection	CRC via COM stack (only if BSW is developed for same ASIL), E2E protection via E2E lib, Data repetition via COM stack (only if BSW is developed for same ASIL)
			o		n/a	n/a	n/a
				o	n/a	n/a	n/a

Table 8: Analysis of ECU27

5.3.4 Analysis of ECU04

The ECU shall detect any potential communication faults affecting the signal CL15ON that could lead to a violation of the safety goal.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
o					n/a	n/a	n/a
	o				n/a	n/a	n/a
		o			n/a	n/a	n/a
			x		The diagnostics can be “not executed” or executed with the wrong timing	WdgM (WdgTrigger interfaces weak, everybody in BSW can re-trigger unintended) OS-Timing protection Weakness: “trusted” SW can overrun/disable such mechanisms by accident. Proprietary OS “safety” extensions that grant freedom from interference also on BSW.	WdgM OS-Timing protection (Weakness1: a “trusted” BSW can overrun/disable such mechanisms by accident Weakness2: OS-timing protection potentially not fully sufficient from ISO 26262 [1] perspective)
				x	For example the diagnostic function implementation could fail(not catch the faults)	Review, tests	n/a

Table 9: Analysis of ECU04

5.3.5 Analysis of ECU06
The correct reading of the HW_LB_OFF input shall be ensured.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	safety measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
x						Multiple reading on SW-C level to discover “latent disturbance” (e.g. stack issues) Reading the signal through ADC and DIO and compare the results “Write protection” against unwanted “PinSet” or “pinReconfigre”(direction in<->out, Pull-up/down…) is not provided by AUTOSAR “detection” can be done proprietary e.g. by CRC of the config register.	No special AUTOSAR mechanism needed only “basics of MCAL”
	o				n/a	n/a	n/a
		o			n/a	n/a	n/a
			x			Pin can be configured in Interrupt Mode (if supported by the µC) so the user will be notified on each change. Background “poll state changes just to check whether interrupts are “lost” Pin reading can be used in polling mode=> Supervision of the timing/read frequency needs to be done by **WdgM or OS SC3** features(supervise the “reader SW-C” synch between update rate and reading frequency)	No special AUTOSAR mechanism needed only “basics of MCAL”
				x		Proprietary implementations in DIO, and the upper layers (correct hand shake) can ensure that the data processing of each module and the RAM content itself is consistent ReadChannel( ) + ReadGroup( ) and compare the appropriate Pin status to ensure correct reading in DIO	n/a

Table 10: Analysis of ECU06

5.3.6 Analysis of ECU07
The correct configuration of the HW_LB_OFF input port and pin shall be ensured.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
o					n/a	n/a	n/a
	x				Wrong/missing initialization or unwanted re-configuration of HW will lead to unpredictable HW PIN/PORT behavior	1. ECC(random HW faults, no protection against SW failures) 2. CRC (detect modification of the source data or even of the applied configuration in the peripheral) 3. Duplication of the source configuration data(compare it before it will be applied) 4. HW-supported protection mechanisms (MPU, MMU, register protection): protection against re-initialization 5. WdgM (maybe to check whether initialization took place only once)	MPU NOT inside BSW(BSW partitioning not specified, especially OS and MCAL require full rights, partitioned BSW implementations are proprietary) CRC-Lib
		o			n/a	n/a	n/a
			o		n/a	n/a	n/a
				o	n/a	n/a	n/a

Table 11: Analysis of ECU07

5.3.7 Analysis of ECU08
The correct transformation of the HW_LB_OFF input to the logical LB_OFF signal shall be ensured.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
o					n/a	n/a	n/a
	x				The pre-condition for a correct transformation (well working HW) is a correct initialization and configuration of the HW	1. ECC(random HW faults, no protection against SW failures) 2. CRC (detect modification of the source data or even of the applied configuration in the peripheral) 3. Duplication of the source configuration data(compare it before it will be applied) 4. HW-supported protection mechanisms (MPU, MMU, register protection): protection against re-initialization 5. WdgM (maybe to check whether initialization took place only once) 6.	E2E MPU NOT inside BSW(BSW partitioning not specified, especially OS and MCAL require full rights, partitioned BSW implementations are proprietary)
		o			n/a	n/a	n/a
			o		n/a	n/a	n/a
				o	n/a	n/a	n/a

Table 12: Analysis of ECU08

5.3.8 Analysis of ECU09
The correct routing of LB_OFF through the AUTOSAR BSW/RTE shall be ensured.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
x					Several I/O stack & RTE modules interfere with each other and filter/translate/modify common data. This implies storing the data (registers, stack etc.)which makes them (illegally)accessible also from outside the I/O stack.	ECC(random HW faults) MPU(protect against unwanted writing	MPU NOT inside BSW(BSW partitioning not specified, especially OS and MCAL require full rights, partitioned BSW implementations are proprietary)
	o				n/a	n/a	n/a
		o			n/a	n/a	n/a
			o		n/a	n/a	n/a
				o	n/a	n/a	n/a

Table 13: Analysis of ECU09

5.3.9 Analysis of ECU10

The ECU shall detect potential faults affecting LB_OFF that could lead to a violation of the safety goal.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
o					n/a	n/a	n/a
	o				n/a	n/a	n/a
		o			n/a	n/a	n/a
			x		The diagnostics can be "not executed" or executed with the wrong timing	1. WdgM (WdgTrigger interfaces weak, everybody in BSW can re-trigger unintended) 2. OS-Timing protection with limitation 3. Proprietary OS "safety" extensions that grant freedom from interference also on BSW.	WdgM OS-Timing protection (Weakness1: a "trusted" BSW can overrun/disable such mechanisms by accident Weakness2: OS-timing protection potentially not fully sufficient from ISO 26262 [1] perspective)
				x	For example the diagnostic function implementation could fail (not catch the faults)	Review, tests	n/a

Table 14: Analysis of ECU10

5.3.10 Analysis of ECU12
The Application-SWC shall determine the LB_OFF and CL15ON status as specified.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
x					Data Integrity in BSW is ensured by the previous requirements and mechanisms, Data Integrity of SW-C private data can still be affected	1. ECC(random HW faults), 2. CRC checks, data duplication, Plausibility checks, 3. MPU(protect SW-C against unwanted writing by other SW-C, protection against write access by "trusted" BSW NOT SPECIFIED by AUTOSAR), 4. divers SW-C designs (2xSW-C with own internal data but different implementation)	CRC-Lib (automatic generated and triggered checks are not supported by AUTOSAR), Memory protection,
	o				n/a	n/a	n/a
		x			Correct vertical data exchange is ensured by previous requirements and mechanisms (CL15ON comes in vertical from BSW), horizontal (RTE) can be affected details see ECU26	Details see ECU26	Details see ECU26
			x		Timing protection mechanisms that rely on proper interrupt handling are weak against "accidental" disabled interrupts (Trusted SW can do that)	Watchdog, HW-timer based mechanisms.	WdgM (WdgTrigger interfaces weak, everybody in BSW can re-trigger unintended), OS_Timing protection Weakness: a "trusted" BSW can overrun/disable such mechanisms by accident
				x		1. Review, tests, 2. divers SW-C designs (2xSW-C with own internal data but different implementation), 3. counter/key based control flow monitoring	WdgM (WdgTrigger interfaces weak, everybody in BSW can re-trigger unintended)

Table 15: Analysis of ECU12

5.3.11 Analysis of ECU13
The Application-SWC shall evaluate the light request conditions based on LB_OFF and CL15ON and their timing as specified.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity		Data Exchange	Timing & Control Flow	Data Processing			
Memory	Initialization & Configuration Data						
x					Data Integrity in BSW is ensured by the previous requirements and mechanisms, Data Integrity of SW-C private data can still be affected	1. ECC(random HW faults), 2. CRC checks, data duplication, plausibility checks, 3. MPU(protect SW-C against unwanted writing by other SW-C, protection against write access by "trusted" BSW NOT SPECIFIED by AUTOSAR), 4. divers SW-C designs (2xSW-C with own internal data but different implementation)	CRC-Lib (automatic generated and triggered checks are not supported by AUTOSAR), Memory protection,
	o				n/a	n/a	n/a
		x			Horizontal data exchange (RTE) can be affected details see ECU26	details see ECU26	details see ECU26
			x		Timing protection mechanisms that rely on proper Interrupt handling are weak against "accidental" disabled interrupts (Trusted SW can do that)	Watchdog, HW-timer based mechanisms.	WdgM (WdgTrigger interfaces weak, everybody in BSW can re-trigge unintended) OS_Timing protection-Weakness: a "trusted" BSW can overrun/disable such mechanisms by accident
				x		1. Review, tests, 2. divers SW-C designs (2xSW-C with own internal data but different implementation), 3. Counter/Key based controlFlow monitoring	WdgM (WdgTrigger interfaces weak, everybody in BSW can re-trigger unintended)

Table 16: Analysis of ECU13

5.3.12 Analysis of ECU14

The Application-SWC shall set or reset the light on command (Lights_ON) based on the LB_OFF and CL15ON evaluation results or if any fault is detected - set the light on command, if a communication fault of CL15_01 message is detected continuously for more than 200ms or set the light on command, if a fault on LB_OFF is detected continuously for more than 200ms.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
x					Data integrity in BSW is ensured by the previous requirements and mechanisms, data integrity of SW-C private data can still be affected	1. ECC(random HW faults), 2. CRC checks, data duplication, plausibility checks, 3. MPU(protect SW-C against unwanted writing by other SW-C, protection against write access by "trusted" BSW NOT SPECIFIED by AUTOSAR), 4. divers SW-C designs (2xSW-C with own internal data but different implementation)	CRC-Lib (automatic generated and triggered checks are not supported by AUTOSAR), Memory protection,
	o				n/a	n/a	n/a
		x			Horizontal data exchange (RTE) can be affected details see ECU26	Details see ECU26	Details see ECU26
			x		Timing protection mechanisms that rely on proper Interrupt handling are weak against "accidental" disabled interrupts (Trusted SW can do that)	Watchdog, HW-timer based mechanisms.	WdgM (WdgTrigger interfaces weak, everybody in BSW can re-trigger unintended) OS_Timing protection-Weakness: a "trusted" BSW can overrun/disable such mechanisms by accident
				x		1. Review, tests, 2. divers SW-C designs (2xSW-C with own internal data but different implementation), 3. counter/key based control flow monitoring	WdgM (WdgTrigger interfaces weak, everybody in BSW can re-trigger unintended)

Table 17: Analysis of ECU14

5.3.13 Analysis of ECU15

The Application-SWC shall activate both daytime running lights (DRL_ON) if a failure of both LB bulbs is detected continuously for 200ms (read_current_L, read_current_R).

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity		Data Exchange	Timing & Control Flow	Data Processing			
Memory	Initialization & Configuration Data						
x					Data integrity in BSW is ensured by the previous requirements and mechanisms, data integrity of SW-C private data can still be affected	1. ECC(random HW faults), 2. CRC checks, data duplication, plausibility checks, 3. MPU(protect SW-C against unwanted writing by other SW-C, protection against write access by “trusted” BSW NOT SPECIFIED by AUTOSAR), 4. divers SW-C designs (2xSW-C with own internal data but different implementation)	CRC-Lib (automatic generated and triggered checks are not supported by AUTOSAR), Memory protection
	o				n/a	n/a	n/a
		x			Horizontal data exchange (RTE) can be affected details see ECU26	Details see ECU26	Details see ECU26
			x		Timing protection mechanisms that rely on proper Interrupt handling are weak against “accidental” disabled interrupts (Trusted SW can do that), The SW-C can rely on a frequent and guarantied “Runnable call/execution” => counts the timing then derived from it	Watchdog, HW-timer based mechanisms.	WdgM (WdgTrigger interfaces weak, everybody in BSW can re-trigger unintended) OS_Timing protection - Weakness: a “trusted” BSW can overrun/disable such mechanisms by accident
				x	The SW-C internal realization can contain systematic faults	1. Review, tests, 2. divers SW-C designs (2xSW-C with own internal data but different implementation), 3. counter/key based control flow monitoring	WdgM (WdgTrigger interfaces weak, everybody in BSW can re-trigger unintended)

Table 18: Analysis of ECU15

5.3.14 Analysis of ECU16

The correct powering of the bulbs according to the Lightsreqest and the specification are to be signaled via set_pwm command.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
o					n/a	n/a	n/a
	x					CRC check of the Configuration parameter	CRC-Lib
		o			n/a	n/a	n/a
			x			Watchdog to ensure at least a “ netime” execution of the cmd API	WdgM
				o	n/a	n/a	n/a

Table 19: Analysis of ECU16

5.3.15 Analysis of ECU 29
The correct transformation of the logical PWM-I-Signal to the SPI BUS message shall be ensured.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
o					n/a	n/a	n/a
	x				Wrong/missing initialization or unwanted re-configuration of HW will lead to unpredictable SPI HW behavior	1. ECC(random HW faults, no protection against SW failures) 2. CRC (detect modification of the source data or even of the applied configuration in the Peripheral) 3. Duplication of the source configuration data(compare it before it will be applied) 4. HW-supported protection mechanisms (MPU, MMU, register protection): protection against re-initialization 5. WdgM (maybe to check whether initialization took place only once)	E2E MPU NOT inside BSW(BSW partitioning not specified, especially OS and MCAL require full rights, partitioned BSW implementations are proprietary) CRC-Lib
		o			n/a	n/a	n/a
			x		Interrupt mode: unexpected long/often locked interrupts or wrong interrupt prioritization can lead to “ignored” SPI “data available” triggers Polling mode: The poll frequency/timing can be affected by wrong OS scheduling/task prioritization configuration, unexpected preemption etc.	Internal watchdog a) direct reset b) “top level interrupt” which cannot be blocked by “locked interrupts” (machine dependent) External watchdog direct reset	WdgM
				o	n/a	n/a	n/a

Table 20: Analysis of ECU29

5.3.16 Analysis of ECU17

The correct routing of the set_pwm request to the µC SPI output shall be ensured.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
x					The CDD and SpiDrv modules interfere with each other and filter/translate/modify common data. This implies storing the data (registers, stack etc.)which makes them (illegally)accessible also from outside the direct data flow	Read the SPI output back in “LoopBack mode” Mirror The SPI Output Define a protocol that requires an answer from the receiver (received “Bit” and “answer the cmd”)	n/a
	x				Wrong/missing initialization or unwanted re-configuration of SpiDrv	1. WdgM (maybe to check whether initialization took place only once) 2. CRC inside configuration source data to check their integrity(not enforced by AUTOSAR, feedback mechanism missing) 3. ECC(random HW faults, no protection against SW failures)	WdgM CRC lib
		o			n/a	n/a	n/a
			x		Wrong/missing scheduling of SpiDrv	Watchdog	AUTOSAR doesn’t specify WdgM usage inside BSW, each watch point would have to be added project specifically; this is the opposite of a re-usable generic BSW. Therefore an “activity/update” counter mechanism potentially has to be part of the transmitted content
				x	A Pin usually used as CS Pin can by modified by accident as part of an illegal DIO “request” or even more worse by direct access to the Port/Pin	HW protection of the µC no specific AUTOSAR feature available because MPU usage in BSW not specified	n/a

Table 21: Analysis of ECU17

5.3.17 Analysis of ECU20
When the bulbs are powered, the Application-SWC shall evaluate the status of the bulbs.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
x					Data Integrity in BSW is ensured by the previous requirements and mechanisms, data integrity of SW-C private data can still be affected	1. ECC(random HW faults), 2. CRC checks, data duplication, plausibility checks, 3. MPU(protect SW-C against unwanted writing by other SW-C, protection against write access by "trusted" BSW NOT SPECIFIED by AUTOSAR), 4. divers SW-C designs (2xSW-C with own internal data but different	CRC-Lib (automatic generated and triggered checks are not supported by AUTOSAR), Memory protection,
	o				n/a	n/a	n/a
		x			Horizontal data exchange (RTE) can be affected details see ECU26	Details see 5.2.26 Analysis of ECU26	Details see 5.2.26 Analysis of ECU26
			x		Timing protection mechanisms that rely on proper Interrupt handling are weak against "accidental" disabled interrupts (Trusted SW can do that), The SW-C can rely on a frequent and guarantied "Runnable call/execution" => counts the timing then derived from it	Watchdog, HW-timer based mechanisms.	WdgM (WdgTrigger interfaces weak, everybody in BSW can re-trigger unintended) OS_Timing protection-Weakness: a "trusted" BSW can overrun/disable such mechanisms by accident
				x	The SW-C internal realization can contain systematic faults	1. Review, tests, 2. divers SW-C designs (2xSW-C with own internal data but different implementation), 3. counter/key based control flow monitoring	WdgM (WdgTrigger interfaces weak, everybody in BSW can re-trigger unintended)

Table 22: Analysis of ECU20

5.3.18 Analysis of ECU30

When the bulbs are powered, the Actuator-SWC shall read and provide the status of the bulbs (read_current_L, read_current_R).

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity		Data Exchange	Timing & Control Flow	Data Processing			
Memory	Initialization & Configuration Data						
x					Data integrity in BSW is ensured by the previous requirements and mechanisms, data integrity of SW-C private data can still be affected	1. ECC(random HW faults), 2. CRC checks, data duplication, plausibility checks, 3. MPU(protect SW-C against unwanted writing by other SW-C, protection against write access by “trusted” BSW NOT SPECIFIED by AUTOSAR), 4. divers SW-C designs (2xSW-C with own internal data but different implementation)	CRC-Lib (automatic generated and triggered checks are not supported by AUTOSAR), Memory protection,
	o				n/a	n/a	n/a
		x			Horizontal data exchange (RTE) can be affected. Details see ECU26	Details ECU26	Details see ECU26
			x		Timing protection mechanisms that rely on proper Interrupt handling are weak against “accidental” disabled interrupts (Trusted SW can do that), The SW-C can rely on a frequent and guarantied “Runnable call/execution” => counts the timing then derived from it	Watchdog, HW-timer based mechanisms.	WdgM (WdgTrigger interfaces weak, everybody in BSW can re-trigger unintended) OS_Timing protection-Weakness: a “trusted” BSW can overrun/disable such mechanisms by accident
				x	The SW-C internal realization can contain systematic faults	1. Review, tests, 2. divers SW-C designs (2xSW-C with own internal data but different implementation), 3. counter/key based control flow monitoring	WdgM (WdgTrigger interfaces weak, everybody in BSW can re-trigger unintended)

Table 23: Analysis of ECU20a

5.3.19 Analysis of ECU21
Detected faults shall be signaled via CAN BUS LBFailure.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
o					n/a	n/a	n/a
	x				Wrong/missing initialization or unwanted re-configuration of HW will lead to unpredictable CAN HW behavior	1. ECC(random HW faults, no protection against SW failures) 2. CRC (detect modification of the source data or even of the applied configuration in the peripheral) 3. Duplication of the source configuration data(compare it before it will be applied) 4. HW-supported protection mechanisms (MPU, MMU, register protection): protection against re-initialization 5. WdgM (maybe to check whether initialization took place only once)	MPU NOT inside BSW(BSW partitioning not specified, especially OS and MCAL require full rights, partitioned BSW implementations are proprietary) CRC-Lib
		x			Corruption of the message could take place.	1. Checksum for the message/signal (as part of the data but not the CAN CRC), 2. Message repetition, 3. Message counter, 4. E2E protection	CRC via COM stack (only if BSW is developed for same ASIL), E2E protection via E2E lib, Data repetition via COM stack (only if BSW is developed for same ASIL)
			o		n/a	n/a	n/a
				o	n/a	n/a	n/a

Table 24: Analysis of ECU21

5.3.20 Analysis of ECU23

The Actuator-SWC shall initiate a diagnosis of each element of the bulb health measurement path and evaluate the results.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
x					Data integrity in BSW is ensured by the previous requirements and mechanisms, data integrity of SW-C private data can still be affected	1. ECC(random HW faults), 2. CRC checks, data duplication, plausibility checks, 3. MPU(protect SW-C against unwanted writing by other SW-C, protection against write access by "trusted" BSW NOT SPECIFIED by AUTOSAR), 4. divers SW-C designs (2xSW-C with own internal data but different implementation)	CRC-Lib (automatic generated and triggered checks are not supported by AUTOSAR), Memory protection,
	o				n/a	n/a	n/a
		x			Horizontal data exchange (RTE) can be affected details see ECU26	details see ECU26	details see ECU26
			x		Timing protection mechanisms that rely on proper Interrupt handling are weak against "accidental" disabled interrupts (Trusted SW can do that), The SW-C can rely on a frequent and guarantied "Runnable call/execution" => counts the timing then derived from it	Watchdog, HW-timer based mechanisms.	WdgM (WdgTrigger interfaces weak, everybody in BSW can re-trigger unintended) OS_Timing protection-Weakness: a "trusted" BSW can overrun/disable such mechanisms by accident
				x	The SW-C internal realization can contain systematic faults	1. Review, tests, 2. divers SW-C designs (2xSW-C with own internal data but different implementation), 3. counter/key based control flow monitoring	WdgM (WdgTrigger interfaces weak, everybody in BSW can re-trigger unintended)

Table 25: Analysis of ECU23

5.3.21 Analysis of ECU24

The correct routing of bulb health measurement values read_current_L, read_current_R through the AUTOSAR BSW/RTE shall be ensured.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
x					Several I/O stack modules & RTE interfere with each other and filter/translate/modify common data. This implies storing the data (registers, stack etc.) which makes them (illegally) accessible also from outside the I/O stack.	ECC(random HW faults) MPU(protect against unwanted writing), Data duplication in/on MCAL routing through two different channels (BSW and CDD) and result comparison on receiver side, data injection into ADC => eadout=>routing=> comparison against expected value	MPU(protect against unwanted writing, NOT SPECIFIED by AUTOSAR on BSW)
	o				n/a	n/a	n/a
		o			n/a	n/a	n/a
			o		n/a	n/a	n/a
				o	n/a	n/a	n/a

Table 26: Analysis of ECU24

5.3.22 Analysis of ECU25
The ADC-HW shall convert the measured current to read_current_L, read_current_R

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
o					n/a	n/a	n/a
	x				Wrong/missing initialization or unwanted re-configuration of HW will lead to unpredictable ADC HW behavior	1. ECC(random HW faults, no protection against SW failures) 2. CRC (detect modification of the source data or even of the applied configuration in the peripheral) 3. Duplication of the source configuration data(compare it before it will be applied) 4. HW-supported protection mechanisms (MPU, MMU, register protection…): protection against re-initialization 5. WdgM (maybe to check whether initialization took place only once)	MPU NOT inside BSW(BSW partitioning not specified, especially OS and MCAL require full rights, partitioned BSW implementations are proprietary) CRC-Lib
		o			n/a	n/a	n/a
			x		Only relevant if the peripheral is actively triggered by SW	Timing supervision	WdgM (currently not specified for BSW modules)
				x	If e.g. an ADC is not working well it will not report a valid bulb health state.	Use two independent ADCs and compare the results, do ADC self-diagnosis by sampling known data and compare against expected results trigger built in ADC HW self-diagnosis	HWTestManager (currently not released yet) could control the activity, MCAL “HWselfTest” Interfaces (currently not specified) could support the activity Implementation currently only via ComplexDeviceDriver and ECU HW_abstraction SW-C, controlled through Actuator-SW-C via AUTOSAR interfaces

Table 27: Analysis of ECU25

5.3.23 Analysis of ECU26

The correct data exchange (timing and content) between the SW-components shall be ensured.

Potential Cause(s) of Failure					Rational/ Failure Mode Example	Safety Measures	AUTOSAR Mechanisms
Data Integrity							
Memory	Initialization & Configuration Data	Data Exchange	Timing & Control Flow	Data Processing			
o					n/a	n/a	n/a
	o				n/a	n/a	n/a
		x				E2E protection, MessageCounter, MPU, safe path through RTE	E2E protection via E2E lib
			o		n/a	n/a	n/a
				o	n/a	n/a	n/a

Table 28: Analysis of ECU26

6 Conclusion

The compliance with ISO 26262 [1] requirements within an AUTOSAR context requires the fulfillment of freedom from interference (FFI) in case of mixed ASILs for application SWCs as well as AUTOSAR RTE and BSWs.

Therefore, safety measures have to be identified to provide appropriate countermeasures for all FFI objects (timing and execution; memory; exchange of information) according to ISO 26262-6 [1].

Within the example one of the aspects was to show by what means the existing AUTOSAR releases can support the requirements for FFI.

In chapter "*SW aspects and potential failure modes*" of this document, an exemplary analysis of required safety measures and already existing safety measures by AUTOSAR was shown. However, based on this example, several points for potential improvement within the current AUTOSAR release were identified.
They are highlighted within the next subsection.

6.1 Potential Safety Improvement for Future AUTOSAR Releases

The following aspects are worth to be considered in following AUTOSAR Releases.

MCAL
The API should be extended in a generic way to support µC/peripheral specific HW self-diagnosis (SW or HW driven)
The specification shall be extended by a check of the applied configuration (already assigned to the HW peripheral) against reference configuration data (e.g. from NVM).
Diagnostics for HW configuration, e.g. read back of configuration registers, are currently not supported by the MCAL, but are needed for technical safety measures.

MPU usage in BSW
The current BSW specification does not define detailed measures how to utilize an MPU already on BSW level.
Potential risk: A fault in BSW can trigger unwanted Wdg activities and therefore the WdgM would be no longer independent. It is then not usable anymore as measure to guarantee FFI.
An unrestricted BSW must either

- Be developed accordingly to the highest required ASIL of the Appl. SW or
- Be implemented in a way that guarantees FFI. (e. g. allows usage of an MPU in BSW).

The BSW Distribution in Safety Systems (see [4] chapter 3 "BSW Distribution in Safety Systems) provides a kind of work-around but no generic "built-in" solution.
A consequently applicable concept for partitioning in all levels of granularity is required, (horizontal, vertical or both).

Timing supervision of BSW
Currently, no AUTOSAR requirements exist which demand to use timing supervision or control flow monitoring points in BSW components. The usage of WdgM in BSW

modules is not specified by the AUTOSAR specification. It is not forbidden but also not mandatory, so “out of the box” generic BSW stacks cannot be configured to use such a feature. Applying the WdgM for that purpose means a modification of BSW in each single application. Re-usable SW should provide this out of the box and configurable to specific conditions

API definition rules
Call by reference APIs (COMStack) require access to the source/target data location even though e.g. sender/receiver are ASIL-X and COM is QM.
Principle API design weakness requires project specific consideration, and therefore potentially proprietary extensions.

In principle it is required to support the proposed mechanisms mentioned in ISO 26262-6:2011(E) Annex D “Freedom from interference between software elements” [1] considering mixed criticality parts in AUTOSAR SW-Cs, RTE and BSW.

7 Abbreviation/Glossary

Abbreviation	Translation (German)	Explanation/Comment
ADC		Analog-Digital Converter
ASIL		Automotive Safety Integrity Level
BCU		Body Control Unit
BSW		Basic Software Components
CAN		Controller Area Network
DRL	German “Tagfahrlicht”	Daytime Running Light
DIO		Digital Input/Output
ECU		Electronic Control Unit
FLM		Front Light Management
FS		Functional Safety
FSC		Functional Safety Concept
FTT	German “Fehlertoleranzzeit"	Fault Tolerance Time
Head Light	German “Scheinwerfer”	The physical part of head lights
High Beam	German “Aufblendlicht/Fernlicht”	The Function High Beam
HWA		Hardware Abstraction
HW		Hardware
LB	German “Abblendlicht”	The Function Low Beam
LS		Light Switch
oc/sc		Open circuit/ Short circuit
OEM		Original Equipment Manufacturer
OS		Operating System
QM		Quality Management
RTE		Runtime Environment
SPI		Serial Peripheral Interface
SWC		Software Component
VFB		Virtual Function Bus

8 References

[1] ISO26262 International Standard, First edition 2011-11-15

[2] Layered Software Architecture

[3] Overview of Functional Safety Measures in AUTOSAR

[4] Guide to BSW Distribution

9 Figures and Tables

Figures

Tables