



# 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     |
|------------|---------|---------------|-----------------|
| 2017-12-26 | 1.0     | Sven Eriksson | Initial release |
|            |         |               |                 |
|            |         |               |                 |
|            |         |               |                 |
|            |         |               |                 |

# 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, you can use headings for each section and then go to Insert > Table of Contents. Microsoft Word has similar capabilities]

[Document history](#)

[Table of Contents](#)

[Purpose](#)

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

[Technical safety requirements](#)

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

[Software Requirements](#)

[Refined Architecture Diagram](#)

## Purpose

[Instructions: Answer what is the purpose of this document?]

The purpose of this document is to further break down the requirements and analysis of the system into software requirements that specify variable names, signal paths, and software protocols and mechanism. The output of this document is a set of requirement from which a software engineer should be able to write a program with the intended software architecture and according to these requirements.

## Inputs to the Software Requirements and Architecture Document

[Instructions:

### REQUIRED:

You are only required to develop this document for the LDW (lane departure warning) amplitude malfunction. So here, provide the technical safety requirements for the LDW amplitude malfunction as well as the refined system architecture diagram from the technical safety concept.

### OPTIONAL:

Expand this document to include software safety requirements for the LDW frequency malfunction as well. Go even further and document software safety requirements for the Lane Keeping Assistance (LKA) function as well.

]

## 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 |
|----|------------------------------|------|------------------------------|-------------------------|------------|
|    |                              |      |                              |                         |            |

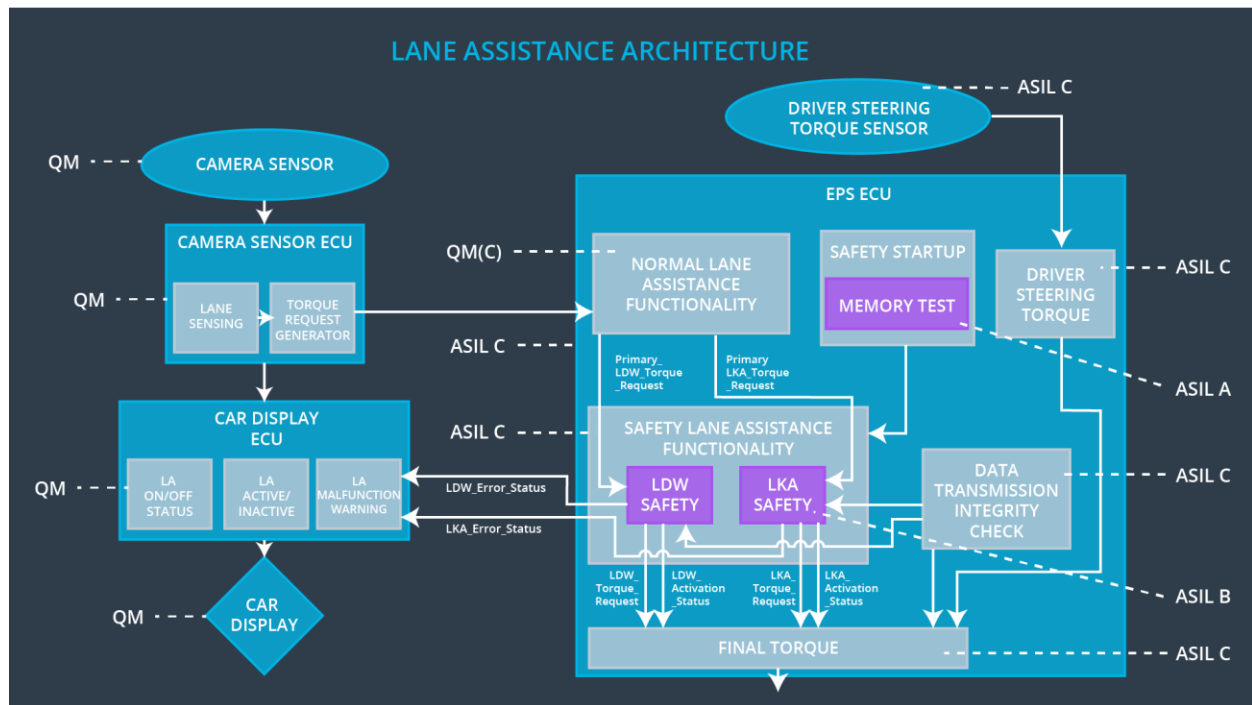
|                                 |   |   |                |                                   |  |
|---------------------------------|---|---|----------------|-----------------------------------|--|
| Technical Safety Requirement 01 | The LDW safety component shall ensure that the amplitude of the 'LDW_Torque_Request' sent to the 'Final electronic power steering Torque' component is below 'Max_Torque_Amplitude' | C | 50ms           | 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 | 50ms           | Data transmission integrity check | '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 | 50ms           | 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 | 50ms           | 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 EPC ECU to check for any faults in memory.  | A | ignition cycle | Memory test                       | 'LDW_Torque_Request' amplitude shall be set to zero. |

## Refined Architecture Diagram from the Technical Safety Concept

[Instructions:

REQUIRED: Provide the refined system architecture diagram from the technical safety concept

]



## Software Requirements

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

[Instructions: Fill in the software safety requirements for the LDW amplitude malfunction technical safety requirements. We have provided the associated technical safety requirements. Hint: The software safety requirements were discussed in the text from the software and hardware lesson.

#### OPTIONAL:

#### CHALLENGE ONE

Develop software safety requirements for the Lane Departure Warning (LDW) frequency function and modify the system architecture as needed.

#### CHALLENGE TWO

Develop software safety requirements for the Lane Keeping Assistance (LKA) function and modify the system architecture as needed.

]

| ID | Technical Safety Requirement | A<br>S | Fault<br>Tolerant | Allocation to<br>Architecture | Safe State |
|----|------------------------------|--------|-------------------|-------------------------------|------------|
|----|------------------------------|--------|-------------------|-------------------------------|------------|

|                                 |   | I<br>L | Time<br>Interval |            |  |
|---------------------------------|---|--------|------------------|------------|--|
| 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             | LDW Safety | 'LDW_Torque_Request' amplitude shall be set to zero. |

| ID                                | Software Safety Requirement  | A<br>S<br>I<br>L | 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 LAF functionality" 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   | C                | LDW_SAFETY_OUTPUT_GENERATOR  | LDW_Torq_Req = 0 (Nm)                        |

|  |  |  |  |  |
|--|--|--|--|--|
|  | the “Final EPS Torque” component. Also see SofSafReq02-01 and SofSafReq02-02 |  |  |  |
|--|--|--|--|--|

| ID                                       | Technical Safety Requirement  | A<br>S<br>I<br>L | Fault<br>Tolerant<br>Time<br>Interval | Allocation to<br>Architecture           | Safe State  |
|--|---|------------------|---------------------------------------|---|---|
| Technical<br>Safety<br>Requirement<br>02 | The validity and integrity of the data transmission for 'LDW_Torque_Request' signal shall be ensured. | C                | 50ms                                  | Data<br>transmission<br>integrity check | 'LDW_Torque_Request'<br>amplitude<br>shall be set<br>to zero. |

| ID   | Software Safety Requirement  | A<br>S<br>I<br>L | Allocation Software<br>Elements | Safe State           |
|--|--|------------------|---------------------------------|----------------------|
| Software<br>Safety<br>Requirement<br>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 |                  | E2ECalc                         | LDW_Torq_Req= 0 (Nm) |
| Software<br>Safety<br>Requirement<br>02-02 | The E2E protection protocol shall contain and attach the control data: alive counter (SQC) and CRC to the data to be transmitted.  |                  | E2ECalc                         | LDW_Torq_Req= 0 (Nm) |



| ID                              | Technical Safety Requirement   | A<br>S<br>I<br>L | 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                | 50ms                         | LDW Safety                 | 'LDW_Torque_Request' amplitude shall be set to zero. |

| ID                                | Software Safety Requirement   | A<br>S<br>I<br>L | Allocation Software Elements | Safe State                                       |
|-----------------------------------|---|------------------|------------------------------|--|
| Software Safety Requirement 03-01 | 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_SAFETY_INPUT_PROCESSING), error_status_torque_limiter(TORQUE_LIMITER), error_status_output_gen(LDW_SAFETY_OUTPUT_GENERATOR) | C                | All                          | N/A  |
| Software Safety Requirement 03-02 | 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 Requirement 03-03 | 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        | N/A  |
| Software Safety Requirement 03-04 | 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_Torque_Request" is set to 0  | C                | All                          | LDW_Torque_Request = 0                           |

|  |   |   |                       |   |
|--|---|---|-----------------------|---|
| Software<br>Safety<br>Requirement<br>03-05 | 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<br>(LDW function deactivated) |
|--|---|---|-----------------------|---|

| ID                                       | Technical Safety Requirement  | A<br>S<br>I<br>L | Fault<br>Tolerant<br>Time<br>Interval | Allocation to<br>Architecture | Safe State   |
|--|---|------------------|---------------------------------------|-------------------------------|--|
| Technical<br>Safety<br>Requirement<br>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                | 50ms                                  | LDW Safety                    | 'LDW_Torque_Request' amplitude shall be set to zero. |

| ID   | Software Safety Requirement  | A<br>S<br>I<br>L | Allocation<br>Software<br>Elements    | Safe State |
|--|--|------------------|---------------------------------------|------------|
| Software<br>Safety<br>Requirement<br>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_ACTIVATION, CarDisplay ECU | N/A        |

| ID                                       | Technical Safety Requirement   | A<br>S<br>I<br>L | Fault<br>Tolerant<br>Time<br>Interval | Allocation to<br>Architecture | Safe State  |
|--|--|------------------|---------------------------------------|-------------------------------|---|
| Technical<br>Safety<br>Requirement<br>05 | Memory test shall be conducted at start up of the EPC ECU to check for any faults in memory. | A                | ignition<br>cycle                     | Memory test                   | 'LDW_Torque_Request'<br>amplitude shall be set to zero. |

| ID   | Software Safety Requirement  | A<br>S<br>I<br>L | Allocation<br>Software<br>Elements | Safe State            |
|--|--|------------------|------------------------------------|-----------------------|
| Software<br>Safety<br>Requirement<br>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<br>Safety<br>Requirement<br>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<br>Safety<br>Requirement<br>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<br>Safety<br>Requirement<br>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_PROCESSING        | Activation_status = 0 |

# 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.]

