

Software Safety Requirements and Architecture

Lane Assistance

**Document Version: 3.0**

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



# Document history

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Editor | Description |
| 2018-03-13 | 1.0 | Abhishek Mantha | Document initialization |
| 2018-03-13 | 2.0 | Abhishek Mantha | Revising for Final Submission |
| 2018-03-14 | 3.0 | Abhishek Mantha | Final Submission |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

[Document history](#_1t3h5sf)

[Table of Contents](#_pyp8fd3vpmy3)

[Purpose](#_c1lz2bx22jid)

[Inputs to the Software Requirements and Architecture Document](#_hjpgfzcjxim1)

[Technical safety requirements](#_lc6owg9q3amb)

[Refined Architecture Diagram from the Technical Safety Concept](#_db3hhee81tpq)

[Software Requirements](#_pul4igmpfvr0)

[Refined Architecture Diagram](#_b7fyegncumoz)

# Purpose

The purpose of the Software Requirements and Architecture Document is to develop requirements and metrics which the item can be verified that will ensure its functional safety.

# 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 | LDW safety block shall ensure that amplitude of ‘LDW\_Torque\_Request’ sent to EPS ‘Final Torque’ component is below ‘Max\_Torque\_Amplitude’. | C | 50 ms | LDW Safety Functionality Block | ‘LDW\_Torque\_Request’ shall be set to 0. |
| Technical  Safety  Requirement  02 | As soon as LDW function deactivates LDW feature, the 'LDW Safety' software block shall send ‘LDW\_Error\_Status’ to car display ECU to turn on warning light. | C | 50 ms | LDW Safety Functionality Block | ‘LDW\_Torque\_Request’ shall be set to 0. |
| Technical  Safety  Requirement  03 | As soon as failure is detected by LDW function, it shall deactivate LDW feature and 'LDW\_Torque\_Request' shall be set to 0. | C | 50 ms | LDW Safety Functionality Block | ‘LDW\_Torque\_Request’ shall be set to 0. |
| Technical  Safety  Requirement  04 | Validity and integrity of data transmission for 'LDW\_Torque\_Request' signal shall be ensured. | C | 50 ms | Data Transmission Integrity Check Component | ‘LDW\_Torque\_Request’ shall be set to 0. |
| Technical  Safety  Requirement  05 | Memory test shall be conducted at start up of EPS ECU to check for any faults in memory. | A | Ignition cycle | Safety Startup Component | ‘LDW\_Torque\_Request’ shall be 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** | **Architecture Allocation** | **Safe State** |
| Technical  Safety  Requirement  01 | LDW safety block shall ensure that amplitude of ‘LDW\_Torque\_Request’ sent to EPS ‘Final Torque’ component is below ‘Max\_Torque\_Amplitude’. | C | 50 ms | LDW Safety Functionality Block | ‘LDW\_Torque\_Request’ shall be set to 0. |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Technical Safety Requirement** | **ASIL** | **Fault Tolerant Time Interval** | **Architecture Allocation** | **Safe State** |
| Technical  Safety  Requirement  02 | As soon as LDW function deactivates LDW feature, the 'LDW Safety' software block shall send ‘LDW\_Error\_Status’ to car display ECU to turn on warning light. | C | 50 ms | LDW Safety Functionality Block | ‘LDW\_Torque\_Request’ shall be set to 0. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Software Safety Requirement** | **ASIL** | **Allocation Software Elements** | **Safe State** |
| Software Safety Requirement 02-01 | When the LDW function is deactivated (activation\_status set to 0), the activation\_status shall be sent to the car displayECU. | C | LDW\_SAFETY\_ACTIVATION, Car Display ECU | N/A |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Technical Safety Requirement** | **ASIL** | **Fault Tolerant Time Interval** | **Architecture Allocation** | **Safe State** |
| Technical  Safety  Requirement  03 | As soon as failure is detected by LDW function, it shall deactivate LDW feature and 'LDW\_Torque\_Request' shall be set to 0. | C | 50 ms | LDW Safety Functionality Block | ‘LDW\_Torque\_Request’ shall be set to 0. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Software Safety Requirement** | **ASIL** | **Allocation Software Elements** | **Safe State** |
| Software Safety Requirement03-01 | Each of SW elements shall output a signal to indicate any error which is detected by element. Error signal = error\_status\_input(LDW\_SAFETY\_INPUT\_PROCESSING), error\_status\_torque\_limiter(TORQUE\_LIMITER), error\_status\_output\_gen(LDW\_SAFETY\_OUTPUT\_GENERATOR) | C | All | N/A |
| Software Safety Requirement03-02 | Software element shall evaluate error status of all 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\_ACTIVATION | Activation\_status = 0 (LDW function deactivated) |
| Software Safety Requirement03-03 | In case of no errors from software elements, status of LDW feature shall be set to activated (“activation\_status”=1) | C | LDW\_SAFETY\_ACTIVATION | N/A |
| Software Safety Requirement03-04 | In case an error is detected by any of software elements, it shall set 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 LDW functionality has been deactivated, it shall stay deactivated till ignition is switched from off to on again. | C | LDW\_SAFETY\_ACTIVATION | Activation\_status = 0 (LDW function deactivated) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Technical Safety Requirement** | **ASIL** | **Fault Tolerant Time Interval** | **Architecture Allocation** | **Safe State** |
| Technical  Safety  Requirement  04 | Validity and integrity of data transmission for 'LDW\_Torque\_Request' signal shall be ensured. | C | 50 ms | Data Transmission Integrity Check Component | ‘LDW\_Torque\_Request’ shall be set to 0. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Software Safety Requirement** | **ASIL** | **Allocation Software Elements** | **Safe State** |
| Software Safety Requirement 04-01 | Any data to be transmitted outside of LDW Safety component (“LDW Safety”) including "LDW\_Torque\_Req" and “activation\_status” (see SofSafReq03-02) shall be protected by an End2End(E2E) protection mechanism | C | E2ECalc | LDW\_Torq\_Req= 0 (Nm) |
| Software Safety Requirement  04-02 | E2E protection protocol shall contain and attach control data: alive counter (SQC) and CRC to data to be transmitted. | C | E2ECalc | LDW\_Torq\_Req= 0 (Nm) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Technical Safety Requirement** | **ASIL** | **Fault Tolerant Time Interval** | **Architecture Allocation** | **Safe State** |
| Technical  Safety  Requirement  05 | Memory test shall be conducted at start up of EPS ECU to check for any faults in memory. | A | Ignition cycle | Safety Startup Component | ‘LDW\_Torque\_Request’ shall be set to 0. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Software Safety Requirement** | **ASIL** | **Allocation Software Elements** | **Safe State** |
| Software Safety Requirement 05-01 | CRC verification check over software code in Flash memory shall be done every time 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 data bus, address bus and device integrity shall be done every time 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 | Test result of RAM or Flash memory shall be indicated to LDW\_Safety component via “test\_status” signal | A | MEMORYTEST | Activation\_status = 0 |
| Software Safety Requirement 05-04 | In case any fault is indicated via “test\_status” signal INPUT\_LDW\_PROCESSING shall set an error on error\_status\_input (=1) so that LDW functionality is deactivated and LDWTorque is set to 0 | A | LDW\_SAFETY\_INPUT\_PROCESSING | Activation\_status = 0 |

# Refined Architecture Diagram

****