

Software Safety Requirements and Architecture

Lane Assistance

**Document Version: [Version]**

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



# Document history

**[Instructions: Fill in the date, version and description fields. You can fill out the Editor field with your name if you want to do so. Keep track of your editing as if this were a real world project.**

**For example, if this were your first draft or first submission, you might say version 1.0. If this is a second submission attempt, then you'd add a second line with a new date and version 2.0]**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Editor | Description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 

# Table of Contents

**[Instructions: We have provided a table of contents. If the table of contents is not showing up correctly in your word processor of choice, please update it. The table of contents should show each section of the document and page numbers or links. Most word processors can do this for you. In** [**Google Docs**](https://support.google.com/docs/answer/116338?co=GENIE.Platform%3DDesktop&hl=en)**, you can use headings for each section and then go to Insert > Table of Contents.** [**Microsoft Word**](https://support.microsoft.com/en-us/help/285059/how-to-create-a-table-of-contents-by-marking-text-in-word) **has similar capabilities]**

[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 software requirements and architecture is to develop requirements and metrics against which the items can be verified that will ensure its functional safety. Software safety requirements are more specific compared to the technical safety requirements as they provide provide precise instructions to the software engineers developing the program.

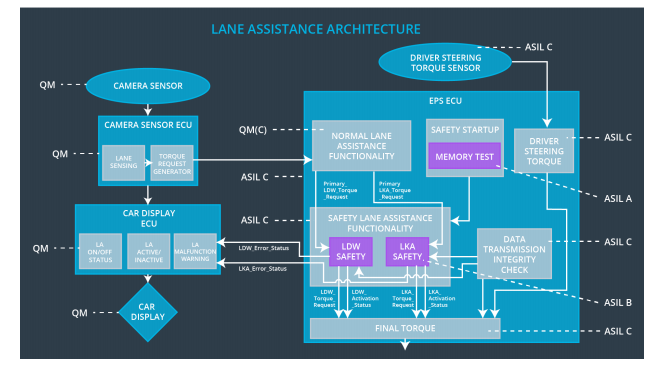
# Inputs to the Software Requirements and Architecture Document

## 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 | 50 ms | LDW safety | Set LDW Torque to zero |
| 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 | 50 ms | LDW Safety | Set LDW Torque to zero |
| 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 | Set LDW Torque to zero |
| Technical  Safety  Requirement  04 | The validity and integrity of the data transmission for 'LDW\_Torque\_Request' signal shall be ensured. | C | 50 ms | Data Transmission Integrity Check | N/A |
| Technical  Safety  Requirement  05 | Memory test shall be conducted at start up of the EPS ECU to check for any faults in memory. | A | Ignition cycle | Memory test | Set LDW Torque to zero |

## 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 | 50 ms | LDW safety | LDW torque output is set to zero |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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\_Amplitude\_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=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 the “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** | **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 | 50 ms | 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 transmitted outside of the 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 02-02 |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Software Safety Requirement** | **ASIL** | **Allocation Software Elements** | **Safe State** |
| Software Safety Requirement03-01 |  |  |  |  |
| Software Safety Requirement03-02 |  |  |  |  |
| Software Safety Requirement03-03 |  |  |  |  |
| Software Safety Requirement03-04 |  |  |  |  |
| Software Safety Requirement03-05 |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Software Safety Requirement** | **ASIL** | **Allocation Software Elements** | **Safe State** |
| Software Safety Requirement 04-01 |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Software Safety Requirement** | **ASIL** | **Allocation Software Elements** | **Safe State** |
| Software Safety Requirement 05-01 |  |  |  |  |
| Software Safety Requirement 05-02 |  |  |  |  |
| Software Safety Requirement 05-03 |  |  |  |  |
| Software Safety Requirement 05-04 |  |  |  |  |

# 

# Refined Architecture Diagram

**[Instructions: Include the refined system architecture. Hint: The refined system architecture should include the system architecture from the end of the software and hardware lesson, including all of the ASIL labels.]**