Safety mechanisms for random ECU hardware failures in compliance with ISO 26262

Master of Science Thesis in Embedded Electronic System Design

DAVID JOHANSSON

PHILIP KARLSSON

Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden June 2015

Safety mechanisms for random ECU hardware failures in compliance with ISO 26262

DAVID JOHANSSON,
PHILIP KARLSSON,

Examiner: PER LARSSON-EDEFORS

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden June 2015

Abstract

The increasing complexity of today's automotive electronic systems makes it challenging for manufacturers to ensure a high safety level in their vehicles. As a response, the ISO 26262 functional safety standard will be introduced for heavy-duty vehicles in 2018. Therefore, the hardware and software solutions developed by Volvo Group Trucks Technology will need to be adapted to comply with this standard.

In addition to an analysis of ISO 26262, this thesis provides a case study of how the Volvo Engine Brake (VEB) can be adapted to comply with the standard. The analysis is focused on the electronic hardware of the engine control unit, and examines various safety mechanisms to improve the current system. The hazard of unwanted activation of the engine brake function is estimated to have ASIL C - the second most critical safety level. To comply with the requirements of ASIL C, the peripheral circuits of the engine brake should include both low and high-side MOSFET switches. Although a hardware-based diagnosis solution for actuator failures is presented, the study shows that a software-based safety mechanism is sufficient, which reduces the amount of extra hardware required. Additionally, if the inputs to the engine brake application are considered to be safety critical in a full evaluation, redundant sensors are required to meet the targets for ASIL C. A number of the solutions proposed in the concept for compliance with the standard are implemented and verified through a prototype.

Acknowledgements

We would like to express our sincere gratitude to Agne Holmqvist and Julianus Larson who have been our supervisors at Volvo Group Trucks Technology. They have assisted us throughout the thesis work and provided us with valuable feedback. Also, we would like to thank Erik Svangård and his group for giving us the opportunity to perform our thesis work at their department. Finally, we would also like to thank Lena Peterson, our supervisor at Chalmers, who has helped us with questions of academic nature.

David Johansson, Philip Karlsson, Göteborg, 3 June 2015.

Contents

Chapter 1

Introduction

The increasing complexity in today's automotive electronic systems has made it more challenging for manufacturers to ensure a high safety level in their vehicles. To give manufacturers a common means to measure and document the safety level of their systems, the ISO 26262 standard was released in November 2011 [1]. Previously, the generic functional safety standard for electrical and electronic (E/E) systems, IEC 61508, was applied by some manufacturers. However, there were complications when developing the special applications of automotive E/E systems in accordance to this standard. ISO 26262 is an adaptation of IEC 61508 intended to give automotive manufacturers a more tailored standard for achieving product safety. Currently, ISO 26262 applies only to series production passenger cars with a maximum gross vehicle weight of up to 3500 kg. A second revision of the standard, which will also include heavy-duty vehicles, is under development and is planned for release in 2018. This release drives companies such as truck manufacturers to begin planning and researching how to adapt their processes and products to comply with ISO 26262.

One company that needs to begin considering ISO 26262 is Volvo Group Trucks Technology (Volvo GTT), for which this project was performed. The Powertrain Engineering division at Volvo GTT is developing hardware and software solutions for engine control, after-treatment control, transmission control and hybrid control Electronic Control Units (ECUs). In the future, all of these systems will have to be developed according to ISO 26262. For this project, the engine braking system is used in a case study to investigate the ISO 26262 standard with regards to the Volvo GTT platform. In heavy-duty vehicles, normal brakes are generally assisted by one or several engine braking mechanisms. In the current Volvo GTT platform, two mechanisms are implemented to supply engine braking: exhaust back pressure and Volvo Combustion Brake (VCB). This system is assessed to be safety critical due to the fact that a sudden unexpected activation of the engine brake may result in a hazardous situation in which people may be injured. Since Volvo GTT has yet to begin incorporating ISO 26262 into their systems, the aim of this thesis work is to provide an initial case study that analyzes what solutions are required for the engine brake system to comply with the standard. The project focuses on the ECU hardware and on the software required to implement safety-critical hardware mechanisms.

1.1 Context

Although there exist publications regarding development in accordance to the current version of ISO 26262 [2][3], Volvo GTT systems are unique and therefore no off-the-shelf solutions are available. Several publications discuss the subject from a process-focused viewpoint by describing ISO 26262 and exemplifying how it can be used in product development such as in [4] and [5]. The examples of the standard being applied in actual development projects are fewer and often limited in detail such as in [6]. This scarceness may be due to the confidentiality of most industry projects. In general, this report is supported by publications discussing the ISO 26262 development process, but its results are mostly based on analysis of the standard's documentation in combination with current Volvo GTT systems and state-of-the-art ECU components such as the Freescale Qorivva [7] and Infineon Aurix [8] microcontrollers.

1.2 Aim

The aim of this thesis work was to provide an analysis of the current hardware architecture of the Volvo GTT engine controller to propose changes in the hardware (HW) and software (SW) implementation in support of the ISO 26262 standard. The project was intended as an initial step towards implementing the changes required for Volvo GTT products to fulfill the ISO 26262 standard in 2018. Since all Volvo GTT subsystems will be required to meet the standard, the aim was to present theoretical and practical solutions gained through the analysis of the engine braking subsystem together with verification methods in such a way that they could be generalized to other subsystems. The solutions which required testing were demonstrated through a proof-of-concept prototype based on next-generation ECUs and existing ECU hardware and software.

1.3 Problem description

To reach the expected conclusions regarding the engine brake system, this project included an analysis of the current HW and SW implementation in the EMS2.3 [9, 10, 11, 12] engine control ECU with respect to the ISO 26262 functional safety standard. The current light-duty edition was used as reference for the analysis, as a revision for use with heavy-duty vehicles had not yet been published. The problem solved in the project is generalized in Figure 1.1. The structure shown in the figure describes the safety concept of ISO 26262, which provides abstraction levels and a work flow to use when developing products according to the standard.

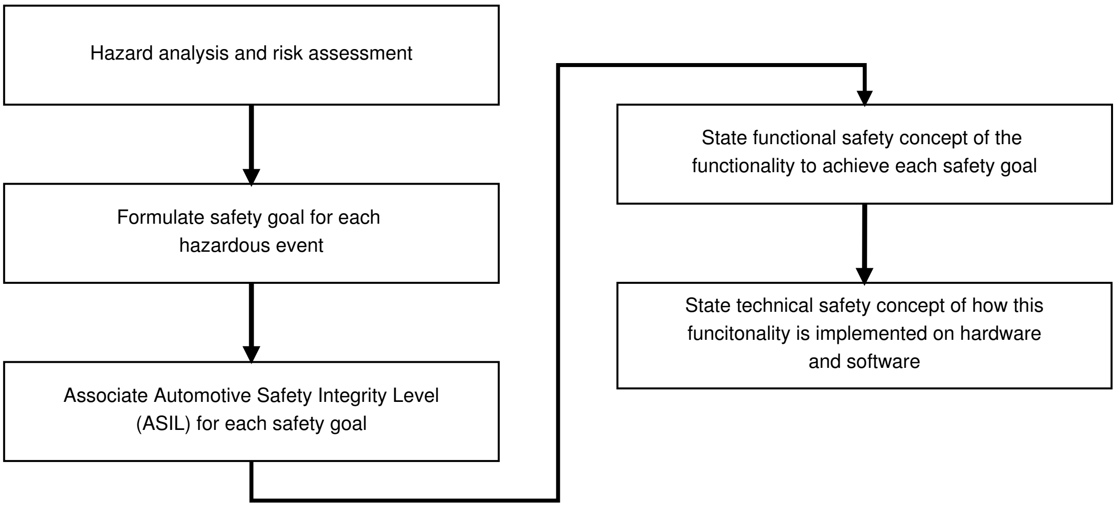

Hazard analysis and risk assessment
Formulate safety goal for each hazardous event
Associate Automotive Safety Integrity Level (ASIL) for each safety goal
State functional safety concept of the functionality to achieve each safety goal
State technical safety concept of how this funcitonality is implemented on hardware and software

Figure 1.1: The ISO 26262 safety concept.

The ISO 26262 safety concepts are hierarchical and each step in the figure provides the requirements for its subsequent step. The hazard analysis and risk assessment phase addresses the hazards and hazardous events, connected to a certain system (an item), that should be prevented, mitigated or controlled. The main objective in this analysis is to identify and categorize the hazards that could trigger malfunctions in the item. Based on this analysis, the next step includes the formulation of safety goals for each hazard and hazardous event. The safety goals are associated with an Automotive Safety Integrity Level (ASIL) based on three parameters: the severity of a malfunction, the probability of exposing a malfunction and the controllability of the situation in case of a malfunction occurring. The next step in the ISO 26262 work flow composing the project problem is to create a concept which states how each safety goal is ensured on a functional abstraction level. This concept, in turn, is used for developing a technical safety concept which states how the functional safety concept is implemented on system level by hardware and software. The technical safety concept is used for designing the actual hardware and software implementation with regards to the safety requirements. For the engine braking mechanisms, these requirements may for example be specified to include redundancy when measuring sensor values, accuracy on pulse-width modulation (PWM) pins or fault-tolerant software algorithms. In addition to the work flow in Figure 1.1, the project included the development of a prototype as a proof of concept. Tests and verification of this prototype were performed to verify that the design complied with ISO 26262.

To keep focus on the aim of the project, the highest priority goal was to provide the technical concepts and implementation alternatives required for compliance with the standard. Consequently, the prototype was not meant as the main result of the project, but was intended as a tool for demonstrating some of the suggested implementation options. Thus, the prototype did not include all suggested HW/SW solutions. Similarly, the early stages such as the hazard analysis were important mostly as a prerequisite for the HW/SW requirements. A full hazard analysis is a significant task and was beyond the scope of the project.

1.4 Method

The method used throughout the thesis project was based on the work flow of ISO 26262, which is described in Section 2.1. In short, this structure was based on a top-down approach in which an evaluation of the vehicle-level safety hazards was used as a base for formulating concepts and a proof-of-concept prototype, solving the problem of ensuring compliance with ISO 26262. As the standard incorporates the complete product life cycle, several assumptions and exemplifications were used to be able to perform the development steps, which in an actual product development project would be performed by other Volvo GTT departments or suppliers. The scope of the project is further discussed in Section 1.5.

A significant part of the information required throughout the project was acquired from the ISO 26262 documents [1], which in addition to requirements also comprise examples and guidelines regarding how to apply the standard. Chapter 2 covers the requirements and guidelines used when applying the standard. In addition to the standard, this chapter was also supported by several academic and industrial articles interpreting and exemplifying development with ISO 26262. Since documentation from other industry product development projects in compliance with ISO 26262 generally is confidential, the process of developing solutions for future Volvo GTT ECUs was mostly supported by internal Volvo GTT knowledge and documentation acquired from hardware component suppliers.

Lastly, a complete Volvo GTT tool chain and an EMS2.3 ECU were available for evaluating current solutions and to prototype new implementations. However, some of these tools were not used for the final proof-of-concept prototype as this was based on the Infineon Aurix Triboard development board. Instead, the safety manual and user manual of this board were used when implementing the concepts.

In order to test and verify the concepts, a test rig was developed. The rig supports injection of electrical faults, automatic triggering of an oscilloscope and reading of software state variables. Since the same rig was used for both the current implementation and the new prototype implementation the improvements could be measured by comparing the results of the two implementations.

1.5 Scope

The scope of the thesis was to perform a case study of how the E/E subsystems related to the engine braking mechanisms need to be modified to fulfill the requirements of ISO 26262. The main targets of the analysis were the peripherals that control the actuators and read the sensors used by the engine brake. Consequently, this thesis does not provide an analysis of the actual actuators and sensors. Even though the engine brake mechanism is described through several software layers, only the low-level software that interfaces directly with hardware peripherals is covered by the analysis. The thesis is focused around part 5 of the standard [13], which covers specification of hardware safety requirements, hardware design, hardware architectural metrics and evaluation of safety goals. Therefore, no formal analysis of the software was performed, and implementation alternatives suggested in the thesis were aimed to improve diagnosis of hardware and to implement hardware safety mechanisms. Similarly, in a full development project, application engineers would have to perform an analysis of the application-level software to determine which failures of inputs and outputs could lead to a violation of the safety goal. Without this analysis it cannot be determined which specific safety mechanisms are required for each sensor. As further argued in Section 4.4.1, it is assumed that the exhaust manifold pressure sensor is the only critical sensor to the engine brake application. However, the analysis of which safety mechanisms that required for this sensor was intended to be generalizable to other sensors considered as safety critical in future development projects.

As mentioned, ISO 26262 covers the complete product life cycle from planning to decommissioning. Throughout the development steps, the standard has requirements for how each step should be verified and documented. This thesis does not include formal verification and documentation as stated by ISO 26262, but includes the steps required to support the suggested solutions with regard to the project goals.

1.6 Organization of the thesis

Chapter 2 introduces the current revision of the ISO 26262 standard. This chapter is intended to provide knowledge regarding the concepts and definitions which are included in the scope of the thesis work. Chapter 3 presents the fundamental principles of the VEB. In Chapter 4, a concept for compliance with the standard is presented based on the theory in previous chapters. The proposed concept is discussed in Chapter 5 and, finally, a conclusion of the thesis work results is presented in Chapter 6.

Chapter 2

The ISO 26262 standard

The functional safety standard ISO 26262 [1], currently applicable for light-duty vehicles, will be applicable also for heavy-duty vehicles through a revision expected to be released in 2017/2018. ISO 26262 is an adaptation of the functional safety standard IEC 61508 created with the purpose of giving automotive manufacturers a common guideline for ensuring that their increasingly complex electric/electronic (E/E) systems are safe. The standard is risk-based, meaning that the risk of hazardous operational situations are assessed and used for determining the required safety measures. The types of hazards considered are the ones caused by malfunctioning behavior of electronic, electrical or programmable safety-critical systems. An automotive safety life cycle is defined by the standard that covers management, development, production, operation, service and decommissioning, and that provides safety requirements for these stages. The ten parts of the standard are divided into subparts called clauses, which in turn contain a number of requirements.

For determining the necessary safety requirements for a certain system, ISO 26262 defines a risk classification scheme called Automotive Safety Integrity Level (ASIL). The ASIL is established through a risk analysis in which hazards connected to the system are assessed. The subsequent sections describe the general work flow defined by the standard, and details regarding the parts used for this project.

2.1 Work flow

ISO 26262 provides requirements for the safety life cycle of a product; a general work flow is described and further elaborated on in the different parts of the standard. The standard's safety life cycle is shown in Figure 2.1 [14]. The numbers in the figure are references to the ISO 26262 parts and clauses. Tailoring of the safety life cycle is allowed, but a proper rationale for compliance with the standard should be documented. The result of a step in the life cycle is referred to as a work product and is used as input for the subsequent step.

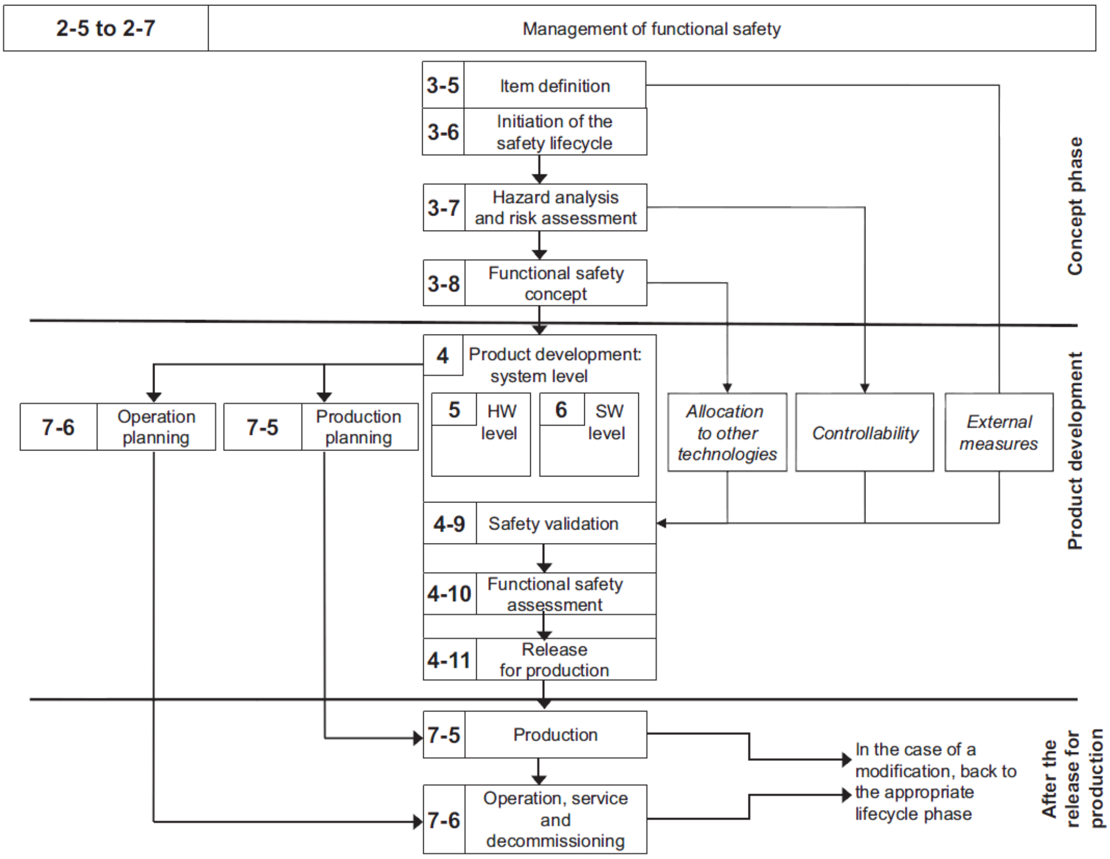

2-5 to 2-7
Management of functional safety
3-5
Item definition
3-6
Initiation of the safety lifecycle
3-7
Hazard analysis and risk assessment
3-8
Functional safety concept
Concept phase
4
Product development: system level
5
HW level
6
SW level
7-6
Operation planning
7-5
Production planning
Allocation to other technologies
Controllability
External measures
Product development
4-9
Safety validation
4-10
Functional safety assessment
4-11
Release for production
7-5
Production
7-6
Operation, service and decommissioning
In the case of a modification, back to the appropriate lifecycle phase
After the release for production

Figure 2.1: The ISO 26262 safety life cycle [14].

2.2 Item definition

The first objective in the ISO 26262 work flow is to define and describe the item to be developed and certified according to the standard. The item definition requires documentation of functional requirements, dependencies, environmental conditions, legal requirements and interaction with other items. The item definition should provide an adequate understanding of the item to serve as a base for the activities in subsequent phases. Both functional and non-functional requirements should be considered together with the dependencies between the item and its environment. ISO 26262 Part 3 [15] clause 5.4.1 recommends that these requirements are derived from the following information:

- The functional concept, describing the purpose and functionality, including the operating modes and states of the item
- The operational and environmental constraints

- Legal requirements (especially laws and regulations), national and international standards
- Behavior achieved by similar functions, items or elements, if any
- Assumptions on behavior expected from the item
- Potential consequences of behavior shortfalls including known failure modes and hazards

The standard also specifies that the interaction with other items, its interfaces and boundaries are defined considering:

- The elements of the item
- The assumptions concerning the effects of the item's behavior on other items or elements, that is the environment of the item
- Interactions of the item with other items or elements
- Functionality required by other items, elements and the environment
- Functionality required from other items, elements and the environment
- The allocation and distribution of functions among the involved systems and elements and
- The operating scenarios which impact the functionality of the item

2.3 Hazard analysis and risk assessment

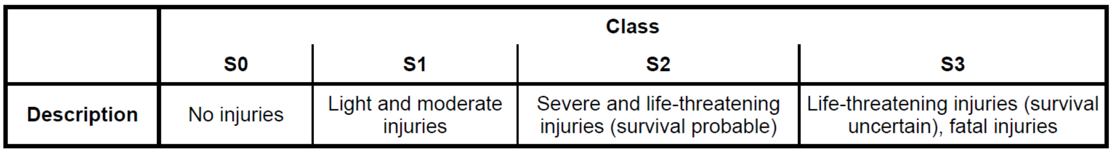
After the item has been defined as described in Section 2.2, ISO 26262 requires that hazard analysis and risk assessment (HARA) shall be performed based on this definition. The objective of the HARA is to determine safety goals of the item such that an unreasonable risk is avoided; i.e. the risk for a hazardous event is sufficiently low if the safety goal is fulfilled. The safety goals are established through systematic evaluation of hazardous events that may be caused by a fault in the item. Each safety goal is given an Automotive Safety Integrity Level (ASIL) that determines what ISO 26262 safety requirements that apply to the goal. Since the HARA is based on the item's functional behavior, the detailed design of the item does not necessarily have to be known [15]. The following paragraphs describe the HARA in more detail.

The first step in the hazard analysis is to determine and describe the operational situations and operating modes in which a malfunction in the item will result in a hazardous event. For example, circumstances such as road type, road shape, road conditions, environmental conditions, vehicle conditions and surroundings should be considered when searching for relevant situations. The operational situations are then combined with hazards to form hazardous events. The hazards are generally associated with a given function (or output) and a guideword, such as “service brakes” and “omission”. Hazards are defined by behaviors and conditions observed on the vehicle level. Thus, each hazard may have several potential causes related to the item’s definition, but these do not have to be considered in the HARA [15].

Based on the situations and hazards listed in the analysis, hazardous events should be determined such that they formulate relevant combinations of operational situations and hazards. For example, a hazardous event can be a combination of an operational situation in which the vehicle is parked and held by the parking brake on a sloped road and the hazard of omission of the parking brake. To determine the safety requirements associated with a hazardous event, the event is classified with respect to three measures: severity, probability of exposure and controllability. The combination of these measures results in an ASIL.

The severity measure is determined by the estimated severity of the potential harm caused by a hazardous event to any persons in the vehicle causing the event or to other persons at risk such as pedestrians or passengers of other vehicles. Table 2.1 [15] shows the four classes of severity that are used in the analysis.

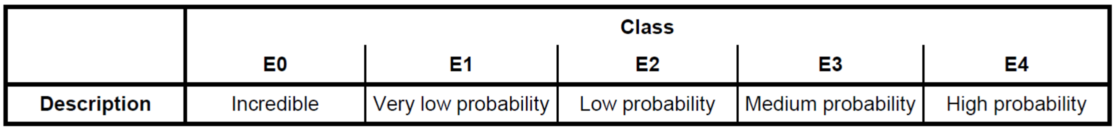
Table 2.1: Classes of severity [15].



	Class			
	S0	**S1**	**S2**	**S3**
Description	No injuries	Light and moderate injuries	Severe and life-threatening injuries (survival probable)	Life-threatening injuries (survival uncertain), fatal injuries

Table 2.2 [15] shows the five classes of probability used to define the probability of the vehicle being exposed to the operational situation of the hazardous event. Depending on the hazard type, the exposure level may be determined by considering the portion of driving time spent in a certain situation or the frequency of occurrence of the situation.

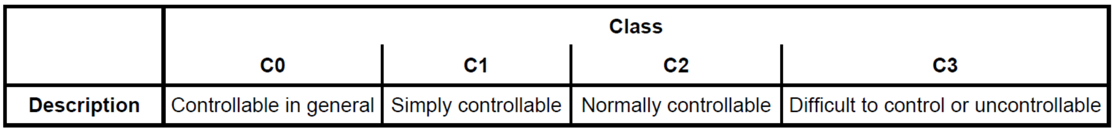
Table 2.2: Classes of probability of exposure regarding operational situation [15].



	Class				
	E0	**E1**	**E2**	**E3**	**E4**
Description	Incredible	Very low probability	Low probability	Medium probability	High probability

The third measure, controllability, describes to which extent the driver or other persons potentially at risk may be able to control the hazardous event. The classes for controllability are shown in Table 2.3 [15] and are based on the probability of the driver (or other persons) being able to control the event or otherwise being able to avoid harm.

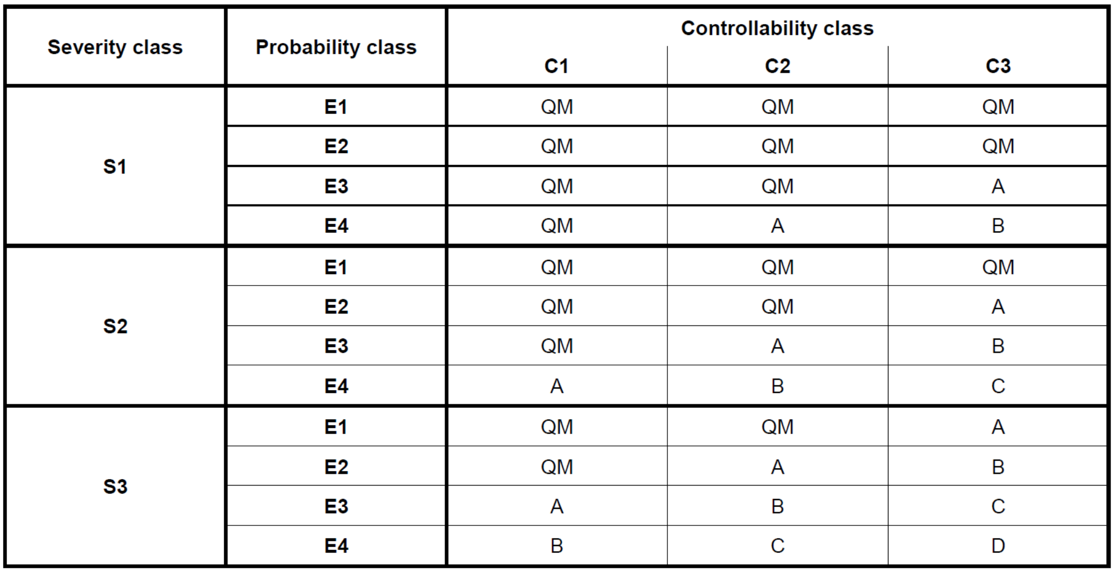
Table 2.3: Classes of controllability [15].



	Class			
	C0	**C1**	**C2**	**C3**
Description	Controllable in general	Simply controllable	Normally controllable	Difficult to control or uncontrollable

When the severity, probability of exposure and controllability have been established and backed up by suitable argumentation, Table 2.4 [15] is used to determine an ASIL for the hazardous event. Of the four ASILs that are defined, A represents the lowest safety integrity level and D the highest one. The fifth class present in the table, QM (quality management), denotes that there are no requirements to comply with ISO 26262. QM is also the result if any of the lowest levels S0, E0 or C0 are present from the previous analysis.

Table 2.4: Table for determining ASIL based on the severity, probability and controllability classes [15].



Severity class	**Probability class**	**Controllability class**		
		C1	**C2**	**C3**
S1	**E1**	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	**E1**	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	**E1**	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

Next, a safety goal is defined for each hazardous event with its ASIL given by the hazard analysis. The safety goal is expressed as a top-level safety requirement on the item in terms of functional objects; it does not include technological details. A simple example of a safety goal is: "Commission of braking shall not occur (ASIL B)". The example is formulated to be independent of the situation in which the hazard occurred. Several hazardous events may be covered by the same safety goal, or a safety goal can be formulated to combine a number of safety goals. In both cases, the ASIL of the safety goal will be the highest ASIL of any of the combined safety goals [15].

2.4 Functional safety concept

The functional safety concept derives functional safety requirements from the safety goals previously assigned to the item, and it serves as the foundation for stating the technical safety concept as is later described in Section 2.5. The solutions presented in the functional safety concepts should comply with the ASIL of each safety goal. It is stated by the standard documents [15] that the concept should address:

- Fault detection and failure mitigation
- Transitioning to a safe state
- Fault tolerance mechanisms, where a fault does not lead directly to the violation of the safety goal(s) and which maintains the item in a safe state (with or without degradation)
- Fault detection and driver warning to reduce the risk exposure time to an acceptable interval
- Arbitration logic to select the most appropriate control request from multiple requests generated simultaneously by different functions

At least one functional safety requirement must be specified for each safety goal. Operating modes, fault-tolerant time intervals, safe states, emergency operations and functional redundancies should be considered for each of these requirements. The fault-tolerant time interval is the time the system has to transition to a safe state after a failure has occurred. If a safe state is not reached within this interval, an emergency operation shall be specified.

It is recommended by the standard that the functional safety requirements are allocated to elements of the item according to architectural assumptions. If multiple requirements are allocated to the same element, then that element should be developed in accordance with the highest ASIL among these requirements. When the item consists of multiple systems, functional safety requirements shall be specified for the individual systems and their interfaces.

The functional safety concept may rely on elements of other technologies; in that case, the functional safety requirements implemented by these items should be derived and allocated to the corresponding element of the architecture. The interfaces to the elements with other technologies must be specified. The standard states that no ASIL should be assigned to elements of other technologies.

To develop a complete set of functional requirements, safety analysis such as failure mode and effects analysis (FMEA), fault tree analysis (FTA) or hazard and operability study (HAZOP) may be used. Part 8 clause 6 of the standard [16] states that the specification and management of safety requirements depend on the ASIL and that the requirements should be:

- Unambiguous, there is common understanding of the requirement

- Atomic, the requirements are formulated in such way that they cannot be divided into more than one safety requirement at the considered level
- Internally consistent: individual requirements do not contradict each other
- Feasible: the requirements can be implemented within the constraint of the item under development
- Verifiable: it is possible to verify that a certain requirement fulfills the safety goal

To validate the concept, the standard suggests that tests, trials or expert judgment may be used together with prototypes, studies and simulations.

2.5 Technical safety concept and requirements

The technical safety concept should refine the functional safety concept into preliminary architectural assumptions. When the functional safety concept has been refined it must be verified that the technical safety concept complies with the functional safety requirements. The technical safety concept is intended to detail the item-level functional safety concepts into system-level technical safety requirements. It should be specified with regard to external interfaces such as communication and user interfaces, to constraints such as environmental and functional constraints, and to system configuration requirements [17].

The technical safety concept should specify the response of the system when faults are induced that may lead to violation of the safety goals. Therefore, the safety mechanisms of the item should be specified to include measures that:

- Detect, indicate and control faults in the system itself
- Detect, indicate and control faults in external devices that interact with the system
- Enable the system to achieve or maintain a safe state
- Detail a concept for warning the driver and for handling functionality degradation
- Prevent faults from being latent

The standard also dictates that the following properties should be specified for each safety mechanism which enables the item to achieve or maintain a safe state.

- The transition to the safe state
- The fault tolerant time interval
- The emergency operation interval, if the safe state cannot be reached immediately
- The measures to maintain the safe state

When the technical concept has been stated, three resulting work products used in subsequent development steps should be available: a technical safety requirement specification, a system verification report and a validation plan.

2.6 Hardware development

The ISO 26262 hardware development process starts with planning the hardware development. This plan should be a part of the overall safety plan for the complete product safety process, and should include the measures and methods to be used for hardware design. The hardware development process is shown in Figure 2.2 [13]. After planning, the next step is to use the previous activities such as the technical safety concept and the system design specification to derive hardware safety requirements. These requirements typically specify the requirements and attributes of various safety mechanisms and are used when designing the hardware. However, they may also specify requirements not concerning safety mechanisms, such as requirements for target values for random hardware failures [3].

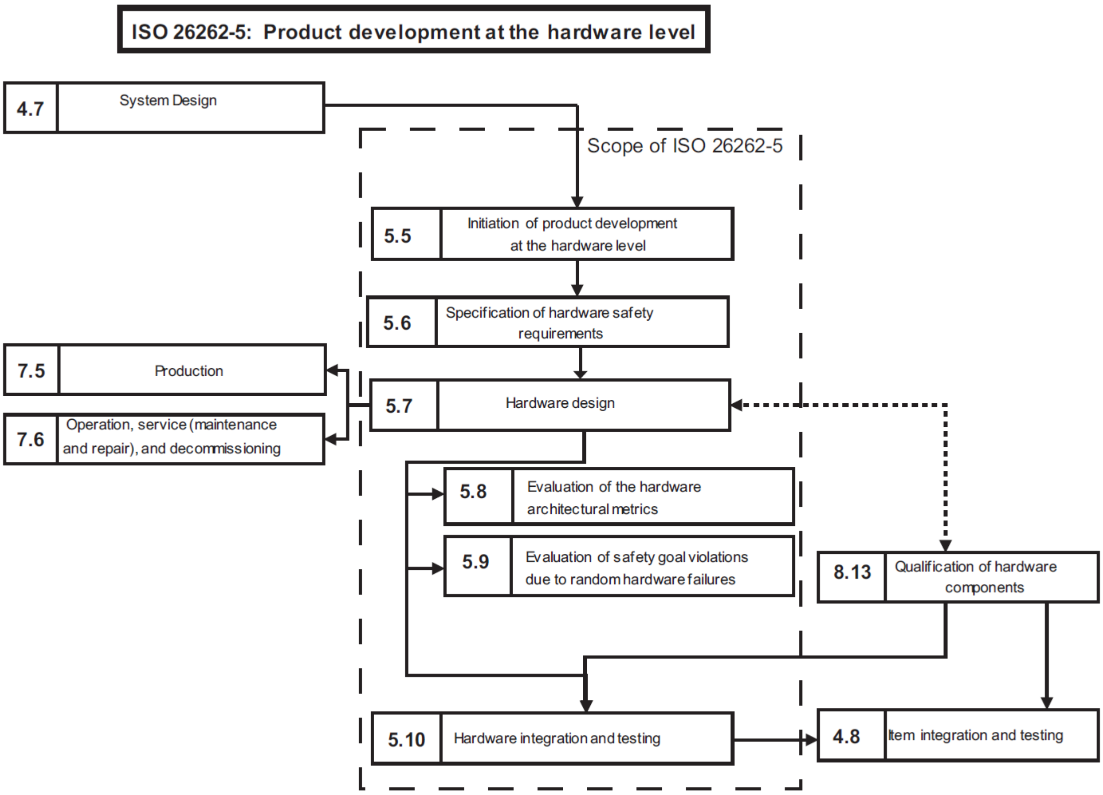

ISO 26262-5: Product development at the hardware level
4.7 System Design
Scope of ISO 26262-5
5.5 Initiation of product development at the hardware level
5.6 Specification of hardware safety requirements
7.5 Production
5.7 Hardware design
7.6 Operation, service (maintenance and repair), and decommissioning
5.8 Evaluation of the hardware architectural metrics
5.9 Evaluation of safety goal violations due to random hardware failures
8.13 Qualification of hardware components
5.10 Hardware integration and testing
4.8 Item integration and testing

Figure 2.2: The ISO 26262 hardware development process [13].

The hardware design process includes architectural design and hardware detailed design. The former represents all hardware components and their interactions, while the latter details the design at electrical schematics level by defining the interconnections and hardware parts composing the hardware components. Generally, safety requirements as well as non-safety requirements are included in this process to create a single integrated design.

In the creation of the hardware architecture, components are given the highest ASIL of the hardware safety requirements they implement. If a hardware element is composed of sub-elements with different or no ASIL, then each of these sub-elements should be treated with the highest ASIL among them if criteria for coexistence can be proved. These criteria are defined in ISO 26262 Part 9 [18] and detail how coexistence is derived from how sub-elements interact with each other with regard to violation of safety goals. In short, this means that if it can be proved that a lower-ASIL sub-element does not contribute to the violation of the safety goal of a higher-ASIL sub-element, then the sub-elements may keep their separate ASILs.

2.6.1 Hardware evaluation

When the detailed hardware design has been defined based on the architecture, the design should be analyzed regarding if it meets the requirements derived from its ASIL. It is possible that this process has to be performed in an iterative manner in which the safety analysis of the design requires going back to modify the hardware architecture or component design. Figure 2.3 [19] shows that the first step of the hardware evaluation is to obtain failure modes, failure rates and diagnostic coverage for the design. These measures are then used for evaluation of hardware architectural metrics and evaluation of safety goal violations due to random hardware failures. Lastly, the results from the analysis are compared to various target values for the different ASILs.

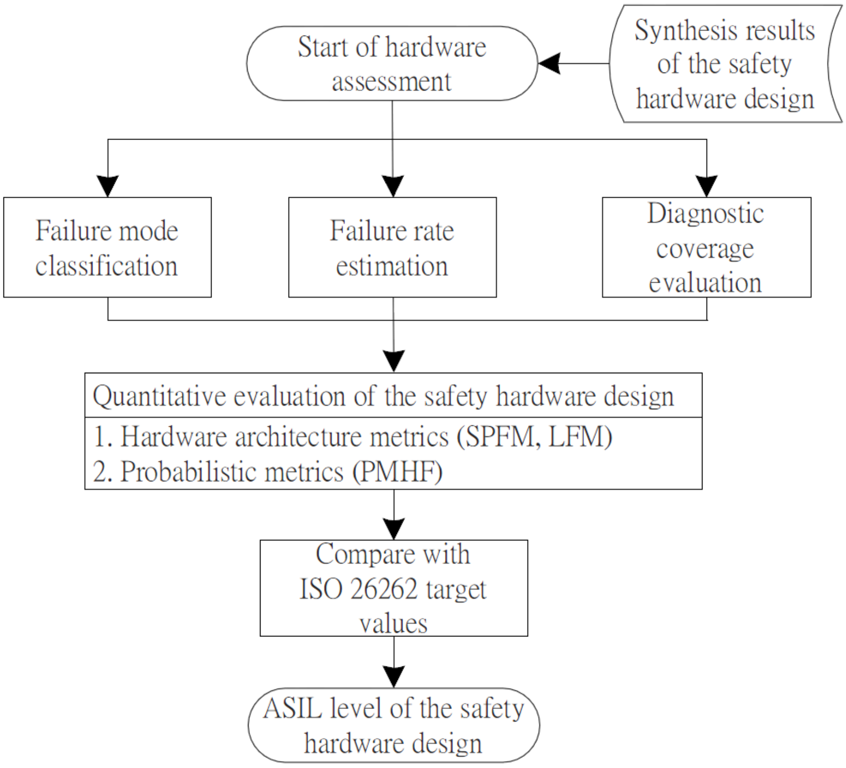

Synthesis results of the safety hardware design
Start of hardware assessment
Failure mode classification
Failure rate estimation
Diagnostic coverage evaluation
Quantitative evaluation of the safety hardware design
1. Hardware architecture metrics (SPFM, LFM)
2. Probabilistic metrics (PMHF)
Compare with ISO 26262 target values
ASIL level of the safety hardware design

Figure 2.3: The ISO 26262 hardware assessment process [19].

The analysis of the hardware design is based on a number of failure modes, which are defined as follows (λ denotes the failure rate for the corresponding failure mode):

- Safe fault (λ_S): a fault whose occurrence will not significantly increase the probability of violation of a safety goal.

- Single-point fault (λ_{SPF}): a fault in an element not covered by a safety mechanism and that leads directly to a violation of the safety goal.

- Residual fault (λ_{RF}): a fault in an element covered by a safety mechanism. The fault is not covered by the element's safety mechanism and leads directly to a violation of a safety goal.

- Multiple-point fault (λ_{MPF}): one fault of several independent faults that in combination leads to a failure. This fault can be classified as either:

  - Perceived ($\lambda_{MPF,P}$): This fault is undetected by any safety mechanism (within a prescribed time), but is perceived by the driver.

- Detected ($\lambda_{MPF,D}$): This fault is detected by a safety mechanism to prevent the fault from being latent within a prescribed time.
- Latent ($\lambda_{MPF,L}$): This fault is neither detected by a safety mechanism nor perceived by the driver.

The total failure rate of each safety-related hardware element can be expressed as:

$$\lambda = \lambda_S + \lambda_{SPF} + \lambda_{RF} + \lambda_{MPF} \tag{2.1}$$

The method for classifying faults into failure modes is described in Figure 2.4 [13]. In addition to the steps in the figure, a multiple-point fault can be classified as a safe fault if it is caused by more than two individual faults and no special circumstances exist that require the fault to be considered in the analysis.

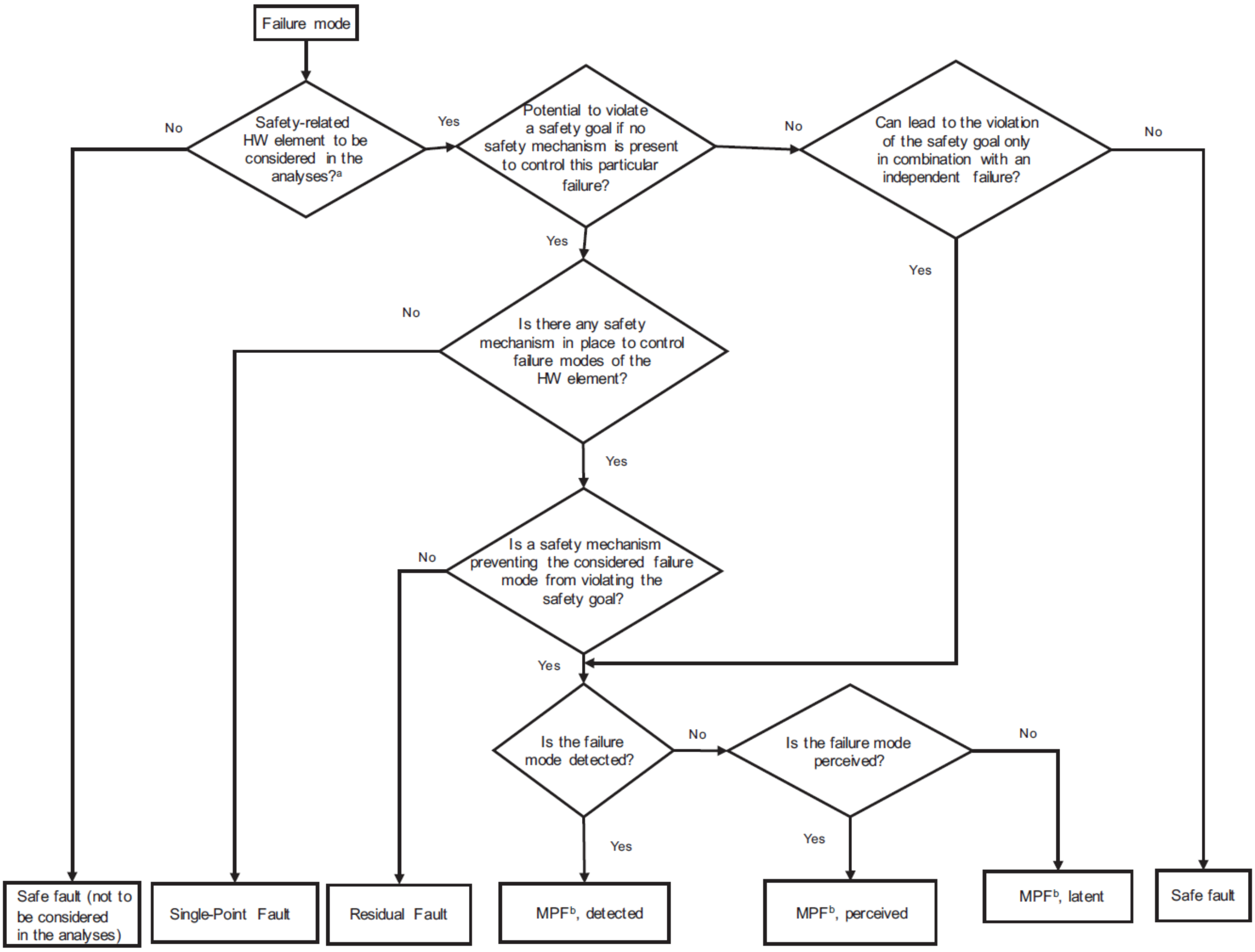

Failure mode
Safety-related HW element to be considered in the analyses?[a]
No
Yes
Potential to violate a safety goal if no safety mechanism is present to control this particular failure?
No
Yes
Can lead to the violation of the safety goal only in combination with an independent failure?
No
Yes
Is there any safety mechanism in place to control failure modes of the HW element?
No
Yes
Is a safety mechanism preventing the considered failure mode from violating the safety goal?
No
Yes
Is the failure mode detected?
No
Yes
Is the failure mode perceived?
No
Yes
Safe fault (not to be considered in the analyses)
Single-Point Fault
Residual Fault
MPF[b], detected
MPF[b], perceived
MPF[b], latent
Safe fault

Figure 2.4: Scheme to classify failure modes. MPF stands for multiple-point fault [13].

The diagnostic coverage of each safety mechanisms must be evaluated in order to find the faults that retain the possibility of violating a safety goal after safety measures have been included in the system. The diagnostic coverage with respect to residual faults is defined as the part of the failure rate (of faults of a single-point origin) in an element that is covered by a certain safety mechanism according to:

$$K_{DC,R} = (1 - \frac{\lambda_{RF,est}}{\lambda}) \cdot 100 \tag{2.2}$$

where $\lambda_{RF,est}$ is a conservative estimation of λ_{RF}. Similarly, the diagnostic coverage of a safety mechanism with respect to latent faults is defined as:

$$K_{DC,MPF,L} = (1 - \frac{\lambda_{MPF,L,est}}{\lambda}) \cdot 100 \tag{2.3}$$

where $\lambda_{MPF,L,est}$ is a conservative estimation of $\lambda_{MPF,L}$. ISO 26262 part 5 Annex D [13] includes guidelines regarding how to estimate the general diagnostic coverage of a safety mechanism. The purpose of these guidelines is also to suggest which safety mechanisms that may be suitable for each type of hardware element.

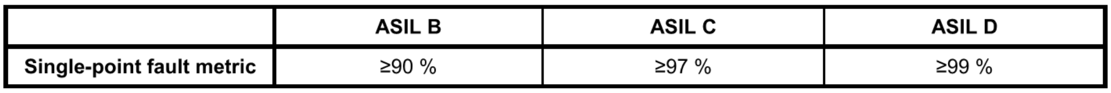
2.6.2 Hardware architecture metrics

When the fault mode classification, the failure rates and the diagnostic coverage of each hardware component have been determined, the hardware is assessed with regard to its ability to suppress failures. Two metrics are calculated to represent the performance of the architecture: single-point fault metric (SPFM) and latent fault metric (LFM). SPFM reflects the item's robustness to single-point and residual faults either by design or by safety mechanism coverage. A high SPFM indicates that the proportion of these faults is low. The metric is calculated as:

$$SPFM = 1 - \frac{\sum\limits_{SR,HW} (\lambda_{SPF} + \lambda_{RF})}{\sum\limits_{SR,HW} \lambda} = \frac{\sum\limits_{SR,HW} (\lambda_{MPF} + \lambda_{S})}{\sum\limits_{SR,HW} \lambda} \tag{2.4}$$

where $\sum\limits_{SR,HW} \lambda_x$ is the sum of λ_x of the safety related hardware elements of the item under consideration. Again, only the hardware elements of which failures have the potential to contribute significantly to the violation of the safety goal are considered for this metric. The calculated SPFM is compared to either derived target values from similar well-trusted design principles or to the target values in Table 2.5 [13] to obtain the achieved ASIL of the item.

Table 2.5: Targets for the single-point fault metric [13].



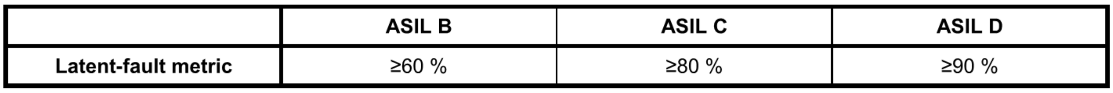
	ASIL B	ASIL C	ASIL D
Single-point fault metric	≥90 %	≥97 %	≥99 %

The latent fault metric represents the item's robustness for latent faults either by design, by safety mechanism coverage or by the driver recognizing that the fault exists before the safety goal is violated. Thus, a high LFM implies that the proportion of latent faults in the item is low. The LFM is calculated according to Equation 2.5.

$$LFM = 1 - \frac{\sum\limits_{SR,HW} \lambda_{MPF,latent}}{\sum\limits_{SR,HW} (\lambda - \lambda_{SPF} - \lambda_{RF})} = \frac{\sum\limits_{SR,HW} (\lambda_{MPF,perceived\ or\ detected} + \lambda_{S})}{\sum\limits_{SR,HW} (\lambda - \lambda_{SPF} - \lambda_{RF})} = \tag{2.5}$$

where $\sum\limits_{SR,HW} \lambda_x$ is calculated as described for Equation 2.4. It is generally accepted to use one drive cycle as detection interval for latent faults. The LFM is compared to either derived target values from similar well-trusted design principles or to the target values in Table 2.6 [13] to obtain the achieved ASIL of the item.

Table 2.6: Targets for the latent fault metric [13].



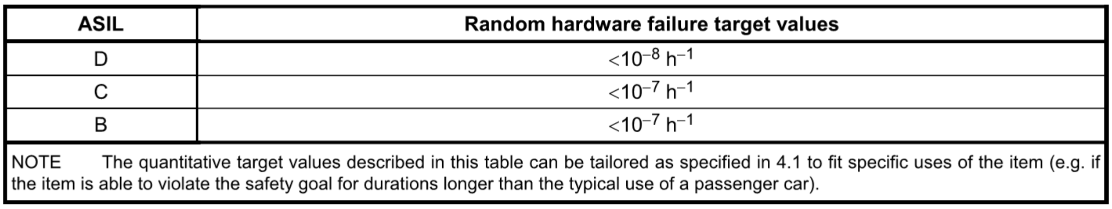
	ASIL B	ASIL C	ASIL D
Latent-fault metric	≥60 %	≥80 %	≥90 %

2.6.3 Random hardware failure metric

In addition to the architecture metrics evaluation, ISO 26262 requires that the design is evaluated with regards to the overall residual risk of violation of each safety goal due to random hardware failures of the item. The standard requires that one of two alternative methods is used to assess this risk. Both methods take into account single-point faults, residual faults and dual-point faults, including the coverage of safety mechanisms. Furthermore, exposure duration should be considered for dual-point faults. The first method requires that each cause of a safety goal violation due to random hardware failures is compared to a target failure rate class. However, this method is not used for this project and only the second method using the probabilistic metric for random hardware failures (PMHF) is described in detail.

The PMHF is calculated as the maximum probability of violation of each safety goal due to random hardware failures. This maximum value is compared to the target values in Table 2.7 [13] to derive the achieved ASIL. The target values in the table are expressed in terms of average probability per hour over the operational lifetime of the item.

Table 2.7: Targets for the probabilistic metric for random hardware failures (PMHF) [13].



ASIL	Random hardware failure target values
D	$<10^{-8}$ h^{-1}
C	$<10^{-7}$ h^{-1}
B	$<10^{-7}$ h^{-1}

NOTE The quantitative target values described in this table can be tailored as specified in 4.1 to fit specific uses of the item (e.g. if the item is able to violate the safety goal for durations longer than the typical use of a passenger car).

In contrast to the hardware architectural metrics calculation in which the complete item was considered, the failure rates included in the PMHF evaluation are only those that contribute to the violation of a certain safety goal. The relevant failure rates for each safety goal can be found using failure modes effects and diagnostic analysis (FMEDA) or fault tree analysis (FTA). As mentioned, exposure durations are taken into account in the analysis in addition to the failure rates for the different failure modes and the diagnostic coverage by safety mechanisms. The exposure duration is considered only for dual-point faults and depends on the safety mechanism covering the fault. For example, it can be the multiple-point fault detection interval associated with the safety mechanism, or the lifetime of the vehicle in case of a latent fault. Equation 2.6 describes how the value of the probabilistic metric for random hardware failures, M_{PMHF}, can be calculated considering the exposure duration. The equation assumes that an intended functionality (the mission block "m") is supervised by a safety mechanism "sm".

$$M_{PMHF} = \frac{\lambda_{m,RF} \cdot T_L + \lambda_{m,DPF} \cdot T_L \cdot 0{,}5 \cdot (\lambda_{sm,DPF,latent} \cdot T_L + \lambda_{sm,DPF,detected} \cdot \tau_{SM})}{T_L} \tag{2.6}$$

where

M_{PMHF} is the value for the PMHF metric

$\lambda_{m,RF}$ is the residual failure rate of the intended functionality

$\lambda_{m,DPF}$ is the dual-point failure rate of the mission block "m"

T_L is the vehicle lifetime

$\lambda_{sm,DPF,latent}$ is the latent dual-point failure rate of the safety mechanism

$\lambda_{sm,DPF,detected}$ is the detected dual-point failure rate of the safety mechanism

τ_{SM} is the multiple-point fault detection interval of the safety mechanism

If the detection interval τ_{SM} is very small, for example if the safety mechanism detects the fault within one driving cycle, the term $\lambda_{m,DPF} \cdot \lambda_{sm,DPF,detected} \cdot \tau_{SM}$ can be neglected, simplifying the formula into:

$$M_{PMHF} = \lambda_{m,RF} + 0{,}5 \cdot \lambda_{m,DPF} \cdot \lambda_{sm,DPF,latent} \cdot T_L \tag{2.7}$$

The factor 0,5 is due to the second order of failure of the dual-point faults. If the order for some reason is irrelevant, this factor can be omitted.

In addition to the above mentioned evaluation, the following requirement applies for ASIL C and D safety goals: a single-point fault occurring in a hardware part is only considered acceptable if certain dedicated measures are taken. Examples of such measures include hardware part over design and special test of materials to reduce the risk of single-point faults. Furthermore, similar measures should also be taken if a hardware part's diagnostic coverage with respect to residual faults is lower than 90%.

Chapter 3

Volvo Engine Brake

This chapter is aimed to give basic insight into how the Volvo GTT engine brake system works with the purpose of providing the knowledge required for understanding the engine brake case study of this thesis.

The main purpose of a truck's engine brake is to relieve the wheel brakes. The engine brake aims to maximize the energy loss in the engine during braking to achieve a highly controllable brake torque. Since this braking is applied before the gearbox in the powertrain, the engine brake is considered the primary retarder of the powertrain. Also, the engine brake torque is related to the engine speed and not to the speed of the vehicle. The Volvo Engine Brake (VEB) comprises of two separate brake mechanisms, an exhaust brake and a compression brake.

3.1 Exhaust Brake

The exhaust brake is located downstream of the turbocharger of the powertrain. It retards the vehicle by restricting the exhaust system, creating a backpressure to the cylinders to generate a negative torque. Since no combustion occurs in the cylinders during engine braking (no diesel is injected), the exhaust manifold pressure is low. A higher pressure is created by limiting the exhaust channel with a variable valve. This pressure limits the exhaust flow from the cylinder during the exhaust stroke (when the exhaust valve is open) resulting in a negative torque on the engine. In the VEB, a butterfly valve is used to create variable backpressure in the exhaust system and therefore also a variable engine brake torque. Figure 3.1 presents a photograph of a butterfly valve.

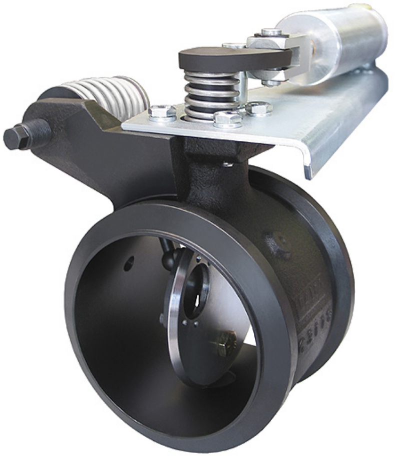


Figure 3.1: A butterfly valve for an exhaust brake.

The butterfly valve is pneumatic and opens in proportion to a supplied air pressure. The air pressure is delivered from an Air Valve Unit (AVU), which is controlled by the ECU. In order to request that a certain air pressure is delivered from the AVU, the ECU sets a PWM signal with a duty cycle proportional to expected pressure. The air pressure to the butterfly valve is adjusted with a closed-loop regulator based on a pressure sensor in the exhaust manifold.

3.2 Volvo Compression Brake

The diesel engines engineered at Volvo GTT are four-stroke internal combustion engines. In a four-stroke engine, the piston completes four separate strokes in a single thermodynamic cycle. The top illustration of Figure 3.2 presents the four strokes of a four-stroke engine.

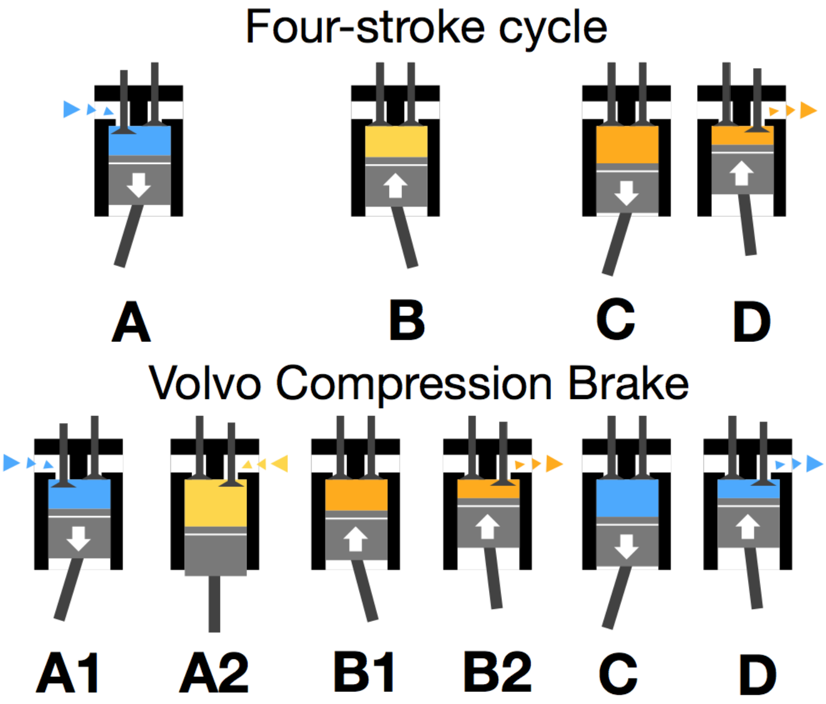


Figure 3.2: Schematic over the four-stroke cycle and the Volvo Compression Brake cycle.

The cycle starts with the intake stroke (A in Figure 3.2). In this stroke, air is drawn into the cylinder past an inlet valve. Following the intake stroke is the compression stroke (B in Figure 3.2). In this stroke both the inlet and the exhaust valves are closed as the piston returns to the top of the cylinder compressing the air. Diesel fuel is injected through a high-pressure injector in a specific sequence when the piston reaches the top end position, resulting in combustion of the air-fuel mix. The explosion carries the piston into the power stroke (C in Figure 3.2), in which the piston is forced downwards. Finally, when the piston has passed the bottom dead center position, the remains of the combustion are exhausted through the exhaust valve. This last stroke is referred to as the exhaust stroke (D in Figure 3.2).

The compression brake generates brake torque by opening the exhaust valves shortly during the intake and compression strokes. The bottom illustration of Figure 3.2 presents the principle of the compression brake. Since the compression brake is used concurrently with the exhaust brake, a backpressure is present in the exhaust system. The compression brake opens the exhaust valves for a short period at the bottom dead center of the intake stroke (A2 in Figure 3.2). Due to the high pressure in the exhaust system, high pressure air flows backwards in to the cylinder. Consequently, the higher pressure inside the cylinder caused by the extra valve opening results in an increased braking power during the compression stroke. At the end of the compression stroke, the exhaust valves are opened once more to release the compressed air (B2 in Figure 3.2); this is where combustion would normally occur. As a result, a vacuum is created inside the cylinder during the power stroke (C in Figure 3.2), which results in brake torque. The exhaust valves are finally opened during the exhaust stroke as in a normal four-stroke cycle.

The compression brake is realized by the addition of an extra cam lobe and rocker arm for each cylinder. When the compression brake is activated, these rocker arms push the exhaust rocker arms to open for the two mentioned extra periods. The exhaust brake rocker arms connect with and control the normal exhaust rocker arms when the cam shaft oil pressure is increased by an ECU-controlled oil valve.

Chapter 4

A concept for compliance with ISO 26262

This section presents a concept for how the Volvo Engine Brake can be designed in compliance with ISO 26262. Based on the theory in Section 2, item definition, hazard analysis, and technical and functional safety concepts are presented for the engine brake case study. An exemplifying hazard analysis is performed, which serves as the base for determining the ASIL of the system. The estimated ASIL is used in subsequent sections where a functional safety concept is stated and refined into a technical safety concept for which hardware metrics evaluations are performed. Finally, a number of the suggested technical concepts are further investigated through the implementation and verification of a prototype.

4.1 Item definition

As stated in Section 2.2, the first step in the ISO 26262 work flow is to define and describe the item to be developed. The safety-critical item studied in this thesis project is the VEB described in Section 3. The engine brake is controlled by application software running on the engine-control ECU. When receiving engine brake demands, the application software calculates the appropriate braking torque and how to produce it using a certain amount of exhaust and compression brake. Figure 4.1 presents an overview of the VEB and its subsystems.

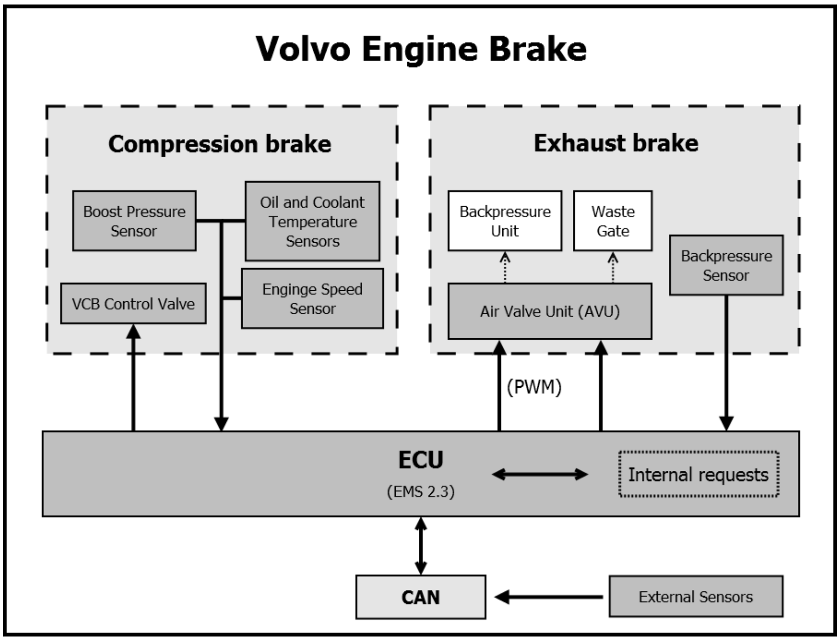


Figure 4.1: The Volvo Engine Brake item.

The application software on the ECU activates the engine brake after receiving external requests over CAN or after requests are generated internally. To activate the brakes, the application software communicates with the platform software on the ECU, which provides drivers for actuators and sensors.

In its current implementation, the compression brake, called the Volvo Compression Brake (VCB), is controlled using the circuit presented in Figure 4.2.

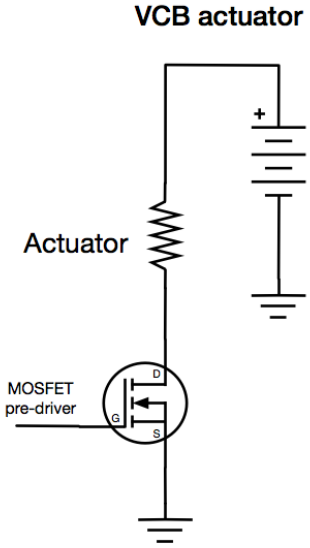


Figure 4.2: Schematic presenting the current implementation of the Volvo Compression Brake actuator control.

The compression brake is activated by driving a GPIO pin on the ECU connected to a low-side MOSFET pre-driver. In turn, this pre-driver drives the gate voltage of the low-side MOSFET in Figure 4.2 which closes a circuit supplying the solenoid of the VCB control valve. This operation activates the compression brake. The application software relies on data from a number of sensors for deciding whether or not to activate the VCB. The sensors connected directly to the engine ECU are the oil and coolant temperature sensors, the engine speed sensor and the boost pressure sensor. Each sensor's value is read by sampling a pin on the ECU.

The ECU activates the exhaust brake by generating a PWM signal to a GPIO pin, which feeds a low-side MOSFET through a pre-driver. The MOSFET is connected to the input of an air valve unit (AVU) which supplies the butterfly valve with air pressure proportional to the duty cycle of the PWM signal. The butterfly valve can thus be regulated linearly by controlling the air pressure from the AVU. However, the braking torque applied by the butterfly valve is not necessarily linearly proportional to the position of the valve, but depends on other conditions such as engine speed. Therefore, the application software uses the exhaust manifold pressure measured by the back pressure sensor for closed-loop regulation of the valve.

The parts of the engine brake system under scrutiny of ISO 26262 are:

- The EMS ECU
- The control area network (CAN) interface that transmits and receives data to and from other systems in the vehicle.

- The application software on the ECU
- The platform software on the ECU
- The actuators used to control the butterfly and VCB control valves
- The sensors used to sample the required data
- The power supply of the ECU

In this thesis project, we use a subset of the listed parts to define the item to be analyzed. The application software on the ECU is assumed to be a safety element out of context (SEooC), which means that the functionality that resides in this software is assumed to produce correct results given correct input. The CAN interface of the ECU is also considered an SEooC as the scope of this thesis work is to examine low-level hardware/software embedded systems of the engine brake in relation to ISO 26262. Thus, the following parts constitute the item for which the safety goals and solutions presented in subsequent sections are valid provided that the SEooC parts are fully functional:

- The EMS ECU
- The platform software on the ECU
- The actuators used to control the butterfly and VCB control valves
- The sensors used to sample the required data

4.2 Hazard analysis

As described in Section 1.5, this project's scope does not include a complete hazard analysis. However, this section provides an exemplifying analysis based on the theory described in Section 2.3 with the purpose of providing a context for the subsequent steps in the ISO 26262 work flow. A full analysis to prove compliance with the standard is an extensive task, but non-formal estimations can be made to predict an ASIL.

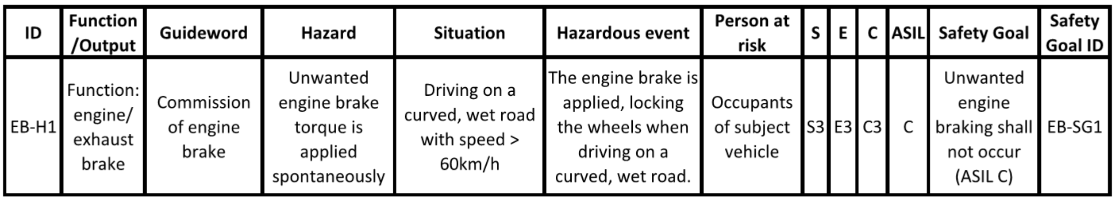
The main hazard connected to the engine brake system is unwanted activation of the engine brake. The reason for this is that the engine brake can lock the vehicle's wheels under certain conditions, possibly leaving the vehicle uncontrollable. This hazard is the reason for why the system is estimated to have a high ASIL. Table 4.1 shows an example hazardous event with the corresponding safety goal for the engine brake item. The severity of the hazardous event where the engine brake is applied spontaneously while driving on a wet road is estimated to S3. This class is chosen because there is a risk that the wheels could lock and the vehicle skids off the road or into another vehicle, potentially causing severe or fatal injuries. A full severity analysis of the event could also result in an S2 classification.

The probability of exposure to the operational situation is estimated to correspond to class E3 because the ISO 26262 guidelines suggest that the probability of driving on a wet road is within the limits for this class. The added condition that the vehicle is driving on a curved road is by intuition more common and thus the wet road condition is the determining factor. One could argue that, especially in countries such as Sweden or the United Kingdom, wet roads could require the highest exposure class E4.

The controllability of the situation is graded to C3 because of the ISO 26262 C3 recommendation for the similar situation “failure of ABS when braking on low friction road surface while executing a turn”. Locked wheels due to spontaneous activation of the engine brake should by intuition result in a worse scenario than failure of the ABS.

As a result, the safety goal may get S2-S3, E3-E4 and C3 as possible classes. This implies an ASIL ranging from B to D. An ASIL below C would however seem unlikely. Both Volvo GTT and TRW Automotive have estimated an ASIL C in their initial analyses, and this is what Table 4.1 exemplifies. For the remaining steps in the ISO 26262 work flow for this project, safety goal EB-SG1 is assigned to be ASIL C. However, additional requirements required for ASIL D are presented alongside the ASIL C implementation.

Table 4.1: Hazard analysis.



ID	Function /Output	Guideword	Hazard	Situation	Hazardous event	Person at risk	S	E	C	ASIL	Safety Goal	Safety Goal ID
EB-H1	Function: engine/ exhaust brake	Commission of engine brake	Unwanted engine brake torque is applied spontaneously	Driving on a curved, wet road with speed > 60km/h	The engine brake is applied, locking the wheels when driving on a curved, wet road.	Occupants of subject vehicle	S3	E3	C3	C	Unwanted engine braking shall not occur (ASIL C)	EB-SG1

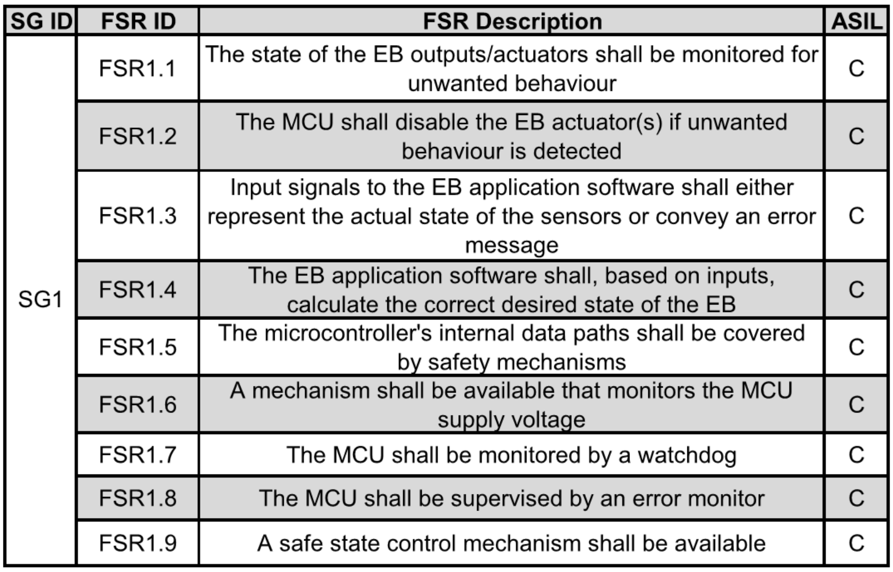
As described in Section 3, the purpose of the engine brake is to relieve the regular wheel brakes. Consequently, only under very specific circumstances could a loss of engine brake torque result in a hazardous event since the wheel brakes normally can be used to stop the vehicle regardless. A scenario with these circumstances could be if the engine brake is absent when the vehicle is braking for an extended amount of time, e.g. when driving down a long downhill slope. In this situation, the wheel brakes could eventually become overheated and start to lose braking torque, possibly resulting in the vehicle accelerating even though the driver is pressing the brake pedal. Even if the operational situation of the vehicle being driven down such a slope may not be overly uncommon, it is highly likely that the driver would notice the loss of the engine brake as he/she would have to use the service brakes to a higher extent. Therefore, this hazardous event is considered to be controllable in general, and thus obtains the controllability class C0. According to the ISO 26262 requirements described in Section 2.3, a hazardous event with a controllability class C0 does not result in any safety requirements by the standard and can thus be omitted from the analysis.

4.3 Functional safety concept

As stated in Section 2.4, the functional safety concept is intended to present the the top-level design requirements needed to comply with the ASIL of each safety goal. The concept should provide an unambiguous input for developing a technical safety concept that in turn defines a base for how to implement the solutions proposed in the functional safety concept.

Table 4.2 presents functional safety requirements derived through a qualitative analysis based on the results of the hazard analysis, the item definition and an examination of the current Volvo system. In a proper development project, these requirements may first be formulated in a preliminary manner and then adapted throughout several iterations of the development process, including subsequent steps. Due to the structure of a thesis project, the functional requirements presented here are the final requirements developed throughout the project.

Table 4.2: Functional safety requirements (FSR).



SG ID	FSR ID	FSR Description	ASIL
SG1	FSR1.1	The state of the EB outputs/actuators shall be monitored for unwanted behaviour	C
	FSR1.2	The MCU shall disable the EB actuator(s) if unwanted behaviour is detected	C
	FSR1.3	Input signals to the EB application software shall either represent the actual state of the sensors or convey an error message	C
	FSR1.4	The EB application software shall, based on inputs, calculate the correct desired state of the EB	C
	FSR1.5	The microcontroller's internal data paths shall be covered by safety mechanisms	C
	FSR1.6	A mechanism shall be available that monitors the MCU supply voltage	C
	FSR1.7	The MCU shall be monitored by a watchdog	C
	FSR1.8	The MCU shall be supervised by an error monitor	C
	FSR1.9	A safe state control mechanism shall be available	C

Functional safety requirements FSR1.1 and FSR1.2 have the purpose of ensuring that the actual state of the engine brake actuators correspond to the desired state based on calculations of the application software. Firstly, faulty behavior of the actuator outputs should be detected. Secondly, when such a behavior is detected, the MCU should control the failure by disabling the actuators, thus preventing violation of safety goal SG1. Solutions for how to implement such functionality is discussed in Section 4.4.4.

The purpose of FSR1.3 is to provide a reliable interface for the application software handling input signals. Therefore, the input signal values should be reliable in compliance with ASIL C, with error messages being sent in the occurrence of a fault. With this functional safety requirement as an assumption, FSR1.4 states that the engine brake application software should calculate a correct desired state of the engine brake actuators, i.e. the software running on the MCU should comply with ASIL C.

FSR1.5 through FSR1.9 describe the mechanisms required for the MCU to comply with ASIL C. Due to the scope of this thesis project and because the MCU is designed by a supplier external to Volvo GTT, FSR1.5 is formulated as a general requirement on the internal parts of the MCU. For the MCU, a full evaluation is provided by the supplier under a non-disclosure agreement and is therefore not included in this thesis. FSR1.6 through FSR1.9 describe external support mechanisms for the MCU which are included as assumptions in the MCU safety manual. This hardware environment for the microcontroller is described in detail in Section 4.4.2.

4.4 Technical safety concept

According to Section 2.5, the technical safety concept should refine the functional safety concept into preliminary architectural assumptions. The technical safety concept is intended to detail the item-level functional safety concepts into system-level technical safety requirements. The following sections present an analysis of what is required for the current engine brake system to comply with ISO 26262. Each part of the system is examined and solutions are discussed in relation to the requirements of the standard. Finally, a complete concept is presented based on these solutions. This concept is evaluated with respect to the failure rate metrics explained in Section 2.6.

4.4.1 Subsystem interface

The software running on the EMS is partitioned into two parts: the application and platform software. The engine brake application is designed using SIMULINK models, which are translated to executable C code through dSPACE Target Link. The translated code is referred to as application software. Low-level drivers for sensors, actuators and communication interfaces are considered parts of the platform software, which is designed directly in the C programming language.

The application and platform software communicate though signal objects. A signal object is an abstraction of a set of protected C variables together with set-and-get functions. In general, these variables hold the value of the signal, such as a sensor value, together with the quality of the signal. The quality is used to determine if the value is reliable or not.

As the purpose of the thesis work was to present hardware and low-level software solutions supporting ISO 26262 for the engine brake subsystem, a complete functional analysis of the application software tied to the engine brake was considered to be out of scope. The technical concepts presented in this thesis are stated with the intention of delivering accurate signal objects to the interface between the application and platform software considering the application software a SEooC. A brief analysis of the application software was performed to map the connected hardware elements and signal objects of the engine brake mechanisms. The analysis started from the SIMULINK models associated with the engine brake. Each signal in the models was followed from the generated application code down to the activation of the actuator or reading of the sensor. Figure 4.3 presents the hardware interface of the engine brake application model.

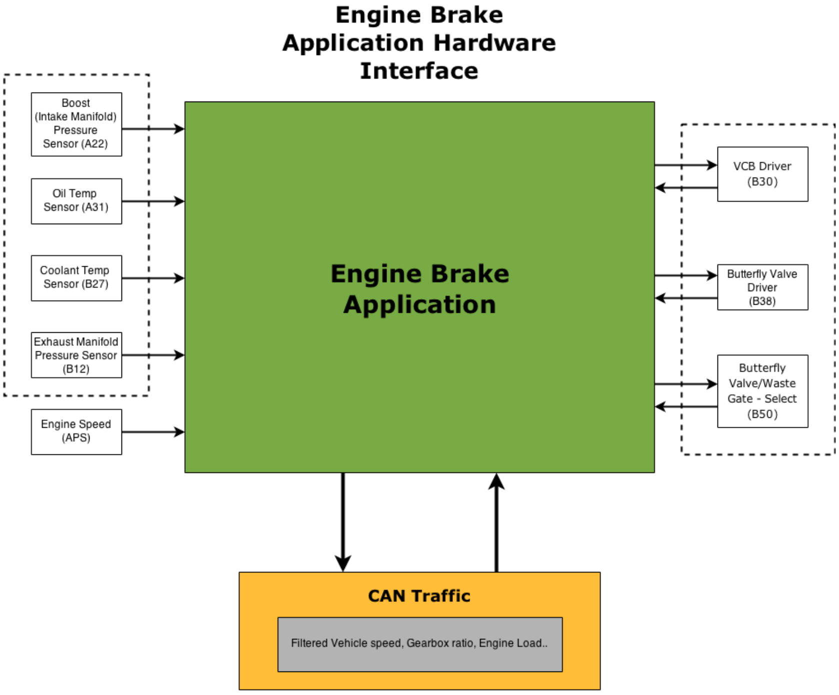

Engine Brake Application Hardware Interface
Boost (Intake Manifold) Pressure Sensor (A22)
Oil Temp Sensor (A31)
Coolant Temp Sensor (B27)
Exhaust Manifold Pressure Sensor (B12)
Engine Speed (APS)
Engine Brake Application
VCB Driver (B30)
Butterfly Valve Driver (B38)
Butterfly Valve/Waste Gate - Select (B50)
CAN Traffic
Filtered Vehicle speed, Gearbox ratio, Engine Load..

Figure 4.3: Engine Brake Application Hardware Interface.

The scope of the ISO 26262 concept is to provide solutions for the different types of components utilized by the engine brake. Figure 4.3 shows that the engine brake application utilizes three actuators and five sensors. The actuator for the butterfly valve is controlled with a PWM signal while the VCB driver and the butterfly/waste gate select signals are either on or off. The duty cycle of the PWM signal controls the butterfly valve and determines the amount of generated backpressure. The PWM signal is controlled with a closed-loop control algorithm in which the backpressure sensor is used as feedback. Thus, a faulty high value would lead to excess braking torque, which could lead to a violation of safety goal SG1.

The backpressure sensor is considered the only critical sensor of the engine brake system due to an assumption on the engine brake application software that it is the only sensor for which a faulty reading could lead to a violation of the safety goal. The sensor values are read by ADC channels in the microcontroller.

In conclusion, technical safety concepts are required for the on/off pins, the PWM pin and the inputs circuitry that reads data from the sensors.

4.4.2 Microcontroller

The microcontroller unit (MCU) is the core of the ECU; it makes the bulk of all control decisions. The overall safety of the product depends on the MCU functioning correctly, which is why a large part of the more implementation-specific recommendations of ISO 26262 concern safety mechanisms internal to the MCU. Many of these recommendations are best solved by hardware mechanisms and are therefore implemented entirely by the microcontroller supplier. The suppliers develop the MCUs as SEooCs specified to meet the ISO 26262 requirements for a range of end applications to enable more generalized products in favor of application-specific, tailored controllers. Naturally, the final item ASIL is determined by a full system analysis where the assumptions and data of the MCU are integrated. With this method, integration engineers, such as the Volvo ECU designers, can rely on the safety level of the MCU and focus on ensuring that the controller is integrated correctly.

The Aurix microcontroller family is used here as an example of an MCU in compliance with ISO 26262. The Aurix family [8] is designed to suit automotive systems up to ASIL D. Its safety manual [20] provides assumptions of use (AoU) that state requirements to be met before the safety level of the MCU is valid. Thus, the integrator's task is to make a specification of which of the application-specific parts of the MCU that are used for safety related functions. A tool provided by Infineon then recommends which software or external hardware mechanisms that should be implemented. The tool also calculates the single-point and latent fault metrics for the MCU mentioned in Section 2.6; the metrics for the system external to the MCU have to be calculated separately. This can be done by adding the remaining failure rates to the MCU failure rates given by the mentioned tool. However, because the total safe-fault rate of the MCU can be large compared to the respective rate of external components, the addition of residual and latent fault rates to those of the MCU can be insignificant to the final metric. Therefore, separate metrics can be calculated for the other safety related ECU components. This method may result in stricter demands on these components, but may also result in a more evenly distributed diagnostic coverage over hardware elements. ISO 26262 Part 5 clause 8.4.7 [13] suggests this strategy to prevent the side effect of some elements becoming insignificant when calculating the architectural metrics due to large differences in failure rates.

As described in Section 2.6, the item also has to comply with the PMHF metric concerning random hardware failure rates for each safety goal. In this respect, Infineon claims that the Aurix MCUs in general use approximately one third of the ASIL D target for PMHF ($10^{-8}h^{-1}$). It is then up to the integrator to distribute the remaining error budget over other safety related elements.

From the perspective of the Aurix safety manual, four main hardware mechanisms are assumed as a hardware environment for the ASIL D of the MCU to be valid. These are an external voltage monitor, an error monitor, a watchdog and a safe state control. It is up to the integrator to design or choose components that implement these measures with guidance by ISO 26262 and the MCU data sheet. A sketch of this monitoring concept is included in Figure 4.4 where the external measures are integrated into one component called a system basis chip (SBC).

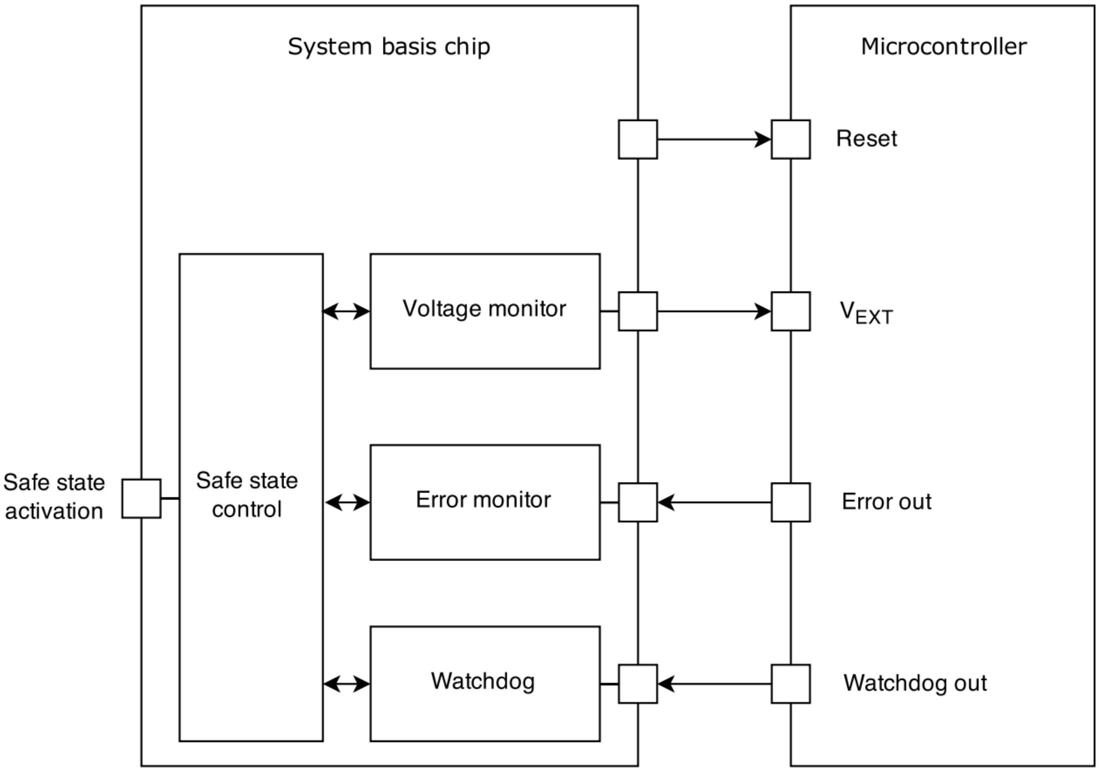


Figure 4.4: Microcontroller hardware environment concept.

The voltage monitor function is discussed with the power supply in Section 4.4.5. The error monitor is defined by the Aurix documentation as an external function that monitors an error pin on the MCU. Which faults are signaled through this protocol can be configured through software, but are typically faults that the MCU cannot handle internally. What action the error monitor should take upon detecting an error is determined by the integrator and may include resetting the MCU or powering down the ECU.

As for the required external watchdog mechanism, the ISO 26262 Part 5 [13] tables of recommendations regarding the diagnostic coverage of various safety mechanism lists several ways of program sequence monitoring. In a system where only medium diagnostic coverage is required, a window-based watchdog may be sufficient. However, for reaching a high diagnostic coverage, the watchdog mechanism should be combined with logical monitoring of the program sequence. This type of monitoring generally demands a more complex software program flow control, and can work with different types of external watchdogs. The Infineon Aurix family only requires a window watchdog to meet ASIL D, but is also compatible with functional watchdogs that operate through a question-answer protocol. Flow monitoring is implemented through internal watchdogs in the Aurix microcontrollers and an external functional watchdog is therefore not required. The Infineon TLF35584 SBC includes independent window and functional watchdogs, of which by their own documentation only the former is required for an ASIL D classification of the MCU.

Lastly, the safe state control is the mechanism that handles faults detected by the voltage monitor, the watchdog and the error monitor. This component typically includes logic which determines the actions to take or the signals to trigger in the event of an error. These actions should ensure that the system enters a safe state, which may include notifying or resetting the MCU or switching the power off for some circuits or for the whole ECU. The TLF35584 includes such a safe state control, which trigger safe state signals to output pins of the SBC. There are no specific instructions in ISO 26262 concerning this mechanism. However, the ability to transition the system to a safe state is essential for most safety-related applications. Thus, it is recommended to follow the MCU vendor's recommendations regarding how the MCU should be controlled in respect to safe states, and integrate that with any potential application independent needs, safety related or not.

The voltage monitor, error monitor, watchdog and safe state control discussed in this section may be integrated as one component as in the TLF35584 SBC, but may also be implemented through several components. Choosing the latter strategy may be beneficial if it results in a decrease in cost or if a more feature-rich version of any of the mechanisms are required, such as a more advanced watchdog.

4.4.3 Subsystem inputs

Based on functional safety requirement FSR1.3, mechanisms should be implemented to supply the application software with the correct input values and quality status. This section discusses solutions for ensuring dependable input values in compliance with ISO 26262.

For reaching medium (90 %) or high (99 %) diagnostic coverage for analogue inputs, Annex D of Part 5 [13] of the standard suggests that the complete signal line is monitored for stuck-at faults, stuck-open, open or high impedance outputs, short circuits between signal lines, drift and oscillation. This monitoring concept is to be seen as a guideline. Thus, if it can be properly justified, a tailored safety mechanism can be used that only covers the fault modes that lead to violation of the safety goal.

ISO 26262 proposes two general methods for achieving high diagnostic coverage for analogue and digital inputs: test patterns and hardware redundancy. The former relies on cyclical tests of I/O unit in which observations are compared to expected values. To discover faults such as sensor value offsets online, more than simple software comparisons are required. A feasible solution could be to develop a sensor that either includes internal tests and redundancy, or that provides a means to perform more exhaustive pattern testing in cooperation with the MCU, possibly by the use of fault injection.

The second approach, which uses hardware redundancy, appears to be the more widely discussed option in other evaluations such as [21]. The most obvious disadvantage of this approach is that it requires more hardware, which increases the cost. One way of mitigating this effect is to use sensors with internal redundancy, removing the need of having two or more physically separate sensors. The feasibility of using internal sensor redundancy is dependent on the application; for example, fully redundant Hall sensors have been used for automotive applications since before ISO 26262 was introduced [22].

Most currently available articles on hardware implementation with regard to ISO 26262 are written considering light-duty vehicles. Since these vehicles do not use active engine braking such as with the VCB, not much has been published that discusses the functional safety of engine brakes. As a consequence, the exhaust manifold pressure sensor has not been considered as safety critical and there are therefore no easily found off-the-shelf solutions for ensuring its safe operation. Thus, if this sensor is assessed as safety critical in a complete safety evaluation of the engine brake application, a suitable sensor has to be developed together with the supplier.

Regardless of the safety mechanism used for the inputs to the MCU, it is critical to ensure a reliable reference voltage to the analog-to-digital converter (ADC). The ADC reference voltage is discussed together with the MCU power supply in Section 4.4.5.

4.4.4 Subsystem outputs

Functional safety requirements FSR1.1 and FSR1.2 dictate that the outputs/actuators of the engine brake should be monitored for unwanted behavior and that they should be disabled in the event of a failure. Thus, safety mechanisms must be implemented that ensure compliance with these requirements to prevent violation of safety goal SG1. According to clause 8 of Part 5 of the standard [13], the single-point fault metric target for ASIL C is 97 %. This target and the 80 % latent fault metric target are used as a goal in the evaluation of possible solutions presented in this section.

In the current Volvo GTT design, both the VCB and the butterfly valve are implemented by connecting their respective actuators between supply voltage and low-side MOSFETs as can be seen in Figure 4.5. A pre-driver chip with built in diagnostics (on pin V_Diag) drives the MOSFET switches (on pin MOSFET DRIVER OUTPUT), which in turn control the state of the engine brake actuators.

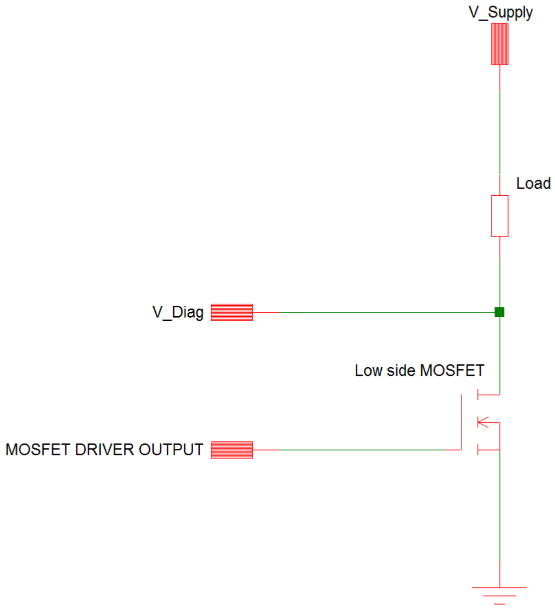


Figure 4.5: The current hardware setup for the actuators of the engine brake.

A short circuit to ground on the low-side of one of the actuators controlling the VEB would lead to a single-point fault violating safety goal SG1 as this would cause activation of the actuator. The current safety mechanism covering this fault composes measuring of the drain voltage of the low-side transistor as seen in 4.5. The short-circuit-to-ground fault is detected and signaled to the MCU if the measured drain voltage is below a certain percentage of a fixed reference voltage of 5 V. However, even though the fault is detected it is not possible to deactivate the load in the current configuration as the short circuit to ground would have overridden the low-side MOSFET. Consequently, from the perspective of ISO 26262, there is no diagnostic coverage of this single-point fault, and measures should therefore be introduced in future designs to provide safety mechanisms to cover the fault.

To be able to deactivate the load when a short circuit to ground is present on the actuator low side, the low-side MOSFET could be complemented with a high-side MOSFET that can switch off the actuator battery voltage as exemplified in Figure 4.6. This extra MOSFET would increase the diagnostic coverage in combination with a diagnosis that is able to detect a sufficient proportion of critical faults. Since this high-side switch would act as a safety mechanism for low-side failures, faults in it would classify as latent faults if not discovered. Therefore, it is recommended to implement a once-per-drive-cycle test of the high-side switch to ensure a sufficient latent fault metric.

Tests have showed that the diagnostics in the current EMS implementation can be improved in regard to detection of short circuits to ground past the low-side MOSFETs.

Short circuits appear with varying resistances to ground, and the tests have showed that if this resistance is above a few Ohm, the fault may pass undetected due to the low reference voltage used by the diagnostics for comparison with the MOSFET's drain voltage. The voltage differential over the low-side MOSFET to ground must be below 80% of the 5 V reference in order for the for the fault to be detected with the pre-driver IC currently in use for diagnosis. With a battery voltage of approximately 28 V, the voltage over the actuator could be as high as $28 - 5 \cdot 0.8 = 24V$ before the fault is detected. As it is highly likely that the actuator would be activated by a 24 V voltage, the diagnosis has to be improved. One way of achieving a better coverage would be to use a pre-driver with a diagnosis that uses the battery voltage as reference rather than 5 V.

Since the diagnosis relies on SPI communication with the MCU, it is important that the components related to the communication are fully functional and meet the timing constraints necessary for avoiding violation of a safety goal. Also, as the MCU is responsible for deactivating the high side when faults are present, it is critical that the software tasks handling the diagnosis are reliable and that they execute within a given fault-tolerant time interval.

Four technical safety concepts are proposed for the peripherals of the VEB actuators to increase coverage of faults that may violate safety goal SG1.

4.4.4.1 Concept 1

In the first concept, which is presented in Figure 4.6, an extra MOSFET has been added as a safety mechanism. This transistor acts as a high-side switch between the power supply and the actuator. As mentioned, the high-side MOSFET is used to deactivate the load whenever the low side is short circuited to ground.

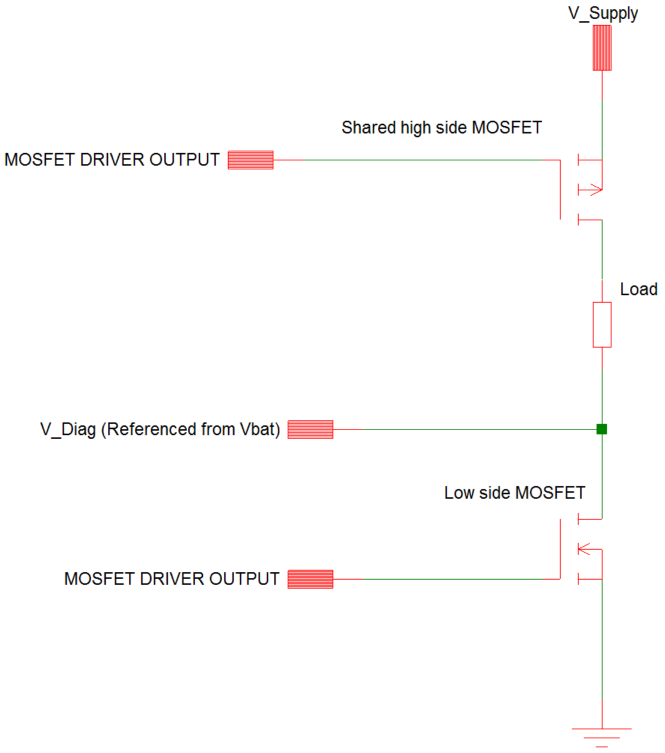


Figure 4.6: A schematic of the first and second technical safety concepts of the VEB actuators.

The fault diagnostic is performed, as previously described, by built in diagnostics of the MOSFET pre-driver IC. As concluded, the diagnosis should be performed with the battery voltage as reference. Faults detected by the pre-driver IC are reported to the microcontroller over SPI. Software then decides if the high-side MOSFET of the faulty output circuitry should be deactivated. Depending on the fault-tolerant time interval for safety goal SG1, the MCU tasks handling the diagnosis and fault control have to be designed to react within a sufficiently short time.

To reduce cost and to save circuit board space, the high-side MOSFET can be shared among several actuators with individual low-side switches. With shared high-side switches, it is important that the actuators for each common switch are chosen in such way that the deactivation of the high side does not introduce additional safety hazards by for example disabling other safety-critical devices.

4.4.4.2 Concept 2

The second technical safety concept is based on controlling the state of the high side switch from the first concept with a fault signal directly generated by the MOSFET pre-driver diagnosis. In this way, the load is disabled without any interaction from the microcontroller. As the pre-driver diagnostics is likely to report other faults on its error pin than the ones critical with respect to SG1, it may happen that the actuator is disabled unnecessarily. Beyond this disadvantage, the microcontroller still has to be alerted when a fault has occurred, and typically also be able to override the pre-driver error pin control of the high-side switch.

4.4.4.3 Concept 3

The third technical safety concept for the VEB actuators is presented in Figure 4.7.

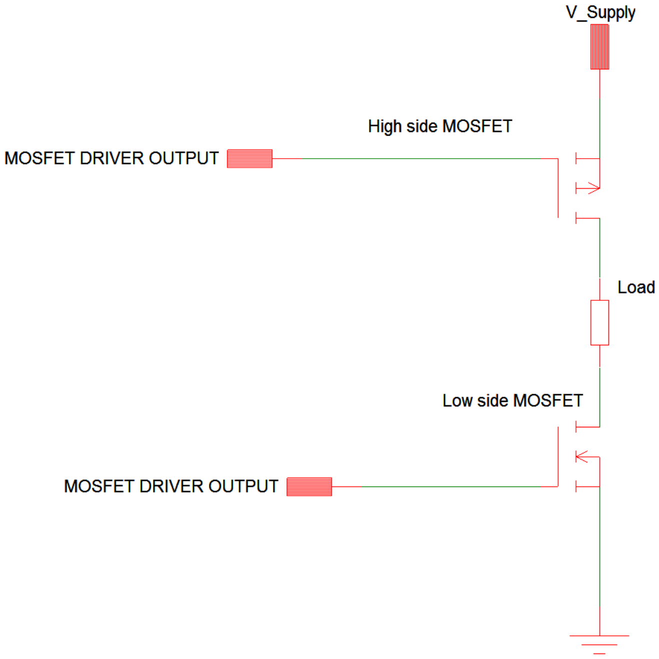


Figure 4.7: The third technical safety concept of the VEB actuators.

Here, each actuator is allotted a separate high-side MOSFET. Furthermore, the high side is controlled synchronously with the normal low-side switch control, leaving the load disconnected from both battery and ground when not activated. Thus, the discussed potential of a single-point failure is eliminated since a short circuit to ground on the low side no longer leads to an activation of the load. While this safety mechanism does not require time-critical interaction with the MCU, extra measures have to been taken to diagnose the MOSFETs. A fault in either switch would result in a multiple-point fault if combined with a second fault. Such a fault would be considered latent if undetected and count towards the latent-point fault metric target. As described in Section 2.6.1, it is generally sufficient to diagnose potential latent faults once per drive cycle, such as at each MCU reset. The diagnosis can be performed by introducing a software scheme in which each MOSFET drain is measured during a short time when one switch is open and the other one closed, and vice versa. Another alternative is to connect a weak pull up resistor to the high side and a weak pull down resistor to the low side and then measure for short circuits.

4.4.4.4 Concept 4

The fourth technical safety concept differs from the other concepts by introducing a custom circuit that deactivates the high-side switch independently of the MCU or pre-driver diagnosis. This circuit is presented in Figure 4.8.

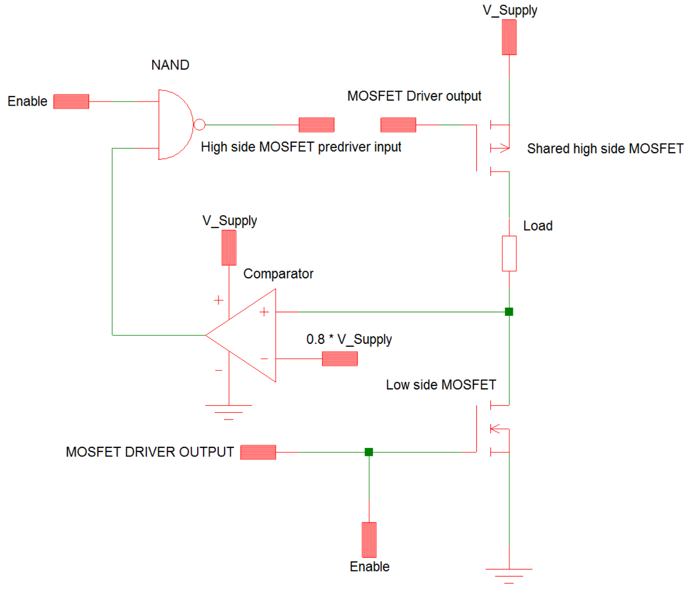


Figure 4.8: The fourth technical safety concept of the VEB actuators.

The added circuitry composes a comparator and a NAND gate. The comparator is used to detect when the low-side MOSFET is short circuited. A short circuit to ground on the low side of the load causes the voltage at the drain terminal of the MOSFET to drop. When the drain voltage is below $0.8 \cdot V_{Supply}$, the comparator sets the input of the NAND gate high. The NAND gate is used in order for the MCU to be able to disable the safety mechanism when the load is intended to be activated. This is accomplished by connecting the signal that controls the VCB to the other NAND input. Because the output of the NAND gate is connected to the input of the high-side MOSFET pre-driver, the high-side MOSFET is automatically disabled before a violation of the safety goal is possible.

When the load is deactivated normally by the low-side transistor, voltage fluctuations may be present at the drain of the MOSFET due to the characteristics of the actuator solenoid. These fluctuations could cause the safety mechanism to improperly trigger a deactivation the high-side switch. This behavior may be problematic if the high-side switch is shared among several actuators and can be avoided by delaying the safety mechanism enable signal while deactivating the low-side switch.

In its current implementation, the VCB valve is controlled by an on/off signal produced on a PWM output pin on the MCU. Since the duty cycle of the PWM signal is either 0 or 100%, any of the four concepts would be possible to implement as safety mechanisms for the VCB.

As described in Section 3, the AVU is controlled by one digital pin and one PWM signal. The former can be controlled in the same way as the VCB actuator, while the PWM signal requires special consideration. Implementing the fourth concept as safety mechanism for the AVU PWM signal would require additional components such as a large capacitor at the input of the comparator. This capacitor needs to be charged to convert the PWM signal to a comparable DC level. As this would take up a significant amount of extra space on the PCB, either concept one, two or three is preferable for PWM signals. However, since the AVU features built-in diagnosis of stuck-at faults and since duty-cycle diagnostics would be of more interest for the PWM signal, digital output monitoring as will be described in Section 5.1 should result in a higher diagnostic coverage for the PWM pin compared to the concepts presented in this section.

4.4.5 Power supply and reference voltages

ISO 26262 Part 5 Appendix D [13] defines general recommendations for which safety mechanisms that should be implemented for reaching a high diagnostic coverage. A faulty power supply may cause the ECU to behave unpredictably, possibly resulting in violation of a safety goal. Thus, for such a central system element, it may be sensible to aim for a high (99 %) diagnostic coverage with respect to residual faults. The standard suggests that this may be achieved through analysis of drift, oscillation, power spikes, under voltage and over voltage on the output of the power supply.

An example of a component that, according to specifications, may provide the required safety mechanisms is the previously mentioned Infineon TLF35584 system basis chip. Since the output voltage of the power supply is considered a safety-critical function, the monitoring of the same can be considered a safety mechanism. The TLF35584 includes two independent comparators for detecting under and overvoltages respectively. The comparators use a common bandgap as reference for error detection. A failure in this safety mechanism would be a second-order–multiple-point fault in combination with a failure of the power supply output voltage. Thus, to prevent latent faults in the voltage monitoring circuit, the currents through both the main regulator bandgap and the monitoring bandgap are monitored. Failure of this next-level monitoring could contribute to a third-order–multiple-point fault, which normally does not have to be covered by a safety mechanism.

There is no separate monitoring of drift or oscillation in the TLF35584 other than built in preventive measures such as capacitive couplings in feedback loops to prevent oscillations and bandgap references to prevent drifting. A safety manual should be provided with the chosen power supply/SBC that provides diagnostic coverage values for such measures to enable a full system analysis. The final fault type that the standard recommends monitoring is, as mentioned, power spikes. Most power supplies handle this through overcurrent protection, which is also present in the TLF35584.

There are no specific recommendations in the standard regarding how ADC reference voltages should be supplied. For the ASIL D-class microcontrollers in the Infineon's Aurix family, there are no requirements in the safety manual [20] regarding ADC references other than that it should be monitored. Thus, it may be suitable to apply the same safety mechanisms for reference voltages that ISO 26262 recommends for power supplies in general. The TLF35584 provides an independent post regulator for generating a voltage reference. This regulator has a more strict specification of output variations compared to the ECU supply regulator. The reference output is followed by two independent voltage trackers which are monitored for deviations from the reference. These trackers are used for sensor supply to ensure that the sensors and the ADC uses the same reference, improving accuracy significantly in an efficient way compared to relying on the precision of absolute voltages. For safety-critical inputs, the two trackers can be used for supplying redundant sensors, whether they are physically separate or not, with independent supply voltages. The fault detection algorithm for the ADC data may then detect differences in sensor output.

4.4.6 Concept statement

This section aims to integrate some of the concepts presented in Sections 4.4.1 through 4.4.5 into a complete hardware concept for the Volvo Engine Brake system to comply with ASIL C according to ISO 26262. Since this thesis work was not a product development project, it was not possible to perform every task as defined by the ISO 26262 work flow. As thus, in contrast to performing hardware fault metrics calculation on a detailed hardware design, this section presents such calculations based on the technical concept.

As described in Section 4.4.2, the Aurix TC277TF microcontroller is used for this project's prototype. Therefore, this MCU is also used for the final technical concept evaluation in this section. As discussed, the fault metrics calculation is performed separately for the MCU through an FMEDA tool provided by Infineon, for which the goal is to meet ASIL C with the SPFM and LFM calculations. Since the targeted MCU is designed for ASIL D, the FMEDA tool claims that the Aurix TC27x family microcontrollers meet the 99 % target for the SPFM and the 90 % target for the LFM, provided that the assumed hardware environment of the MCU is present. Here, the previously mentioned and by Infineon recommended system basis chip Infineon TLF35584 is assumed to provide this hardware environment, making the FMEDA valid. Furthermore, it is assumed that all the software tests and software safety mechanisms required by the FMEDA tool are implemented. Due to the design of the FMEDA tool, all internal faults required for the MCU to operate without extra peripherals are included in the analysis, which results in an ASIL D classification.

In the case of the VEB case study, several inputs and outputs are required by the application; integrating these into the fault rate analysis requires activation of certain peripheral modules in the FMEDA tool such as the ADC, the I/O ports and the SPI.

Unless safety mechanisms are assigned to these modules, they will add single-point and latent faults to the FMEDA calculations. Even so, since the SPFM and LFM are calculated through division with the total failure rate of the MCU, the failure rates added by these peripheral modules are insignificant in comparison to the combined failure rates of the entire ECU. Therefore, the FMEDA tool claims that the ASIL D targets are met even with the added modules without any safety mechanisms. Consequently, the challenge in meeting the ASIL C targets for SPFM and LFM is in the design of the safety-critical circuits external to the MCU.

In contrast to the calculation of the SPFM and the LFM where the ECU, excluding the MCU, can be considered separately, the calculation of the PMHF metric must include all parts of the system critical to the safety goal to obtain a total failure rate. Infineon claims that the Aurix TC27x family is designed to use around one third of the ASIL D target for PMHF ($10^{-8}h^{-1}$). However, the provided FMEDA tool does not include PMHF calculations; the effects on the PMHF metric of adding the mentioned peripheral modules is therefore unknown. As explained in Section 2.6.3, the PMHF metric can be expected to increase linearly with the residual or dual-point failure rates. Thus, even as the SPFM and the LFM are nearly unchanged by the introduction of the peripheral modules, the total residual failure rate of the MCU increases noticeably, and in doing so also threatens to invalidate Infineon's claim regarding how much of the PMHF budget is used by the MCU. In conclusion, it may be wise to implement the safety mechanisms suggested by the FMEDA tool for the peripheral modules to greatly decrease their contribution to the PMHF metric. These mechanisms include use of redundant ADC channels and digital ports, and loop-back monitoring of output signals.

As made clear in Section 4.4.3, the safety measures required for the sensor inputs of the VEB application depends significantly on an evaluation of the application with regard to how critical the inputs are. Here, it is assumed that the backpressure sensor is critical with respect to safety goal SG1 and should be covered by an appropriate safety mechanism. Since this mechanism depends significantly on sensor design, a generalized 2-way redundant channel solution is chosen for this concept. Estimated diagnostic coverage and failure rates are presented in the end of this section together with the output peripherals. Since algorithms for error detection with 2-way redundant input channels are outside the scope of this project, the ISO 26262 suggestion of a possible high (99 %) diagnostic coverage with this method is used for calculations.

Section 4.4.4 presents four safety mechanism concepts for the output peripherals of the VEB. The first of these concepts was chosen for the calculations in this section. For this concept, ON Semiconductor's NCV7519 low-side MOSFET pre-driver IC [23] is used for controlling the low-side MOSFET and for performing low-side diagnostics. The concept relies on errors messages being transmitted to the MCU over an SPI, and on the MCU for closing a high-side switch in the event of a critical error. Figure 4.9 shows a schematic of the output peripheral circuit used in this concept. The circuit includes three output channels, which are all controlled and monitored by the NCV7519 pre-driver IC. The top channel controls the VCB valve, while the remaining two channels control the air valve unit which controls the butterfly valve for exhaust braking. All channels include supporting circuitry such as gate or drain clamps to tolerate voltage spikes, and resistors to protect the pre-driver IC. The inputs to the NCV chip are connected to the microcontroller.

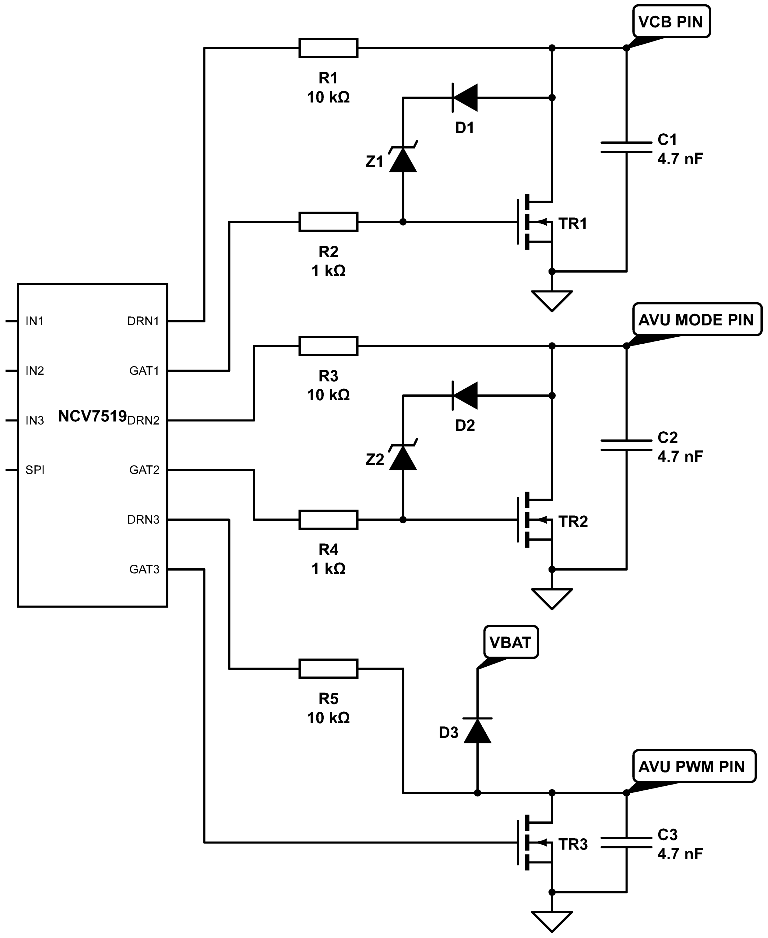


Figure 4.9: Schematic of the NCV7519-based output peripheral circuitry.

The main drawback of the solution presented in Figure 4.9 is that it depends solely on the NCV chip as a safety mechanism. ON Semiconductor claims that their NCV7519 has a failure rate of approximately 38 FIT under the environmental conditions provided by us based on Volvo data. However, no information was made available regarding failure mode distribution. Thus, it is possible that the NCV may fail in such a way that its outputs change to an erroneous state, potentially leading to an activation of for example TR1, which could violate safety goal SG1. Then again, the pre-driver diagnostics would recognize this unexpected activation as a short-to-ground fault if the NCV has read the correct inputs from the MCU. In this case, an error would be reported to the MCU, which could open the high-side switch. Conversely, if the cause of the erroneous output state is that the input has been misread, the NCV diagnostics would not detect an error.

If additional information regarding failure mode distributions cannot solve the problem, then additional safety mechanisms must be added for the NCV chip. For example, the transistor gate voltages could be read back to the MCU to detect faulty states in the NCV outputs. This safety mechanism would also cover errors in the MCU port output circuitry. However, to use one extra MCU pin for every output would quickly exhaust the available pins. A solution for the pin shortage could be to use a multiplexer or an SPI register component to read the gate voltages. The FMEDA calculations presented in the end of this section assume that some implementation of gate voltage read back is used.

Another possible fault in the NCV pre-driver is that the diagnosis may stop working and, for example, send "All OK" messages to the MCU. A fault such as this may be classified as a multiple-point fault as it requires an additional fault to occur before the safety goal may be violated. As previously described, to prevent such faults from being latent, it is enough to perform tests once every driving cycle. Because there are no built in tests in the NCV7519, faults must be induced by external components. This type of fault induction is used in some other Volvo ECUs and can be implemented by connecting the transistors' drains to a common transistor which can induce a short-to-ground fault. As with the gate voltage read back, it is assumed for the following calculations that a once-per-cycle test of the NCV diagnostics is implemented.

Table 4.3 shows the results of an FMEDA analysis performed on the concept for engine brake peripherals presented in this section. The full FMEDA is available in Appendix A. Internal failures of the MCU are not included in these calculations. However, as described, the separate FMEDA for the MCU supports the ASIL D compliance of the MCU and is therefore not limiting in comparison to the peripherals. Also, because the MCU is reported to have a PMHF metric in the order of 3 Failure In Time (FIT), the total PMHF would be this value added to the one in Table 4.3 covering the peripheral circuits. Clearly, the total PMHF is well below the 100 FIT target of ASIL C.

Table 4.3: Results of the FMEDA, including target values for ASIL C.

Metric	**Value**	**Target for ASIL C**
Single-point fault metric	98,6 %	97 %
Latent fault metric	98,0 %	80 %
PMHF	1,9 FIT	100 FIT

Approximate component failure rates for the FMEDA were acquired from the Siemens SN29500 industry standard [24]. Since the sensors used for the two redundant input channels are not specified, a relatively high failure rate of 50 FIT was assumed.

The most challenging part of performing the FMEDA was to determine the diagnostic coverage of the included safety mechanisms. ISO 26262 suggests values for diagnostic coverage based on generalized spectra of critical faults, while also stating that diagnostic coverage should be calculated with respect to violation of the specific safety goal in question. Thus, as it is only unintended activation that is the critical error in the engine brake system, it should be sufficient for the NCV diagnostics to be able to diagnose such an error. Since ISO 26262 Part 5 [13] recommends a DC fault model for diagnosis for both medium (90 %) and high (99 %) diagnostic coverage with digital I/O, and as several of this model's failure modes are not critical to safety goal SG1, we have assumed a diagnostic coverage of 95 % for the NCV diagnostics.

There are two ways to handle failures of wires for compliance with ISO 26262: to classify the wires as electrical components or as mechanical parts. In the former case, Part 8 clause 13 of the standard [16] states that safety related basic hardware parts can be addressed sufficiently through standard quality control. In the latter case, Part 3 clause 8 [15] states that the implementation of safety requirements for elements of other technologies shall be ensured through specific measures that are outside the scope of ISO 26262. In conclusion, both methods require no special process outside of normal quality work. Thus, the failure rates of wires are not included in the FMEDA. Also, based on Volvo experience, it is unlikely that a wire failure would lead to the activation of an actuator.

The most critical part of the FMEDA in this specific case is the single-point fault metric. Since this metric has a much stricter target for ASIL C than the latent fault metric, it is the limiting factor in this analysis. The PMHF metric scales linearly with the residual failure rate and is therefore low for this relatively small peripheral system where the total safety related failure rate is 134 FIT, based on the FMEDA. Thus, only the 34 FIT exceeding the 100 FIT PMHF target of ASIL C has to be covered by safety mechanisms. The latent fault metric includes the assumed start-up test of the NCV diagnostics, with an assigned failure mode percentage of 50 % to this failure, which is likely to be a significantly higher ratio than required. Even so, with no diagnostic coverage with respect to latent faults in the NCV diagnostics, the latent fault metric is still above the 80 % target. Thus, the mentioned extra test circuitry may not be required.

Chapter 5

Prototype design and verification

While Section 4.4.6 describes a concept aimed to meet the ASIL C requirements on the Volvo Engine Brake, this chapter's purpose is to present how some of the solutions from the concept were implemented and verified in a prototype. The goal when designing the prototype was to experience our solutions from a more practical perspective, with the aim of discovering features and drawbacks of the presented concepts and of the Infineon Aurix microcontroller architecture.

The prototype was centered around an Infineon Aurix development board containing the TC277TF microcontroller. The reason for this choice of hardware was that using the Infineon products would result in valuable input for Volvo's upcoming decision about which MCU platform to use in future ECUs. Since the company has extensive experience in working with Freescale MCUs, it was seen as more rewarding to study the Infineon platform.

The subsequent subsections describe each considered safety mechanism, as well as how the prototype software architecture was implemented. Furthermore, these sections do not cover the complete technical concept of Section 4.4.6, but rather present solutions in a more general manner. The reason for this is that since the prototype was built as a separate system without existing Volvo hardware and software, such as for example the VEB application software, it was more rational to test solutions without considering precisely how they would be integrated with the Volvo Engine Brake. The thesis work's goal of using the VEB to find generalized solutions that can be adapted to other subsystems further supports this rationale.

5.1 Microcontroller

The Aurix TC277TF microcontroller around which the prototype was built is a SoC designed for safety-critical automotive applications. The MCU includes internal safety mechanisms such as lockstep cores to enable compliance with high ASILs according to ISO 26262. The subsequent sections describe the safety related features of those peripheral modules of the MCU that are relevant to the engine brake case study of this thesis project.

Serial peripheral interface

ISO 26262 Part 5 [13] suggests that certain failure modes should be analyzed to achieve diagnostic coverage of general faults regarding communication interfaces such as the SPI. For a diagnostic coverage of 60 %, failures of the communication peer, message corruption, message delay, message loss and unintended message repetition should be analyzed. To achieve 90 %, these failure modes should be complemented with diagnosis of re-sequencing and message insertion faults. For a diagnostic coverage of 99 %, masquerading should also be performed.

The failure modes for a 60 % diagnostic coverage are typically analyzed by adding one-bit hardware redundancy to act as a parity check. To reach 90 % diagnostic coverage, the standard instead suggests that multi-bit hardware redundancy is implemented. The standard also presents four other safety measures which may be used to achieve 90 % diagnostic coverage. These are read back of sent messages, transmission redundancy, information redundancy and frame counting. To reach 99 % diagnostic coverage, the standard presents that complete hardware redundancy, inspection using test patterns or a combination of the safety mechanisms presented for 90 % diagnostic coverage may be utilized as safety measures.

The SPI peripheral module of the AURIX microcontroller, called the QSPI, used for the prototype features four separate SPI modules, each capable of communicating with up to 16 slaves. For each channel, it is possible to add a one-bit parity check as safety mechanism resulting in a diagnostic coverage of 60 %. For reaching higher diagnostic coverage, Infineon assumes that an end-to-end safe protocol is implemented in software. Such a protocol could imply that data is transferred in packets for which cyclic redundancy check (CRC) checksums are calculated. This method would yield the same result as multi-bit hardware redundancy, resulting in a diagnostic coverage of 90 %. As there are no other safety mechanisms embedded in the QSPI module, Infineon also suggests that a redundant QSPI module could be used for read-back of sent data. Comparing this to the Cobra55 microcontroller by Freescale, the Cobra55 has a greater number of safety mechanisms embedded in hardware. These include a combination of information redundancy, frame counting, timeout monitoring and a CRC checker.

The prototype features a one-bit parity check together with flags indicating transmission and timeout error for the SPI communication with the NCV7519 MOSFET predriver. According to the standard, these safety mechanisms corresponds to a diagnostic coverage of 60 %. The coverage can be increased by adding additional safety mechanisms in software, or by connecting a read-back line using a redundant QSPI module. The diagnostic coverage can be further increased to 99 % by adding software functions that periodically transmits test patterns. When transmitting these test patterns, the microcontroller expects a certain answer from the slave, and can therefore diagnose the communication. By transmitting faulty frames it is also possible to verify that other safety mechanisms work and to discover latent faults.

Analog-to-digital converter

The ADC peripheral module of the Aurix TC277TF composes several independent analog-to-digital converter groups. The converter independence enables the use of redundant input channels in highly safety-critical applications as recommended by ISO 26262. Furthermore, three separate sources for requesting conversions are implemented in the module. The queued request source enables high-priority conversions to be queued in a user-defined order that overrides the two other request sources, thus ensuring execution of critical conversions.

The ADC module also includes more explicit safety features: broken-wire detection, multiplexer diagnostics and converter diagnostics. The latter two are necessary to test the operation of the converter itself, and to test external multiplexers. From the perspective of ISO 26262, the availability of mechanisms for testing the ADC is crucial in obtaining a good latent-point fault metric. The former mentioned safety feature, the broken-wire detection, is a mechanism to precharge the converter's capacitor to either V_{REF} or V_{GND} before each sample phase to detect proper connection to an external analog sensor. The precharged value is considered as out of range, which means that an error is detected if the capacitor has not obtained a voltage within the valid range during the sample phase. No built-in comparison for the broken-wire detection is included in the ADC module – the error detection must be performed by software. This mechanism is not enough in itself for reaching a high diagnostic coverage of faults in the signal lines between sensors and the ADC. However, it may be useful in low-ASIL systems or to complement additional safety mechanisms such as pattern testing or channel redundancy. Possibly, the broken-wire detection may also be more useful if the most critical faults of the targeted ADC channel are out-of-range errors.

Digital output monitoring

PWM signals play a crucial role in any engine ECU in controlling actuators that require more linear control than simple on/off switching. The General Purpose Timer unit (GPT) and the Generic Timer Module (GTM) are the modules through which the Aurix TC277TF can generate PWM signals. Moreover, the Aurix MCU includes a module for monitoring digital signals called the Input Output Monitor (IOM), which is useful for ensuring the correct output from the MCU. The IOM includes 16 Logic Analyzer Modules (LAMs) which can be configured to analyze a variety of signals including I/O and those internal to the MCU. A LAM can be set up to compare a monitor signal to internal counters to find errors such as too long or too short duty cycle or period. Another option is to generate a reference signal in one of the other MCU modules to be compared to the monitor signal within a LAM. In conclusion, the IOM can be used as a safety mechanism to cover errors in digital signals generated by the MCU such as, for example, a PWM output. Depending on from where in the output signal line the monitor signal is fed back from, the safety mechanism could potentially also cover some faults in other ECU components such as MOSFET drivers.

Fault events in the IOM generate alarm events to the MCU's Safety Management

Unit (SMU), which is a component in the Aurix safety architecture that centralizes all alarm signals related to the various hardware and software safety mechanisms of the MCU. The SMU can be configured to handle each alarm differently depending on severity by for example triggering a reset, generating an interrupt or activating a fault signaling protocol to report the fault to the external environment of the ECU. Both the IOM and the SMU performs logging to enable tracing of alarm sources [25].

5.2 Prototype hardware

The prototype hardware was designed around two evaluation boards: an Infineon Aurix microcontroller board and an ON Semiconductor board for the NCV7519 MOSFET predriver [23]. Furthermore, an Infineon BSP772T high-side switch [26] was used in the testing of concepts one and four from Section 4.4.4. Figure 5.1 shows a schematic of the main part of the prototype, which includes the Aurix TC277TF microcontroller and the NCV7519 for controlling two low-side transistors and a common high-side switch. The two low-side MOSFETs were used to test on/off and PWM signals separately. The NAND logic and the comparator present in the schematic was used to enable or disable the high-side switch and the hardware safety mechanism from concept four in Section 4.4.4.

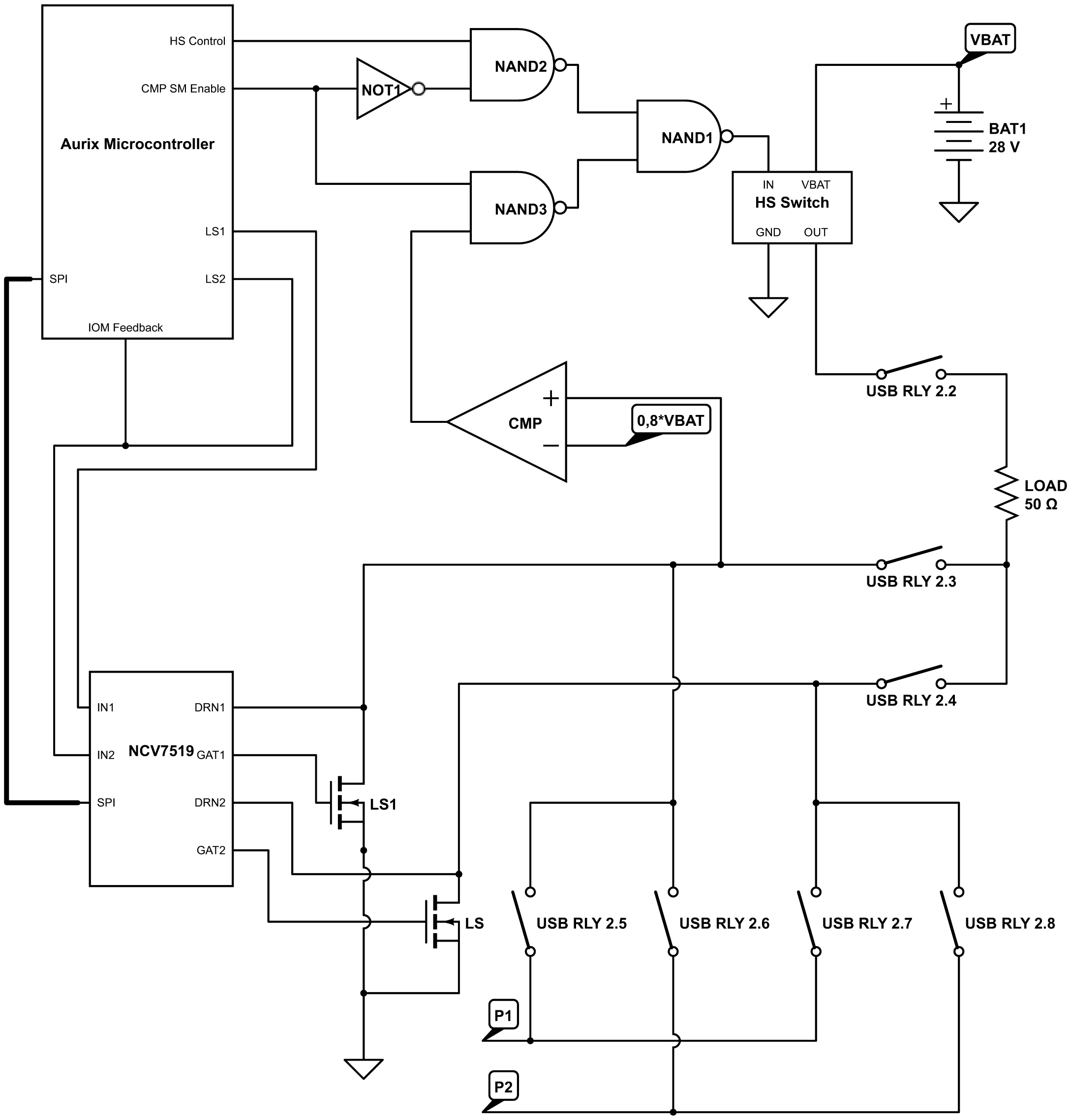


Figure 5.1: Schematic of the main part of the prototype built in the project. The NAND logic is used to switch between the comparator-based and the NCV7519-based diagnosis concepts. P1 and P2 are pins used for oscilloscope monitoring or fault injection. The USB relays are controlled by a PC.

Three USB-controlled relay boards were used in the prototype to enable scripted overrides of switches for testing. The relays were also used for fault injection. P1 and P2 in Figure 5.1 are test planes where fault injection or monitoring hardware can be connected to the low side of the test load. Appendix B includes a schematic of the fault injection circuit used in the prototype through a connection to P2. The fault injection circuit consists of three changeover relays that together with a resistor network can set connections to either V_{BAT} or GND with a variable resistance. By use of this circuitry, it was possible to inject short circuits with varying strengths and polarity to the main circuitry in Figure 5.1.

5.3 Prototype software

The software of the prototype was designed with a similar structure as the current Volvo software of the EMS. This similarity provided a foundation on which the concepts could be implemented and tested in a realistic manner. Another advantage is that it also became easier to compare the prototyped concepts to current solutions used by Volvo GTT. The general structure of the prototype software together with its hardware interface is presented in Figure 5.2.

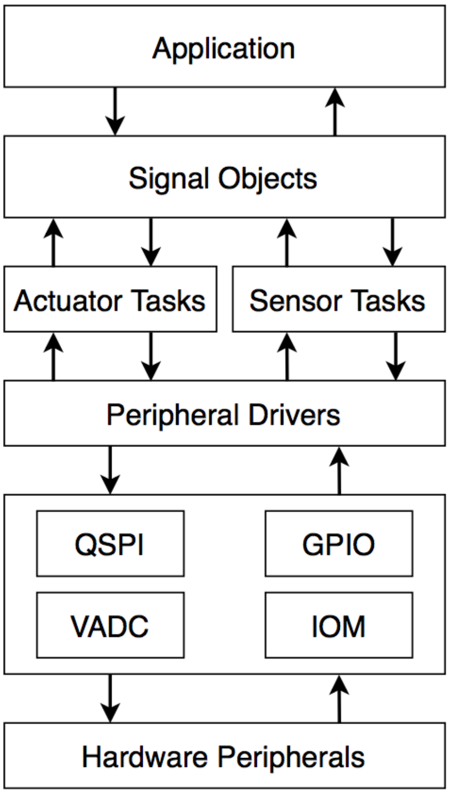


Figure 5.2: The general structure of the abstraction layers in the prototype starting from application software down to hardware peripherals.

The software consisted of several layers in which the application represented the top layer. As writing the application control algorithms of the engine brake function was outside the scope of this project, the application part of the prototype was used to provide stimuli to the actuators and to implement logic for the safety mechanisms.

Similarly to the the software used by Volvo GTT, the application communicated with lower layers using signal objects, which are sets of protected variables accessed using set-and-get functions. There are two kinds of signal objects, one containing the data of a signal, such as the state of an actuator, and another containing the quality of the signal, such as whether or not there is an electrical fault present in the MOSFET driver for the actuator. The layers below the application software is referred to as the platform software.

For creating periodic tasks, the OSEK/VDS compliant real-time operating system ERIKA by ERIKA Enterprises was used [27]. The actuator and sensor tasks were implemented as periodic tasks with periodicities ranging from 1 to 100 ms. These tasks processed the signal objects and based on the results controlled peripheral drivers and updated quality signal objects. The peripheral drivers were used as wrappers for the low-level implementations of the QSPI, GPIO, VADC and IOM peripherals, which were connected to the external hardware.

5.4 Prototype verification

The testing and verification of the prototype was carried out using script-based testing from a PC. Figure 5.3 presents an overview of the setup that was used when performing the tests.

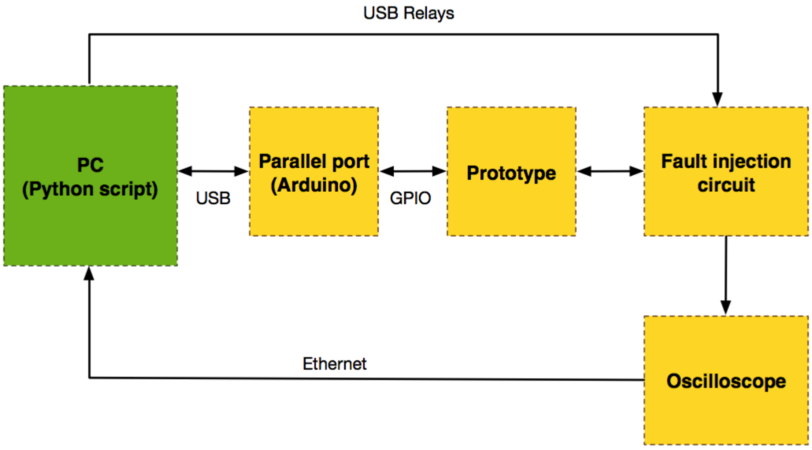


Figure 5.3: An overview of the setup that was used to test the prototype.

By controlling the relays the fault injection circuit in Appendix B and in Figure 5.1 over USB while measuring the outputs of the prototype using an oscilloscope, we injected electrical faults and studied the results. The prototype was connected to the PC through a parallel port realized on an Arduino Uno [28]. The application part of the prototype software was designed to read the parallel port and then to process the command received from the PC through the Arduino. The commands were implemented to control the output pins and to output faults that were detected by the prototype. Furthermore, the tests were written using a custom python test library specifically created for the prototype. The code that handled communication with the relay cards and the Arduino was wrapped in higher-level method calls, which could be called in a scripts-like manner through a top-level python script.

5.5 High-side MOSFET safety mechanisms

The current implementation of the electronics that control the Volvo Combustion Brake is not equipped with a high-side switch. Thus, a short circuit to ground on the low side of the load/actuator could lead to an unexpected activation of the engine brake without any possibility of deactivating the actuator other than to shut down the system. The prototype features the technical safety concepts described in Section 4.4 and Figure 5.1, which realizes supplies mechanisms so that the fault can be avoided. Both concept one and four were tested and verified using the prototype hardware and software with variable short-circuit-to-ground resistance. The result when short circuiting the low side using safety mechanism one (see the first technical safety concept for the VEB actuators, which uses the NCV7519 pre-driver for diagnosis) is presented in Figure 5.4.

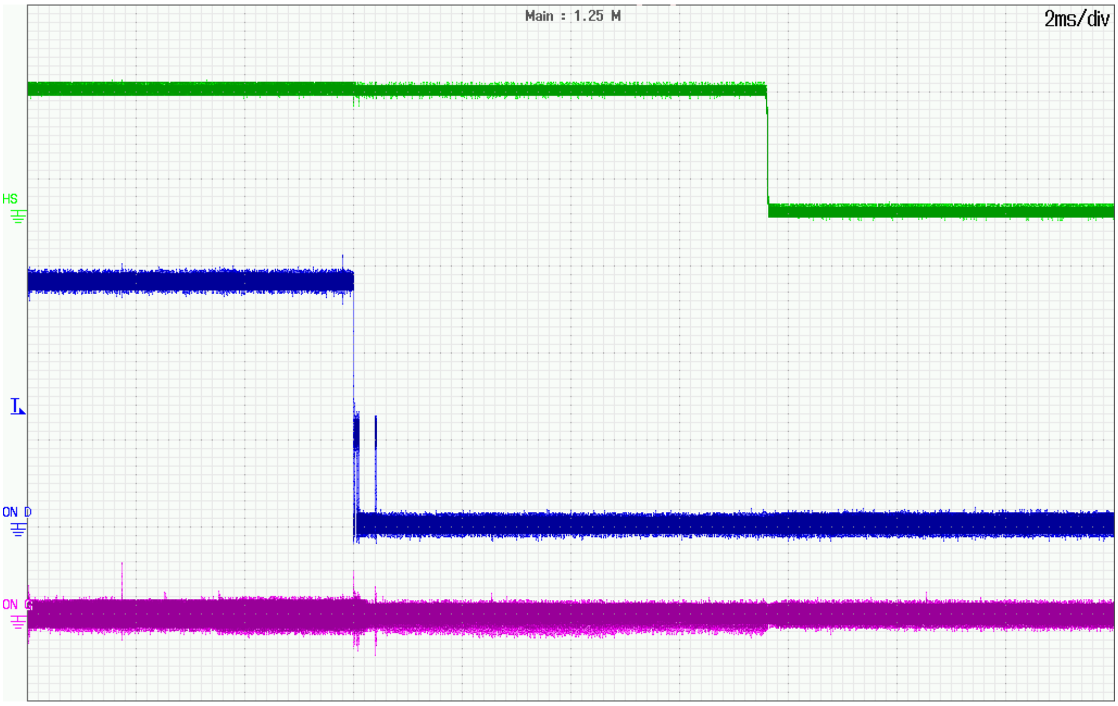


Figure 5.4: Screen capture from the oscilloscope containing the result when injecting a low-resistance short circuit to ground on the low side when safety mechanism one is active. The green (top) channel represents the high-side source voltage (20 V/div), the blue (middle) channel represents the low-side drain voltage (10 V/div) and the purple (bottom) channel represents the low-side gate voltage (2 V/div).

The test in Figure 5.4 was performed by short circuiting the low side without any additional short circuit resistance. From the figure it can be observed that when the short circuit is injected, the voltage at the drain of the low-side MOSFET drops to ground. The ripple is introduced by the relays that were used for applying the short circuit. Approximately 10 ms after the short circuit is injected, the prototype software disables the high side, which deactivates the load. This interval is related to the 10 ms periodicity of the tasks that control the actuators and that poll the pre-drivers for diagnostic status. Therefore, this interval could be shortened by decreasing the period of these tasks down to for example 1 ms. During the time between the injection of the short circuit and the deactivation of the high side, battery voltage is applied over the load. Due to the nature of the solenoids and the oil valves such as the ones that controls the VCB, this interval is considered so short that a faulty activation of the actuator is avoided.

Even though the current implementation of the VCB does not feature a dedicated high side, it is possible to retrieve diagnosis from the MOSFET pre-driver of the EMS through JTAG debugging. When injecting short circuits on the low side of the VCB in the current EMS2.3, it could be observed that the pre-driver was only capable of detecting low-resistance short circuits up to approximately a fifth of the resistance of the load. This is a consequence of that the currently used MOSFET pre-driver performs its diagnosis with a 5 V reference. The updated pre-driver that was used in the prototype has been improved, and instead compares the drain voltage to battery voltage. Consequently, short circuits with higher resistances can be detected. The test in Figure 5.5 presents the result of a short circuit to ground of 50 Ω, which is equal to the resistance of the load.

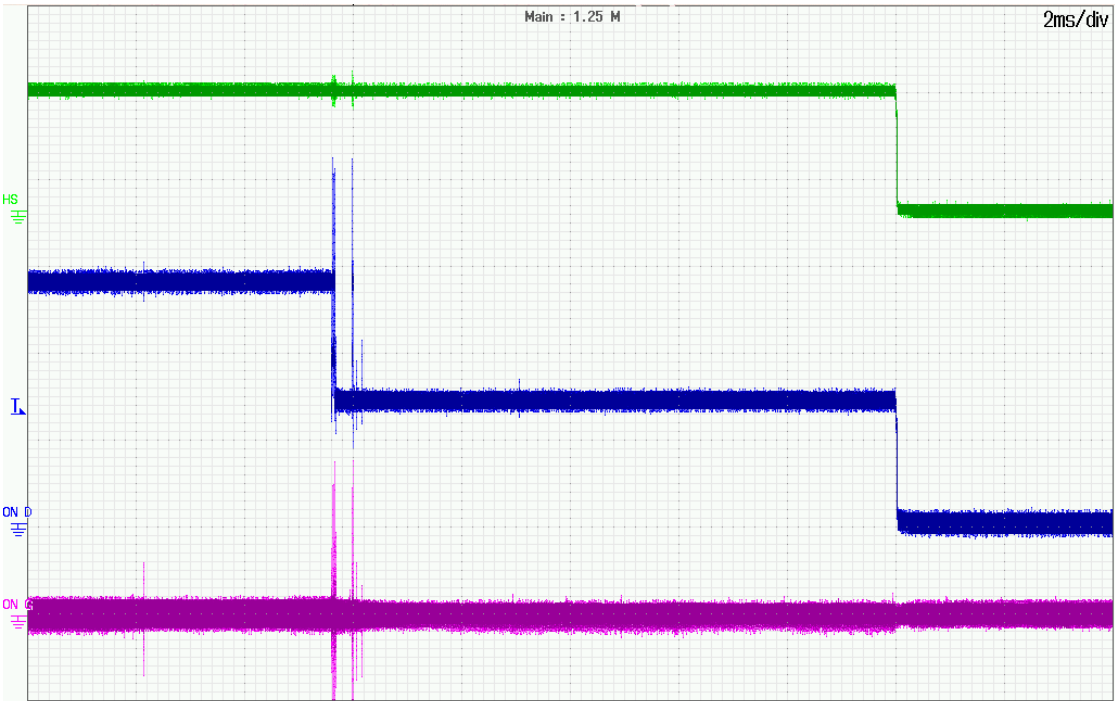


Figure 5.5: Screen capture from the oscilloscope containing the result after injecting a high resistance short circuit to ground on the low side when safety mechanism one is active. The green (top) channel represents the high-side source voltage (20 V/div), the blue (middle) channel represents the low-side drain voltage (10 V/div) and the purple (bottom) channel represents the low-side gate voltage (2 V/div).

By comparing Figure 5.5 to Figure 5.4, it can be observed that approximately half of the battery voltage is now present over the short circuit and drain of the low side. The remaining half of the battery voltage is therefore also present over the load. Unlike with the current design of the EMS, the short circuit is detected and the high side is automatically disabled when the prototype software detects the fault. Short circuits of this nature could occur and they may lead to unexpected behavior of the actuator. Therefore, an higher amount of scenarios are covered by the safety mechanism with the updated pre-driver, resulting in a larger diagnostic coverage.

Safety mechanism two (see technical safety concept four in Figure 4.8) was also tested and verified using the prototype. Similar electrical tests was performed also for this mechanism by injecting short circuits on the low side, intended to provoke a violation of the safety goal considering unexpected activation. The result when injecting a short circuit with low resistance is presented in Figure 5.6.

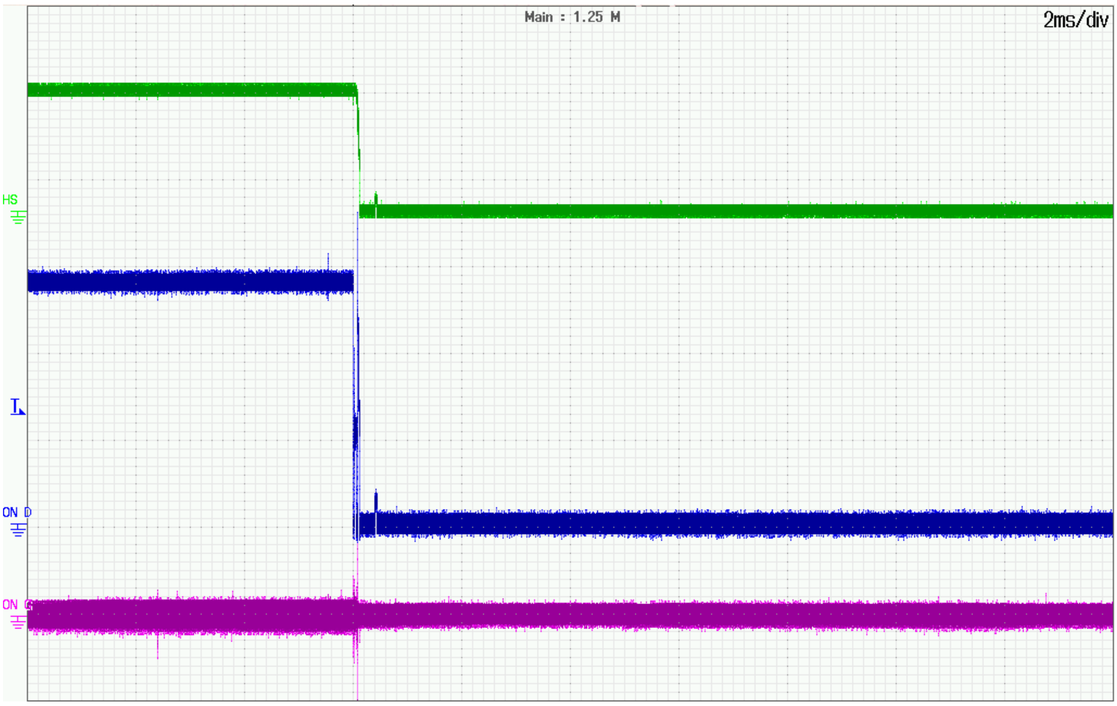


Figure 5.6: Screen capture from the oscilloscope containing the result when injecting a low resistance short circuit to ground on the low side when safety mechanism two is active. The green (top) channel represents the high-side source voltage (20 V/div), the blue (middle) channel represents the low-side drain voltage (10 V/div) and the purple (bottom) channel represents the low-side gate voltage (2 V/div).

From Figures 5.5 and 5.6 we observe that the second safety mechanism is significantly faster than the first one. This is expected since the diagnosis in the second solution uses a dedicated comparator that has direct control of the high-side switch; no interaction with software is required. The threshold for disabling the high side is explicitly set by setting a threshold voltage at one of the inputs of the comparator. The VCB actuator is guaranteed to activate when applying a voltage over 18 V. Thus, in the prototype, the threshold was set to 22.4 V by the use of a voltage divider from battery voltage. This voltage can be adjusted depending on the actuator specification by adjusting the resistances used in the voltage divider. If the voltage at the drain of the low-side MOSFET is below 22.4 V, the high side is automatically switched off as can be seen in the figure. Therefore, this safety mechanism is tolerant to both high- and low-resistance short circuits.

Chapter 6

Discussion

This thesis work was aimed to be an initial study for the Volvo GTT engine ECU department of how their control units have to be adapted for the future release of the second revision of ISO 26262. As such, the question to be answered by the thesis was of a general kind. Considering this, the thesis answers the question by presenting a summarized description of the relevant parts of ISO 26262, while also discusses concepts for how the ECU subsystems could be adapted to comply with the standard, both in a general manner and in the perspective of the engine brake case study.

The project has showed how to adapt the current design of the engine brake control system to comply with ISO 26262. Based on the concepts and new microcontroller architectures discussed throughout the thesis, no radical changes are required of the ECU hardware to ensure compliance of future designs. Still, any additional hardware such as the suggested comparators or fault injection circuitry increases the complexity of the system, which in turn may also lead to unforeseen fault sources. It is up to the engineers developing the next-generation platforms to consider the suggested concepts and the recommendations regarding designs for ISO 26262, and compare these when additional data such as cost and application requirements are available.

To limit the scope of the thesis work, the application part of the Volvo GTT software was not analyzed in detail. This restriction was likely the most limiting part of the project as it was challenging to perform a case study of the engine brake system without considering the software governing its functionality. However, the aim of the project is considered to have been met regardless of this, since the purpose of performing the case study was to simplify the process of discovering and presenting solutions that can be generalized to other subsystem, which has been accomplished. It was found that the case study was necessary for being able to structure the project according to the ISO 26262 safety development cycle, but that it was more efficient to test and present the prototyped solutions individually.

The following sections discuss key parts of the project in more detail.

6.1 The next revision of the standard

The concepts and solutions presented in this thesis were derived according to the 2011 version of the ISO 26262 standard which applies only to light-duty vehicles. The next revision of the standard, which will apply also to heavy-duty vehicles, is not likely to introduce any conceptual changes. However, according to sources that are part of the development of the next revision of the standard, it is likely that there will be higher expectations of the drivers of heavy-duty vehicles than on car drivers. The reason for this is that the driver is considered to be a professional driver with more experience than a driver of a personal vehicle. Higher expectations would imply that one could assume a higher controllability of the hazardous situations when performing the hazard analysis, possibly resulting in a lower ASIL. On the other hand, a heavy-duty vehicle could also increase the severity of hazards which may increase the ASIL.

6.2 The prototype

As described in Section 5.5, two different safety mechanisms were implemented for avoiding violation of safety goal SG1 in the event of a short circuit on the low side of the VCB actuator. The first safety mechanism was designed so that the application software was responsible for deactivating the high-side switch before the safety goal was violated. The second safety mechanism was implemented directly in hardware so that the high side was deactivated before the safety goal was violated without interaction from the application software. The second safety mechanism proved to be faster than the first one with the cost of increased hardware complexity. Furthermore, the increased complexity might introduce new potential hazards as extra circuitry is added to the design. It is for example possible that the comparator-based design has a higher chance of false triggering due to its quickness.

A complete analysis of the software that governs the engine brake was beyond the scope of the thesis work. Such an analysis could imply that several changes would be required for the software responsible for diagnosing the actuators to comply with ASIL C or D. Therefore, the second safety mechanism may be preferable since these parts of the software would no longer be included in the analysis - the diagnosis and fault handling would be performed entirely by hardware. According to the results presented in Section 4.4.6, there is enough margin regarding the ASIL C target values to cover for the extra hardware complexity introduced by this safety mechanism.

6.3 Diagnostic coverage

The main purpose of the calculations of failure rate metrics described in Section 4.4.6 is to exemplify how to apply the part of the standard presented in Section 2.6.1, and to discover any challenges in this process. Our experience from performing the calculations is that the challenge is to determine the diagnostic coverage of the safety mechanisms in the system. If the guidelines for diagnostic coverage of general faults in typical elements as presented in Part 5 of the standard is followed, significantly more complex and expensive solutions would be required. For example, the standard recommends digital outputs to be monitored for several kinds of stuck-at faults, but also for drift and oscillations. For a simple part such as an actuator, it is clear that all such faults do not require any time-critical diagnostics. On the other hand, ISO 26262 emphasizes that diagnostic coverages should be calculated with regard to the safety goal; it also often allows exceptions if any arguments can be made from previous experience. Thus, it appears that it is up to the integrator to motivate their own method for estimating the effectiveness of their safety mechanisms.

Overall, the purpose of ISO 26262 is not to standardize actual implementations electronic systems, but rather to provide a common ground for manufacturers to create a more efficient safety process, ultimately resulting in increased product safety. Because of this, it can be recommended that methods for determining essentials such as diagnostic coverages are developed with regard to how partners and suppliers are adapting their processes. Of course, collaboration is also advised for other parts of the safety process such as the hazard analysis. Naturally, the investigation performed in this project is made from an engineer's point-of-view – a proper safety development cycle should developed for Volvo that clarifies the challenges mentioned in this thesis in such a way that a common method for performing electronic hardware safety evaluation is available.

6.4 Ethical aspects

The purpose of ISO 26262 is to provide manufactures with common means to document and measure the safety of their electrical systems. By adapting to the standard, automotive companies should be able to achieve a development process that can identify potential safety hazards and reduce the risk of them occurring to a low level, reducing the risk to human life.

In addition to the motivation of saving lives and the ability of claiming safe products, companies may use ISO 26262 to avoid safety-related legal claims against them. Recent events in which car manufacturers have been forced to re-call or repair millions of vehicles has likely led to companies searching for a common legal base for such situations. Although less altruistic than the previously mentioned motivations, any reason for designing with ISO 26262 may still result in increased safety of end users.

Another ethical aspect to consider is that even if proper safety mechanisms are included in the system and all the requirements of the standard are met, the driver can in many cases disregard the safety measures. This is especially true for measures designed to detect and warn for possible latent faults; if the driver warning is disregarded, the safety mechanism would have no effect.

This thesis provides design engineers at Volvo GTT with concepts and guidelines regarding how to design ECUs in compliance with ISO 26262. Thus, the aim is that the thesis will assist them in creating even safer vehicles in future development projects.

Chapter 7

Conclusions

The purpose of this project was to provide an analysis to predict what will be required in future Volvo GTT powertrain control units in order to comply with the second iteration of ISO 26262 when it is released in 2018. Therefore, a significant part of this report is allotted to describing the standard and how to apply it. The work with analyzing and presenting the standard itself has showed that even though the standard documents are detailed and require following a significant amount of references to different parts of the standard, the actual process of performing hazard analysis, safety goal formulations, technical requirements and design evaluation can be summarized and explained in a clear way.

The Volvo Engine Brake case study has shown that the current engine brake implementation in the engine ECU would not comply with ISO 26262 due to insufficient safety mechanisms – especially regarding diagnosis and control of faults in the microcontroller and the engine brake actuator control circuitry. However, an evaluation of a technical concept and of a prototype including the required safety mechanisms has shown that the system can be adapted to comply with the standard without introducing a significant amount of additional hardware. Still, it has been made clear that regardless of which of the suggested safety mechanism concepts that are used in future implementation of the system, a high-side switch is required to complement the low-side switches of critical output pins so that actuators can be disabled in the event of a critical fault.

The project has showed that one of the more challenging parts of designing hardware solutions with regard to ISO 26262 is to estimate the diagnostic coverage of safety mechanisms. To strictly follow the generalized examples in the standard regarding safety mechanisms and diagnostic coverage would result in a poorly tailored and unnecessarily costly implementation. Thus, if an overly conservative strategy is undesirable, standardized ways of assessing safety mechanisms should be developed in collaboration with industry partners.

Acronyms

ADC Analog to Digital Converter

ASIL Automotive Safety Integrity Level

AVU Air Valve Unit

CAN Controller Area Network

CRC Cyclic Redundancy Check

DC Diagnostic Coverage

DC Duty Cycle

E/E Electrical and Electronic

ECU Electronic Control Unit

EMS Engine Management System

FIT Failure In Time ($10^{-9}h^{-1}$)

FMEDA Failure Modes Effects and Diagnostic Analysis

FRS Functional Safety Requirement

FTA Fault Tree Analysis

GPIO General Purpose Input and Output

GPT Generic Purpose Timer

GTM Generic Timer Module

HARA Hazard Analysis and Risk Assessment

HW Hardware

I/O Input/Output

IC Integrated Circuit

IOM Input Output Monitor

LAM Logic Analyzer Module

LFM Latent Fault Metric

MCU Microcontroller Unit

MOSFET Metal-Oxide-Semiconductor Field-Effect Transistor

MPF Multiple-Point Fault

PMHF Probabilistic Metric for Random Hardware Failures

PWM Pulse Width Modulation

QM Quality Management

QSPI Queued Serial Peipheral Interface

RF Residual Fault

SBC System Basis Chip

SEooC Safety Element out of Context

SG Safety Goal

SMU Safety Management Unit

SoC System on Chip

SPF Single-Point Fault

SPFM Single-Point Fault Metric

SPI Serial Peripheral Interface

SW Software

USB Universal Serial Bus

VCB Volvo Compression Brake

VEB Volvo Engine Brake

Volvo GTT Volvo Group Trucks Technology

Bibliography

[1] *International Standard ISO 26262 - Road vehicles — Functional safety*, ISO, Geneva, 2011.

[2] P. Srivastava, M. L. Karle, U. S. Karle, and A. A. Deshpande, "Development of Electrical Power Assisted Steering (EPAS) Considering Safety and Reliability Aspects as per ISO 26262," in *SAE Technical Paper.* SAE International, 01 2015.

[3] S.-H. Jeon, J.-H. Cho, Y. Jung, S. Park, and T.-M. Han, "Automotive hardware development according to ISO 26262," in *Advanced Communication Technology (ICACT), 2011 13th International Conference on*, Feb 2011, pp. 588–592.

[4] K. Beckers, M. Heisel, T. Frese, and D. Hatebur, "A structured and model-based hazard analysis and risk assessment method for automotive systems," in *Software Reliability Engineering (ISSRE), 2013 IEEE 24th International Symposium on*, Nov 2013, pp. 238–247.

[5] N. Adler, S. Otten, P. Cuenot, and K. Müller-Glaser, "Performing Safety Evaluation on Detailed Hardware Level according to ISO 26262," *SAE Int. J. Passeng. Cars - Electron. Electr. Syst.*, vol. 6, 04 2013.

[6] M. Ellims, H. Monkhouse, and A. Lyon, "ISO 26262: Experience applying part 3 to an in-wheel electric motor," in *System Safety, 2011 6th IET International Conference on*, Sept 2011.

[7] *Freescale Qorivva*, Freescale, [Accessed 2015-02-03]. [Online]. Available: http://www.freescale.com/webapp/sps/site/homepage.jsp?code=POWER_ARCH_5XXX

[8] *Infineon Aurix family*, Infineon Technologies AG, [Accessed 2015-02-09]. [Online]. Available: http://www.infineon.com/cms/en/product/microcontroller/32-bit-tricore-tm-microcontroller/aurix-tm-family/channel.html?channel=db3a30433727a44301372b2eefbb48d9

[9] *EMS2.3 HW Performance specification: PD1:59323942_006.ADOC_REV;6*, TRW Automotive.

[10] *EMS2.3 Schematic: PD1:59319774_008.A-DOC_NOREV;B*, TRW Automotive.

[11] *EMS2.3 Interface Control Description: PD1:05901583_038.ADOC_ REV;6*, TRW Automotive.

[12] *EMS2.3: Volvo Technical Requirements: 21368090.06*, TRW Automotive.

[13] *International Standard ISO 26262 - Part 5: Product development at the hardware level*, ISO, Geneva, 2011.

[14] *International Standard ISO 26262 - Part 2: Management of functional safety*, ISO, Geneva, 2011.

[15] *International Standard ISO 26262 - Part 3: Concept phase*, ISO, Geneva, 2011.

[16] *International Standard ISO 26262 - Part 8: Supporting processes*, ISO, Geneva, 2011.

[17] *International Standard ISO 26262 - Part 4: Product development at the system level*, ISO, Geneva, 2011.

[18] *International Standard ISO 26262 - Part 9: Automotive Safety Integrity Level (ASIL)-oriented and safety-oriented analyses*, ISO, Geneva, 2011.

[19] Y.-C. Chang, L.-R. Huang, H.-C. Liu, C.-J. Yang, and C.-T. Chiu, "Assessing automotive functional safety microprocessor with ISO 26262 hardware requirements," in *VLSI Design, Automation and Test (VLSI-DAT), 2014 International Symposium on*, April 2014, pp. 1–4.

[20] *Infineon Aurix Safety Manual*, Infineon Technologies AG, 2015.

[21] P. Sinha, "Architectural design and reliability analysis of a fail-operational brake-by-wire system from ISO 26262 perspectives," *Reliability Engineering & System Safety*, vol. 96, no. 10, pp. 1349 – 1359, 2011.

[22] V. Hiligsmann and P. Riendeau, "Full redundant programmable linear Hall sensor," in *Sensors, 2003. Proceedings of IEEE*, vol. 1, Oct 2003, pp. 222–228 Vol.1.

[23] *NCV7519 Low-side MOSFET Pre-driver - Data sheet*, ON Semiconductor, 2014.

[24] *Siemens Standard SN29500*, Siemens AG, 2004.

[25] *Infineon Aurix TC27x User Manual*, Infineon Technologies AG, 2014.

[26] *BSP772T Smart Power High-Side-Switch - Data sheet*, Infineon Technologies AG, 2004.

[27] *ERIKA OS*, ERIKA Enterprise, [Accessed 2015-05-13]. [Online]. Available: http://erika.tuxfamily.org/drupal/

[28] *Arduino Uno*, Arduino, [Accessed 2015-05-30]. [Online]. Available: http://www.arduino.cc/en/Main/ArduinoBoardUno

Appendix A

FMEDA

Component Name	Failure rate/FIT	Safety-related component to be considered in the calculations?	Failure Mode	Failure rate distribution	Failure mode that has the potential to violate the safety goal in absence of safety mechanisms?	Failure mode coverage wrt. violation of safety goal	Residual or Single-Point Fault failure rate/FIT	Failure mode that may lead to the violation of safety goal in combination with an independent failure of another component?	Failure mode coverage with respect to latent failures	Latent Multiple-Point Fault failure rate/FIT
C1	2	YES	Open	22			0			0
			Parameter Change	29			0			0
			Short	49	X	95	0,049			0
R1	2	YES	Open	84			0	X	90	0,168
			Parameter Change	11			0			0
			Short	5			0			0
D1	3		Open	36			0			0
			Parameter Change	15			0			0
			Short	49			0	X	90	0,147
Z1	3	YES	Open	45			0			0
			Parameter Change	35			0			0
			Short	20			0			0
TR1	5	YES	Open	5			0			0
			Parameter Change	17			0			0
			Short	51	X	95	0,1275			0
	5	YES	Output Low	22	X	95	0,055			0
			Output High	5			0			0
							0			0
R2	2		Open	84			0			0
			Parameter Change	11			0			0
			Short	5			0			0
C2	2	YES	Open	22			0			0
			Parameter Change	29			0			0
			Short	49	X	95	0,049			0
R3	2		Open	84			0	X	90	0,168
			Parameter Change	11			0			0
			Short	5			0			0
D2	3		Open	36			0			0
			Parameter Change	15			0			0
			Short	49			0	X	90	0,147
Z2	3	YES	Open	45			0			0
			Parameter Change	35			0			0
			Short	20			0			0
TR2	5	YES	Open	5			0			0
			Parameter Change	17			0			0
			Short	51	X	95	0,1275			0

	5	YES	Output Low	22	X	95	0,055			0
			Output High	5			0			0
							0			0
R4	2		Open	84			0			0
			Parameter Change	11			0			0
			Short	5			0			0
C3	2	YES	Open	22			0			0
			Parameter Change	29			0			0
			Short	49	X	95	0,049			0
R5	2	YES	Open	84			0	X	90	0,168
			Parameter Change	11			0			0
			Short	5			0			0
D3	3		Open	36			0			0
			Parameter Change	15			0			0
			Short	49			0			0
TR3	5	YES	Open	5			0			0
			Parameter Change	17			0			0
			Short	51	X	95	0,1275			0
	5	YES	Output Low	22	X	95	0,055			0
			Output High	5			0			0
							0			0
NCV7519	38	YES	Diagnosis failure	50			0	X	90	1,9
			Output error	50	X	95	0,95			0
							0			0
Sensor	50	YES		50	X	99	0,25			0
				50			0			0
							0			0
							0			0
							0			0
							0			0
							1,8945			2,698

Total failure rate:	149
Total safety related:	134
Total not safety related	15

METRICS	VALUE	TARGET
Single point fault metric	98,59	97
Latent fault metric	97,9576929	80
PMHF	1,900181729	100

Appendix B

Fault injection circuit

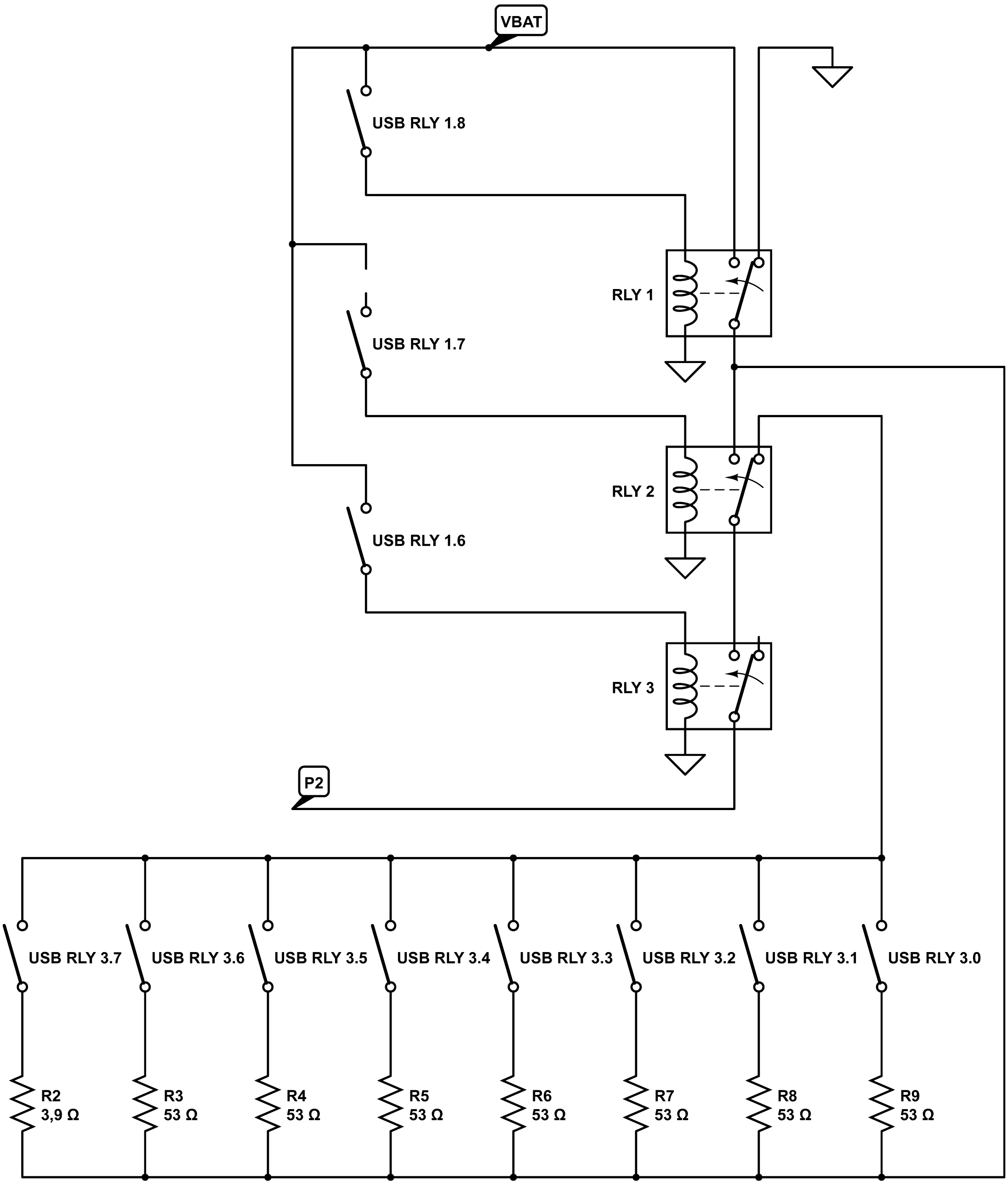


Figure B.1: Schematic of the fault injection circuit used in the prototype. PC-controlled relays are included which can introduce connections on P2 to either V_{BAT} or GND with a number of different resistances set by a resistor network. P2 is connected to the main ciruit in Figure 5.1 of Section 5.2.