MBSD Lab #3 A.Y. 2023/24

# Purposes

* Perform some parts of the Functional and Technical Safety Concept analysis, according to ISO26262, of a “one pedal controller” for a car.
* Implement some of the safety concepts in the Simulink model of the controller developed in Laboratory #2.
* Perform unit and integration tests on the implemented safety-related functionalities.

It is available an example of a Functional Safety Concept for the item Front Light Manager (FLM).

The deliverable, composed of

* the report (the following pages of this document)
* the Simulink models on where the safety concepts have been implemented
* all the needed files to replicate the software testing results

has to be provided as a .ZIP file up to **June 23rd at 23:59.** It shall also contain a brief report explaining the design of the controller using the following template.

It is sufficient that only one of the group members uploads it.

**Important hint:**

For the following analysis, consider as ASIL C all the safety goals related to unintended acceleration (those leading to an increase of the vehicle’s speed modulus) and as ASIL B the warnings to the driver and the unintended deceleration (those leading to a decrease of the vehicle’s speed modulus).

# Model-Based Software Design, A.Y. 2023/24

# Laboratory 3 Report

## Components of the working group (max 2 people)

* Matteo Gravagnone, s319634
* Danilo Guglielmi, s318083

Functional Safety Concept

One pedal

# Functional safety architecture

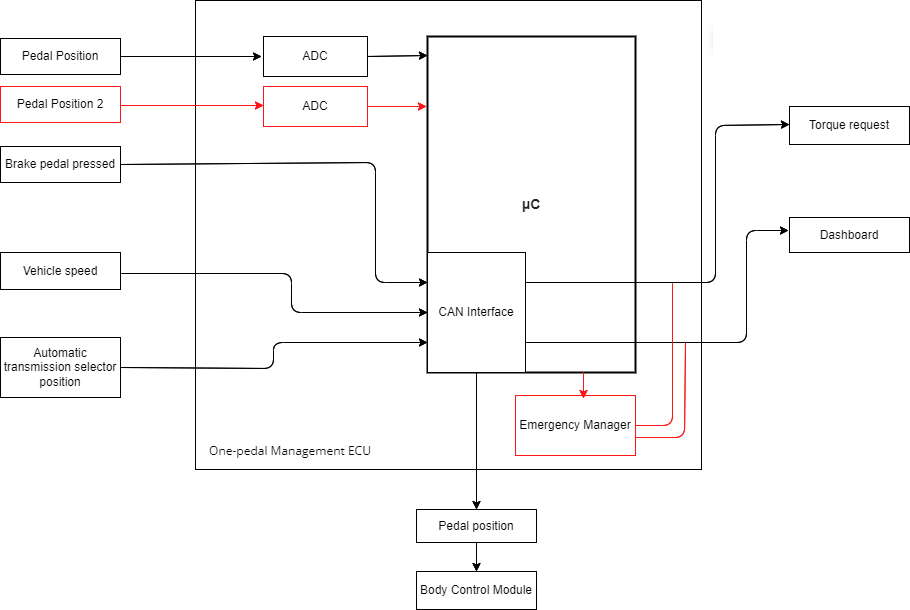


Figure 1 Functional safety architecture (from the safety concept)

# Attributes of the safety goals

*Fill in the attribute/parameters of the safety goal*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Safety goal** | **Attributes/Parameters of the safety goal** | | | | |
| Integrity (ASIL) | Safe state | Fault tolerance time | Warning concept | Degradation concept |
| SG1 | C | Switch to N | 100 ms | The driver must be notified on the dashboard | Motor is turned off |
| SG2 | B | Switch to N | 100 ms | The driver must be notified on the dashboard | Motor is turned off |
| SG3 | B | Warning of the malfunction | 100 ms | The driver must be notified on the dashboard | Warning system is deactivated |

# Functional (and technical) safety requirements and allocation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **Define functional safety requirements** | | **Allocation of requirements on systems and elements** | |
| **Safety requirements** | **Remark** | **If applicable, allocate the safety requirements to other Items / Systems** | **If applicable, allocate the safety requirements to equipment other technologies to minimize risk.**  **That could be e.g. hydraulic, mechanical equipment** |
| **Safety goals** | **The vehicle must not accelerate unintentionally** | SR1: If the pedal position interpreted is not valid (between 0 and 1), the torque request is set to 0. | No | No | Hydraulic braking system |
| SR2: The torque should be limited in the correct interval depending on the current state: if B between [-80, 80], if D between [0; 80], if R between [-40, 0], 0 if N or P. | No | No | Hydraulic braking system |
| **The vehicle must not decelerate unintentionally** | SR1: If the pedal position interpreted is not valid (between 0 and 1), the torque request is set to 0. | No | No | No |
| SR2: The torque should be limited in the correct interval depending on the current state: if B between [-80, 80], if D between [0; 80], if R between [-40, 0], 0 if N or P. | No | No | No |
| **The vehicle should be able to detect malfunctions in the warning system** | SR1: Monitor the functionality of the warning system periodically with specific diagnostic routines. | No | No | No |
| SR2: Notify the driver with a specific error message and activate a dashboard light if a warning malfunction is detected. | No | Warning lamp in the Cockpit-Display | No |

# ASIL preliminary architecture[[1]](#footnote-1)

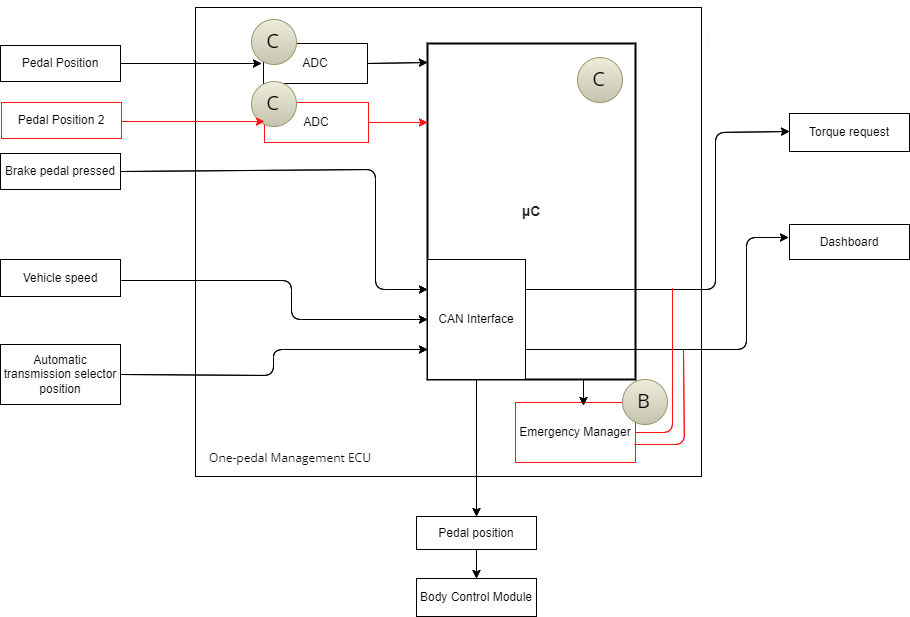


Figure 2 Preliminary architecture without ASIL decomposition

# Implementations[[2]](#footnote-2)

## Functional redundancies

The system can have at least 2 circuitries that can read the pedal position at the same time.

The μC can be replaced, in case of failure, by a simpler circuit, called Emergency Manager, that warns the driver of the failure and sets the torque request to zero. In our implementation, the μC is also responsible for these actions.

## Implemented plausibility checks

* *BrakePedalPressed* is a boolean value so plausibility checks are not needed.
* *ThrottlePedalPosition* needs to be in the range [0; 1] so if the position is not valid, a warning flag is set. A secondary *ThrottlePedalPositionRedundancy* input is also checked in the same range, a warning flag is set. If the discrepancy between the two variables is larger than 0.1 (10% of the pedal travel), a warning flag is set. An overall *PedalWarningFlag* is set to true if at least one of the previous three flags is set.
* AutomaticTransmissionSelectorState is seen as an integer value between 0 and 4. If it is not in this range, a warning flag called *SelectorWarningFlag* is set.
* TorqueRequest\_Nm is limited in the correct interval depending on the current AutomaticTransmissionState value: if B between [-80, 80], if D between [0; 80], if R between [-40, 0], 0 if N or P.
* If at least one between *PedalWarningFlag* and *SelectorWarningFlag* is set, the controller sets *Warning* to true,switches to N and sets the torque request to zero.

# Software testing

## Implemented unit tests

*Describe in English the test performed to verify the correct functionality of the safety mechanism implemented.*

## Implemented integration tests

*Describe, in English, the scenarios tested at the integration level to verify the proper integration between the various units implementing the safety mechanisms.*

1. See document 02-iso26262.pdf, slides 89, 90, 91, 92, 93. [↑](#footnote-ref-1)
2. In the ISO26262 the implementations are based on a document called *Technical Safety Concept*, but for simplicity we move straight from the *Functional Safety Concept* to software implementations.

   A guideline for the implementation phase can be found in the document 02-iso26262.pdf from slide 81, in particular slide 86. [↑](#footnote-ref-2)