

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 |
| 27/12/2018 | 1.0 | Manjunath Gasthi | Initial version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 

# 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

This document identify requirements for the software components at component level to

identify potential problems on software design and architecture that could lead to a violation of

safety goals. These Requirements are more detail oriented than the technical safety concept

requirements.

# 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 electronically  power steering Torque‘  component is below  ‚Max\_Torque\_Amplitude’. | C | 50 mS | LDW Safety | LDW Torque  Request  Amplitude  shall be set to  zero. |
| Technical  Safety  Requirement  02 | The validity and integrity of  the data transmission for  'LDW\_Torque\_Request’  signal shall be ensured. | C | 50 mS | LDW Safety | LDW Torque  Request  Amplitude  shall be set 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 | LDW Torque  Request  Amplitude  shall be set to  zero. |
| 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  Request  Amplitude  shall be set to  zero. |
| Technical  Safety  Requirement  05 | Memory test shall be  conducted at start up of the  EPS ECU to check for any  faults in memory. | C | Ignition  cycle | Data  Transmission  Integrity  Check | LDW Torque  Request  Amplitude  shall be set 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 preprocessed  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\_LD  W” (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” |  | TORQUE\_LIMITER | “limited\_LDW\_  Torq\_Req” = 0  (Nm=Newtonmeter) |
| 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 Software Safety  Requirement 01-01 and  Software Safety  Requirement 02-02. | C | LDW\_SAFETY\_OUTPU  T\_GENERATOR | LDW\_Torq\_Re  q= 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 | LDW Safety | LDW torque  output is set  to zero. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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  Software Safety Requirement  01-01-03–02) shall be protected  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 | LDW torque  output 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  \_ACTIVATION | 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  \_ACTIVATION | LDW\_Torq\_Req = 0 |
| 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 | N/A |
| 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  \_ACTIVATION | 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 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  displayECU. | C | LDW\_SAFET  Y\_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 | A | Ignition  Cycle | Memory Test | 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 | 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 | LDW\_SAFETY\_  INPUT\_PROCE  SSING | Activation\_status = 0 |

# 

# Refined Architecture Diagram

****