

Software Safety Requirements and Architecture

Lane Assistance

**Document Version: [1]**

**Template Version 1.0, Released on 2017-06-21**



# Document history

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Editor | Description |
| 10/10/2017 | 0.1 | Gustavo Espindola | First draft |
| 12/10/2017 | 1 | Gustavo Espindola | Release candidate 1 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 

# Table of Contents

Inhalt

[Document history 2](#_Toc495585982)

[Table of Contents 2](#_Toc495585983)

[Purpose 3](#_Toc495585984)

[Inputs to the Software Requirements and Architecture Document 3](#_Toc495585985)

[Technical safety requirements 3](#_Toc495585986)

[Refined Architecture Diagram from the Technical Safety Concept 4](#_Toc495585987)

[Software Requirements 4](#_Toc495585988)

[Refined Architecture Diagram 10](#_Toc495585989)

# Purpose

This documents purpose is to create easily implementable requirements, which must map to the technical safety requirements in a clear manner. This can include specific variable names, communication protocols and communication mechanisms.

# Inputs to the Software Requirements and Architecture Document

## Technical safety requirements

Technical Safety Requirements related to Functional Safety Requirement 01-01 are:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Technical Safety Requirement** | **ASIL** | **Fault Tolerant Time Interval** | **Architecture Allocation** | **Safe State** |
| Technical  Safety  Requirement  01 | The LDW safety component shall ensure that the amplitude of the  'LDW\_Torque\_Request' sent  to the 'Final electronic power  steering Torque' component  is below 'Max\_Torque\_Amplitude. | C | 50ms | Power Steering ECU – Specifically the LDW safety block. | LDW functionality set off and requested torque set to 0. |
| Technical  Safety  Requirement  02 | As soon as the LDW function  deactivates the LDW feature,  the 'LDW Safety' software  block shall send a signal to  the car display ECU to turn  on a warning light. | C | 50ms | Power Steering ECU – Specifically the LDW safety block | LDW functionality set off and requested torque set to 0. |
| Technical  Safety  Requirement  03 | As soon as a failure is  detected by the LDW  function, it shall deactivate  the LDW feature and the  'LDW\_Torque\_Request' shall  be set to zero. | C | 50ms | Power Steering ECU – Specifically the LDW safety block. | LDW functionality set off and requested torque set to 0. |
| Technical  Safety  Requirement  04 | The validity and integrity of  the data transmission for  'LDW\_Torque\_Request'  signal shall be ensured. | C | 50ms | Power Steering ECU – Specifically the LDW safety block. | LDW functionality set off and requested torque set to 0. |
| Technical  Safety  Requirement  05 | Memory test shall be  conducted at start up of the  EPS ECU to check for any  faults in memory. | A | Duration of ignition cycle | Power steering ECU bootloader | LDW functionality set off and requested torque set to 0. |

## Refined Architecture Diagram from the Technical Safety Concept



# Software Requirements

**Lane Departure Warning (LDW) Amplitude Malfunction Software Requirements:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Technical Safety Requirement** | **ASIL** | **Fault Tolerant Time Interval** | **Allocation to Architecture** | **Safe State** |
| Technical  Safety  Requirement  01 | The LDW safety component shall ensure that the amplitude of the LDW\_Torque\_Request sent to the Final Electronic Power Steering Torque component is below Max\_Torque\_Amplitude | C | 50ms | LWD Safety block | LDW functionality set off and requested torque set to 0. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Software Safety Requirement | ASIL | Allocation Software Elements | Safe State |
| Software  Safety  Requirement  01-01 | The input signal "Primary\_LDW\_Torq\_Req" shall be read and pre-processed to determine the torque request coming from the "Basic/Main LAFunctionality" SW Component. Signal "processed\_LDW\_Torq\_Req" shall be generated at the end of the processing. | C | LDW\_SAFETY\_INPUT\_PROCESSING | N/A |
| Software Safety Requirement 01-02 | In case the "processed\_LDW\_Torq\_Req" signal has a value greater than "Max\_Torque\_Ampltide\_LDW"(maximum allowed safe torque), the torque signal "limited\_LDW\_Torq\_Req" shall be set to 0, else "limited\_LDW\_Torq\_Req"shall take the value of "processed\_LDW\_Torq\_Req". | C | TORQUE\_LIMITER | limited\_LDW\_Torq\_Req = 0 Nm |
| Software Safety Requirement 01-03 | "limited\_LDW\_Torq\_Req" shall be transformed into a signal "LDW\_Torq\_Req" which is suitable to be transmitted outside of the LDW Safetycomponent ("LDW Safety") to the "Final EPS Torque" component. Also see SofSafReq02-01 andSofSafReq02-02 | C | LDW\_SAFETY\_OUTPUT\_GENERATOR | LDW\_Torque\_Req = 0 Nm |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Technical Safety Requirement** | **ASIL** | **Fault Tolerant Time Interval** | **Allocation to Architecture** | **Safe State** |
| Technical  Safety  Requirement  02 | The validity and integrity of the data transmission for LDW\_Torque\_Request signal shall be ensured | C | 50ms | Data Transmission integrity check | N/A |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Software Safety Requirement** | **ASIL** | **Allocation Software Elements** | **Safe State** |
| Software Safety Requirement 02-01 | Any data to be transmittedoutside of the LDW Safetycomponent ("LDW Safety")including "LDW\_Torque\_Req" and "activation\_status" (see SofSafReq03-02) shall beprotected by an End2End(E2E)protection mechanism | C | E2Ecalc | LDW\_Torq\_Req = 0 Nm |
| Software Safety Requirement 02-02 | The E2E protection protocol shall contain and attach the control data: alive counter (SQC) and CRC to the data to be transmitted. | C | E2ECalc | LDW\_Torq\_Req = 0 Nm |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Technical Safety Requirement** | **ASIL** | **Fault Tolerant Time Interval** | **Allocation to Architecture** | **Safe State** |
| Technical  Safety  Requirement  03 | As soon as a failure is detected by the LDW function, it shall deactivate the LDW feature and the LDW\_Torque\_Request shall be set to zero | C | 50 ms | LDW Safety block | Requested LDW torque is set to zero |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Software Safety Requirement** | **ASIL** | **Allocation Software Elements** | **Safe State** |
| Software Safety Requirement03-01 | Each of the SW elements shall output a signal to indicate any error which is detected by the element. Error signal = error\_status\_input(LDW\_SAFET Y\_INPUT\_PROCESSING), error\_status\_torque\_limiter(TOR QUE\_LIMITER), error\_status\_output\_gen(LDW\_ SAFETY\_OUTPUT\_GENERAT OR) | C | ALL | N/A |
| Software Safety Requirement03-02 | A software element shall evaluate the error status of all the other software elements and in case any 1 of them indicates an error, it shall deactivate the LDW feature(“activation\_status”=0) | C | LDW\_SAFETY\_ACTIVATOIN | Activation\_status = 0  (LDW function deactivated) |
| Software Safety Requirement03-03 | In case of no errors from the software elements, the status of the LDW feature shall be set to activated (“activation\_status”=1) | C | LDW\_SAFETY\_ACTIVATOIN | N/A |
| Software Safety Requirement03-04 | In case an error is detected by any of the software elements, it shall set the value of its corresponding torque to 0 so that “LDW\_Torq\_Req” is set to 0 | C | ALL | LDW\_Torq\_Req = 0 |
| Software Safety Requirement03-05 | Once the LDW functionality has been deactivated, it shall stay deactivated till the time the ignition is switched from off to on again. | C | LDW\_SAFETY\_ACTIVATOIN | Activation\_Status = 0  (LDW function deactivated) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Technical Safety Requirement** | **ASIL** | **Fault Tolerant Time Interval** | **Allocation to Architecture** | **Safe State** |
| Technical  Safety  Requirement  04 | As soon as the LDW function deactivates the LDW feature, the LDW Safety software block shall send a signal to the car display ECU to turn on a warning light | C | 50 ms | LDW Safety | LDW torque output torque is set to zero |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Software Safety Requirement** | **ASIL** | **Allocation Software Elements** | **Safe State** |
| Software Safety Requirement 04-01 | When the LDW function is deactivated (activation\_status set to 0), the activation\_status shall be sent to the car display ECU. | C | LDW\_SAFETY\_ACTIVATIO  N, CarDisplay  ECU | N/A |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Technical Safety Requirement** | **ASIL** | **Fault Tolerant Time Interval** | **Allocation to Architecture** | **Safe State** |
| Technical  Safety  Requirement  05 | Memory test shall be conducted at start-up of the EPS ECU to check for any faults in memory | C | 50ms | Ignition Cycle (ECU bootloader) | LDW torque output is set to zero. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Software Safety Requirement** | **ASIL** | **Allocation Software Elements** | **Safe State** |
| Software Safety Requirement 05-01 | A CRC verification check over the software code in the Flash memory shall be done every time the ignition is switched from off to on to check for any corruption of content. | A | MEMORYTEST | Activation\_Status = 0 |
| Software Safety Requirement 05-02 | Standard RAM tests to check the data bus, address bus and device integrity shall be done every time the ignition is switched from off to on (E.g.walking 1s test, RAM pattern test. Refer RAM and processor vendor recommendations ) | A | MEMORYTEST | Activation\_Status = 0 |
| Software Safety Requirement 05-03 | The test result of the RAM or Flash memory shall be indicated to the LDW\_Safety component via the “test\_status” signal | A | MEMORYTEST | Activation\_Status = 0 |
| Software Safety Requirement 05-04 | In case any fault is indicated via the “test\_status” signal the INPUT\_LDW\_PROCESSING shall set an error on error\_status\_input (=1) so that the LDW functionality is deactivated and the LDWTorque is set to 0. | A | MEMORYTEST | Activation\_Status = 0 |

# 

# Refined Architecture Diagram



[End of Document]