The challenge of safety and security in automotive systems

Conference Paper · May 2014

DOI: 10.1109/SACI.2014.6840056

CITATION
1

READS
291

2 authors:

Catalin-Virgil Briciu
Polytechnic University of Timisoara
4 PUBLICATIONS **12** CITATIONS
SEE PROFILE

Ioan Filip
Polytechnic University of Timisoara
94 PUBLICATIONS **259** CITATIONS
SEE PROFILE

The Challenge of Safety and Security in Automotive System

Catalin-Virgil Briciu*, Ioan Filip*
* "Politehnica" University of Timisoara/Faculty of Automation and Computer Science, Timisoara, Romania
briciucatalin@yahoo.com, ioan.filip@aut.upt.ro

***Abstract*— In the same time with the growing of complexity for E/E systems, the level of safety needed to be fulfilled by the work products increased very fast. Could we determine the way to fulfill a standard safety level for all manufacturers? Are these standardized and applicable? The article sheds light these standards and provides the basic knowledge to design a functional safety system from the software point of view. Functional safety concepts are described in the ISO 26262[1] standard where concepts as ASIL, risk assessment methods and hazards analysis are described very clear. The article briefly describes these concepts in a manner related to software development. Also, in AUTOSAR complaint system the needed for functional safety concepts is very huge because in the context of standardized interfaces between modules can leads also to some errors. But for avoiding this, the AUTOSAR requirements provide some methods that are taken into consideration and described alsoin the article. Last part of the article presents a lightweight implementation of a safety system considering as use case the designing of a remote keyless entry system.**

I. INTRODUCTION

Nowadays, the automotive industry is continuously changing due to increasing of the customer amount and complexity of specifications and new technologies, but nevertheless of their concerning about "The Internet of Things" (interconnectivity between all electronically devices) because they want that their devices interconnected. Also, they should take into consideration the law regulations regarding the environment and safety. To resume all things from above, can be considered that the main concerns are about building safe, connected and green embedded systems[2]. In this article, the focus will be on the concepts that make a system more and more safe. To be declared a safe system, it shall fulfill some "functional safety requirements" and the development of it (including both software and hardware) should be made after a predefined process.

What is interesting is the fact that safety laws were defined also in the ancient world. As an example, the Codex Hammurabi (~1780 BC)[3] may serve as a basis. This was needed in those times because the technology level was very low and a guarantee of the "hand-made" work is more than needed to achieve one "technical" system. But now, do we need such laws because almost all actions of the worker are aided by a computer? In this way, such examples are relevant and this is some of most discussed topic, but behind the "open doors" there are many discussions between the customer and automotive companies. It is known the case of Toyota problem with accelerator pedal that kills one people. For this, Toyota calls back about 8.5 billion cars to the garage and also some penalties to the injured peoples. This cause a lot of waste money and time also for company and customers. Honda Civic defect chip in a DC/DC converter that because of a short cut to ground blows a main fuse and the motors stops. In Europe about 3751 cars were affected and call back to producer. Another well-known case is of a loss of power steering on the Mazda cars. Three accidents are known and checked currently by investigation agents. These three cases should raise the question that "Is indeed needed the functional safety regulations?"

Growing complexity of embedded systems (means from both software and electronic point of view) in automotive industry leads to introducing of an automotive standard to assure the functional safety requirements. The standard is called ISO 26262 and defines the process for development of the automotive equipment and introduces the actions that should be taken in case of possible hazards caused by malfunctions of electronic and electrical systems in passenger vehicles. The standard is derivate from the IEC 61508 standard, the generic functional safety standard for electrical and electronic (E/E) systems. ISO 26262 standards was created and released as a draft version in June, 2009 and since then for the lawyers this was considered as a state of art. Based on this, a court process could be started against the automotive companies. Confusion between functional safety and quality is made almost every day by the unfamiliar developers with this topic. Quality could be defined as something that "missing of it annoys", but functional safety refers to something that "missing of it hurts". From the "Vocabulary" section of the ISO 26262 standard a definition for functional safety can be considered: "Absence of **unreasonable risk** due to hazards caused by malfunctioning behavior of E/E systems". In this context, the unreasonable could be considered both unacceptable,

excessive and risk could be considered the combination of the probability and extent of the damage. Failures of the automotive systems can be failures due to hardware (some short cuts, low/high voltage) or maybe failures at the design level (e.g. software errors). The minimization of the risk means at first seen the reducing of the probability of errors, but this is more complicated than this. In fact, minimization of the individual components will not lead to elimination of the risk associated to entire system. Reducing of the error occurrence probability below of a predefined threshold given by risk assessment of the entire system with all interconnected sub-systems (e.g. software components, sensors, actuators, mechanical devices, external devices) in fact should be considered as fulfilling the functional safety requirements.

The following chapters should be considered as key parts of the ISO26262 standard[4]: *Vocabulary, Management of functional safety, Concept phase, Product development at the system level, Product development at the hardware level, Product development at the software level, Production and operation, Supporting processes, ASIL (Automotive Safety Integrity Level)and safety oriented analysis and Guideline on ISO26262*. From these chapters, it could be seen that the standard provides a lifecycle (starting from quotation, concept refinement, development, industrialization, product validation) methodology for development of the automotive systems and also add methods to support tailoring of the necessary activities for the phases of the lifecycle process. Also, the aspects of the functional safety of the entire development (*Prototype planning, Specifications, Design and realization, Integration &Test)* process are defined inside of these chapters. The risk classes are described using the methods inherited from the automotive specific risk based approach and the ASIL methods are used for realization of the specifications for achieving an acceptable risk. Based on these specifications defined previously, architecture is realized and then the integration and validation test cases and measurement for detecting if the considered risks are below the predefined threshold and are considered acceptable. In this case, the functional safety system is considered as a safe one; otherwise another development cycle should be applied again.

II. AUTOSAR AND FUNCTIONAL SAFETY

For a better understanding of the following that will be described it is very good to understand the difference between safety and security. These two concepts go "hand-in-hand" [5], but refer to two different things. Safety was described above as "absence of unreasonable risk" due to hazards caused by malfunctioning behavior of the E/E systems", whereas "security" means protection of the system by undesired access from other components. AUTOSAR provides some techniques to be taken into consideration when a functional safety system is developed. To develop a system that should be considered fully AUTOSAR compatible some strategies such the ones from below should be applied.

a) **Memory partitioning**: these method allows firstly that safety and non-safety SWC (software components) to be developed on the same ECU (Electronic Control Unit). Separation of the components from the memory point of view can be made allowing each component a memory area predefined. For example, such IDEs for developing automotive applications allows developers through "pragma" C option to define their region for the variables. This job could be made also by the system architect who can define at the starting of the project section as "START_SWC_1_RAM_INIT_BYTE_AREA" and "START_SWC_2_RAM_INIT_BYTE_AREA" and also their "STOP_" equivalents. Also such definition should be made for each data type and the developers will use these macro definitions while developing their SWCs. Defensive programming could be consider relevant also for this method because allows communication between components using a safe way. In [1], authors described the safe communication between SWCs using the Real Time Environment Layer. Each SWC provides some interfaces and parameters describes in some "ecuconfig" files delivered by AUTOSAR software architect. At micro scale, through defensive programming means the numerical protection (division by zero, NULL pointers, cast to integral types, range limits, saturation and overflow/underflow). Also the ECC (Error Correcting Code) methods are used here to assure the integrity of the FLASH/EEPROM areas. Together with this ECC methods, the consistency check methods and data retention survey are used.

a) **Program flow monitoring**: is a method these help developers to implement the required 2oo3 (two out of three) decision needed in the functional safety systems. In this case, at specified moments of time some inputs or outputs could be checked and also if the requirements about timing constraints are met. In this category, also the support for dual controller plays an essential role because in this case, another "watch-dog" controller can supervise the core (main controller).

b) **Time determinism and time constraints** modeling allows that SWC and BSW (Basic Software) modules to known in each point of execution exactly if the timing constraints are met or not and. Also, this allows synchronized time bases between networks, components, hardware jitters and also could determine in the run-time the timing violations parts of code. Also, in the operating system (e.g. AUTOSAR OS and OSEK) these constraints are very clear specified, otherwise the task scheduling mechanism is not working properly. For example, we have runnable tasks (e.g. cyclic task, 5ms, 10ms, 20ms, interrupts) and in this case if the execution of the operations inside of the 5ms task is greater than 30% of the lower task period then in case of some interrupts and maybe at some points where more than one cyclic task operation should be executed, a jitter is introduced in the

operating system and this means that the time base is changing. This is not well because the system is not deterministic and time constraints requirements defined at the beginning of the projects are not met.

c) **End to end communication protection**. Accordingly to [6], the concept of end to end communication (known as E2E communication) protection assumes that safety-related data exchange shall be protected at run-time against the effects of faults within the communication link. As faults can be considered the following: random HW faults (e.g. corrupts register of the SPI interface, bad design in the overflow range), design systematic errors (e.g. bugs in software in at least one communication stacks, not all requirements were implemented in the communication layer through Virtual Function Bus layer or RTE layer) or maybe environmental faults caused by interferences or natural hazard phenomena. The benefits of using this E2E protection is that the errors (faults) in the communication links will be find in the integration phase.

d) **Logic and BIST (Built-In-Self Test) methods** [7]. In case that functional safety is necessary in the system, then the standards recommend having a chip with the BIST capabilities. This means that before setting the chip into a functional mode, it is necessary to perform this BIST tests to detect any possible memory or hardware errors. (E.g. some memory cell are not usable anymore) coming from the production modes. There are two possibilities for implementing BIST capability: chips can have implementing another controller to perform this operations, or hardware monitors (dedicated devices) for this.

e) **Hardware specific devices**. Devices like watch-dog timers, glitch filters, and self-correcting hardware devices can be classified as being part of this category. Watch-dog timers have the responsibility to lead the program execution or the hardware device (maybe both) in a safe state, in case some anomalies are detected in the software. These anomalies can be writing/reading operations from un-allocated memory, a hung situation that can leads to physical damages in best case. A possibility to implement this is a periodical task to check the state of the system and if a task is detected to not get out in a predefined time then the system is considered to be hang-out leading to watch-dog intervention and correcting this behavior putting the program execution in a safe state. Glitch filters are used for critical pins like reset, PLL (phased lock loop), communication pins for removing noises that can appears due to electromagnetically fields for example.

Security can be assured in functional safety implementing the following requirements:

a) An electronic immobilizer (known as IMMO) shall be used to protect the car from driving from the unauthorized people or entities. IMMO is built using a transponder chip which communicates through low frequency waves with the key-fob slot and terminal control of the car. The communication is realized using some cryptographic functions such AES128, HITAG2 or HITAG3, but this is OEM specific.

b) The ECU shall be protected from flashing (programming) from unauthorized entities. Such requirement can be fulfilled if through programming a flash boot loader application decides based on some logic if the flashed programmed is an original OEM program or just a “malware” application. Also this is based on some cryptographic security functions.

c) Diagnosis application for configuring the car variant and setting/getting the DTCs (Data Trouble Code) shall be protected and only authorized diagnosis application should be accepted. This is realized with some security codes.

d) Depending on the national laws of the country/countries in which the car is supposed to run the car variant is configured based on them and the handling should be made only in some specific configurations. Not all configurations of parameters shall be available for one car.

III. FUNCTIONAL SAFETY CONCEPT

The functional safety concept means the description of the functional correlations that should be achieved to fulfill the safety goals and standards. For each safety goal, some functional safety requirements should be defined. This definition of the functional safety concept is made through several steps which will be described below.

First step refers to Hazard analysis and Risk assessment. In this step the identification of the hazardous events and assessment of the corresponding risk for humans on the vehicle level and then based on this the relevant functional safety goals are determined. After this step has been realized, it is needed a review from an independent source of it can be made also from the customer. Here the risks could be classified based on three concepts [8]: severity, frequency of event (known in automotive as exposure) and actions (or controllability) and also the ASIL (Automotive Safety Integrity Level) as described in Figure 1. Severity describes the possible damages of the hardware, equipment, environment or possible injuries of the peoples. An action describes the possibility to reduce/avoid the possible faults and also reducing the frequency to zero or duration of it below a threshold where the damages are controllable.

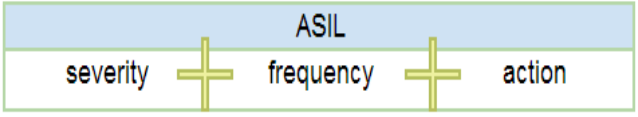


Figure 1. ASIL structure

Each safety requirements is assigned to an ASIL or QM (in case that document quality level is enough for that functionality) and based on this the minimum testing requirements. For example, for an ASIL D (the highest level) the tester should be independent by the development team and unit testing is required with 100% code and decision coverage. In Figure 2, the level of ASIL concept is described.

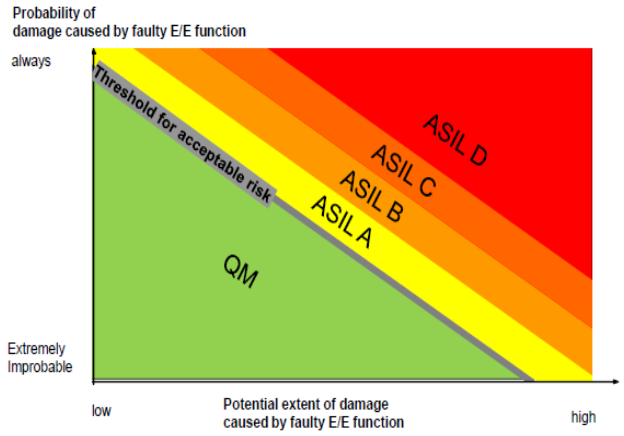


Figure 2. Functional safety overview

Below an example is presented (based on 2010 Hyundai Veracruz Recall issue).

TABLE I. EXAMPLE OF FAULT ANALISYS

Item	Content
Item ID	#1
Malfunction description	The lamp switch may malfunction.
Possible scenario	A malfunction of the lamp switch may cause the brake lights to not illuminate when the brake pedal is depressed and to remain illuminated when the brake pedal is released.
Reason category	Typical driving scenario.
Reason for functional safety	Prevent the other participant of on the traffic about your intentions.
Severity	Collisions with delta v > 40 km. Life-threatening injuries or fatal injuries cannot be excluded.
Frequency	Driving at night
Action	Less than 10% of average drives or read users are normally able to avoid the hazardous by activation/deactivation of the brake lights. It is assumed that most of the drivers see if the driver in front of them reduces the speed, but maybe it will take too much time to brake in a hazardous situation.
Fault Tolerance Time	500 milliseconds
ASIL Category*	B
Safe State	The brake lights illuminates when the brake pedal is depressed and not illuminate when the brake pedal is released.

The next steps are referring to description of the Functional Safety Concept. Firstly, it is necessary to describe the functional properties for the use cases found at step 1. For each of them, another table is made based on the **Item ID** and a detail explanation and quantification of the terms used shall be done. In case that are some emergency methods are needed, it is mandatory to define them. On the next steps, the functional safety requirements on function level representing the safety strategy shall be created. These will contain in which way the functional safety goal will be achieved and not implemented. Also it is possible to define requirements outside of the analyzed functions, but which will help achieving the safety goal (e.g. mechanical dimensions, warnings signs).Identification of the needed elements that can be used for achievement of this safety goal can be first identified in the current preliminary architecture in case that the system is a new one, or already in the frozen architecture if the system is a reuse one. Very important to understand here is the fact that these elements should be functional due to the fact that functional safety concept supposed to be made at function level. The next steps consist from definition of warnings and degradation concepts, definition of the driver's action or other endangered person. At this level, it shall be stated what will happen if a dangerous fault has been detected (e.g. driver warning, emergency operation) and also what action is needed to be taken from the driver side to be made in case that warnings of malfunction occurs. The

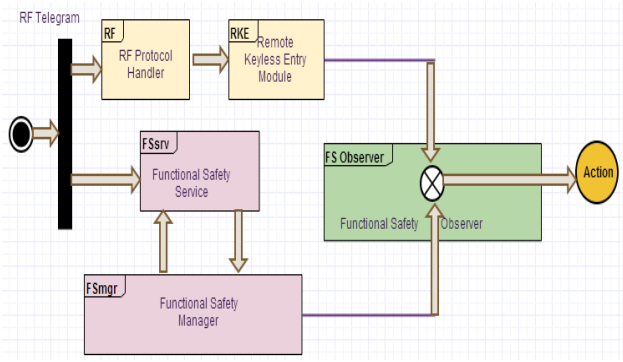
final step is the refinement of all concepts/idea/requirements stated in previous steps. Here it is necessary to describe through diagrams (e.g. UML diagrams) the functional redundancies and the information flow.

After the functional Safety Concept, the technical Safety Concept shall be defined. In the previous phase, the functional safety requirements were described and based on them through derivation the technical safety requirements are state. These requirements will described in which way the safety requirements shall be implemented to achieve the safety goals. If there is some safety requirements not allocated to some functional modules, then the allocation shall be made in this step. This implies at least a basic knowledge of the system even if the components should be treated as black boxes; the interfaces between these system components shall be specified. What can be observed here is the fact that decomposition is widely used. Also, a better understanding of the faults and consequences that can appear in the system is mandatory because the last step should be the definition of probability target values for the safety goal and for the involved elements.

IV. Implementation of an ASIL A system. Use case

In [9], authors described a Remote Keyless Entry System on an AUTOSAR system and in which way the SWC are developed as services and software components. That structure should be changed if some functional safety requirements are added. There are some cases when the lock button press shall be safety relevant. For example, in USA the double lock (which will close the car without opportunity to be opened from inside) is considered as functional safety. The double lock functionality suppose that two consecutive lock button presses on within a predefined time (e.g. 3 seconds) shall lead to the closing of the car without possibility to open the doors from inside. This can be very dangerous in case that somebody is at shopping and left someone else in the car, but pressing twice on the lock button and being very hot outside can cause for the person's inside the car to suffocate or any other injuries. From the things stated above, the functional safety requirement may be defined: to protect the system against undesired double lock actions means the rejection of the RF telegram in case that is not coming within the predefined time and not from a correct sequence. For this, the following architecture from Figure 3 shall be designed. We will assume that the cryptology protocol is a simple one in which just the telegram counter and the timestamp of the telegram are variable.

Figure 3. System architecture



The FSsrv(Functional Safety Service) Module is a module that replicates the behavior of the main path RF-RKE Application modules. It is performing the telegram identification, decryption, authentication and check also the integrity of the received telegram. At a higher level, the FSMgr (Functional Safety Manager) is a module that handles the process monitoring and means that checks if the first path outputs are the same with the second path outputs. In this case, the telegram is delivered to the RKE Application to be processed, otherwise it should be rejected. The FSObserver is the module that handles the evaluation of the process monitoring. Only if the outputs of the second path are the same with the output of the first one, the command for double lock shall be executed. This module acts as a watchdog application and if an error is detected in at least one path, then the current telegram processing shall be aborted and the telegram rejected. Also, it is highly recommended to take into consideration the followings:

a) The design of the SWC and entire architecture (or at least of the Functional safety relevant part) shall be a multilayer architecture. Using such architecture, the errors in the high layers are avoided because a high layer application is started only if an application from the next below layer is requesting the start of it. This save on one hand the CPU load and on the other hand saves the unwanted behaviors because a high layer application will process a request only if it was already initialized. Otherwise, the request will not be taken into consideration.

b) All states machine variables should be enumeration type which values were been generated using the extended Hamming distance algorithm.

c) 2oo3 majority is used for all critical variables and also for the state machine variables. Also, it should be stored in three non-consecutive RAM addressed.

The method described above represents a simple method to develop a safety system with the ASIL A fulfilled in context of AUTOSAR concept. Also the time needed for developing such modules is not drastically increased compared to the development of the non ASIL systems. (Only with Quality Management).

V. CONCLUSIONS

Standardization of the functional safety concept provides an easier and clearly method to develop a safe system. The entire development process is not so simple as it seems at first seen because all possible fault causes shall be taken into consideration, but a very analysis can leads to the elimination of all "hard" bugs from the software that can provokes damages or injuries to the end-users. Following the methodology for development such systems the accomplishment of the "safety" mission can be very easy. Also, because the AUTOSAR provides standard interfaces and modules to assure safety the mission becomes more and more easily. But in the end, the implementation of software to accomplish this seems to be the easiest step if the entire methodology was fulfilled. As a final conclusion, it is needed to be stated that the functional safety concepts needs a qualified person to do all needs steps and it is necessary to distinguish that even that safety and security goes hand in hand and refers to the same goal, but the concepts are different.

REFERENCES

[1] ISO 26262-1:2011 Road Vehicles – Functional Safety, ISO Standards Catalog

[2] Per Johannessen, OjvindHalonen, Ola Osmarck, Functional Safety Extensions to Automotive SPICE According to ISO 26262, Software Process Improvement and Capability Determination Communications in Computer and Information Science, Volume 155, pg 52-63, 2013

[3] The Code of Hammurabi: Introduction, The History Guide, 2009

[4] Paul Garden, Automotive Safety: Achieving ISO 26262 Compliance, DesignWare Technical Bulletin, 2013

[5] Stefan Bunzel, AUTOSAR architecture expands safety and security applications, 2011

[6] Simon Furst, BMW Group, Safetronic 2011, 8 Nov 2011

[7] YashSaini, Arun Jain, Safety & Security architecture for automotive ICs, 25 Sep, 2013

[8] National Instruments, What is the ISO 26262 Functional Safety Standard, 23 Feb 2012

[9] Catalin-Virgil Briciu, IoanFilip, Franz Heininger, A new trend in automotive software: AUTOSAR concept, Applied Computational Intelligence and Informatics (SACI), pg. 251 – 256, Timisoara, 2013