Functional Safety in Automotive Electronics

Zhang Xinbo*, Luo Feng

Clean Energy Automotive Engineering Centre
School of Automotive Studies
Tongji University, No. 4800 Cao'an road, 201804
Shanghai, China

***Abstract* -- With the increasing functionality and complexity of automotive electronics more risks of safety are introduced. It is necessary to perform the functional safety process throughout the safety lifecycle. In addition, the release of functional safety standard ISO 26262 also makes safety an important requirement of the design and implementation process for automotive electronics products. This paper presents a functional safety analysis approach in automotive electronics. It includes the whole product life cycle: concept, implementation, verification and validation, and all related aspects such as management, culture and product. With this approach automotive electronics can perform functional safety successfully.**

Keywords - Functional safety; ISO26262; Automotive Electronics; ASIL

I. INTRODUCTION AND OVERVIEW OF ISO 26262

Safety is one of the key issues of future automobile development. With more and more controllers are integrated into vehicle, development and integration of these ECUs will strengthen the need for safe system development processes and the need to provide evidence that all reasonable product safety objectives are satisfied. With the trend of increasing technological complexity, software and hardware implementation, there are increasing risks from systematic failures and random failures. ISO 26262[1] introduces guidance to avoid these risks by providing appropriate requirements and processes.

Functional safety is influenced by the development processes (including such activities as requirements specification, design, implementation, integration, verification, validation and configuration), the production and service processes and by the management processes.

ISO 26262 addresses the safety-related aspects of development activities and work products. ISO 26262 is based upon a V-model as a reference process model for the different phases of product development. Through the entire definition in whole development phase and all related aspects, the purpose is to decrease the safety risks, improve the reliability and quality [2].

Fig. 1 shows the overall structure of ISO 26262, that includes those parts: Part 1: Vocabulary; Part 2: Management of functional safety; Part 3: Concept phase; Part 4: Product development at the system level; Part 5: Product development at the hardware level; Part 6: Product development at the software level; Part 7: Production and operation; Part 8: Supporting processes; Part 9: Automotive Safety Integrity Level (ASIL)-oriented and safety-oriented analyses; Part 10: Guideline on ISO 26262.

II. FUNCTIONAL SAFETY MANAGEMENT

A. Content of Safety Management

In ISO26262 protocol safety management is defined in part 2 and part 8. There are three key points: Track all planning and organizational tasks which need to be performed during the safety life cycle. Provide evidence of the required safety. The responsible person of safety management is safety manager. The management of functional safety consists of three elements [3]: project independent safety management, safety management during development, safety management after start of production (SOP). Fig. 2 shows the relationship of these three elements.

B. Overall Safety Management of the company

Overall safety management of the company is independent of projects, is performed in the whole daily work. Overall safety management mainly includes following four points.

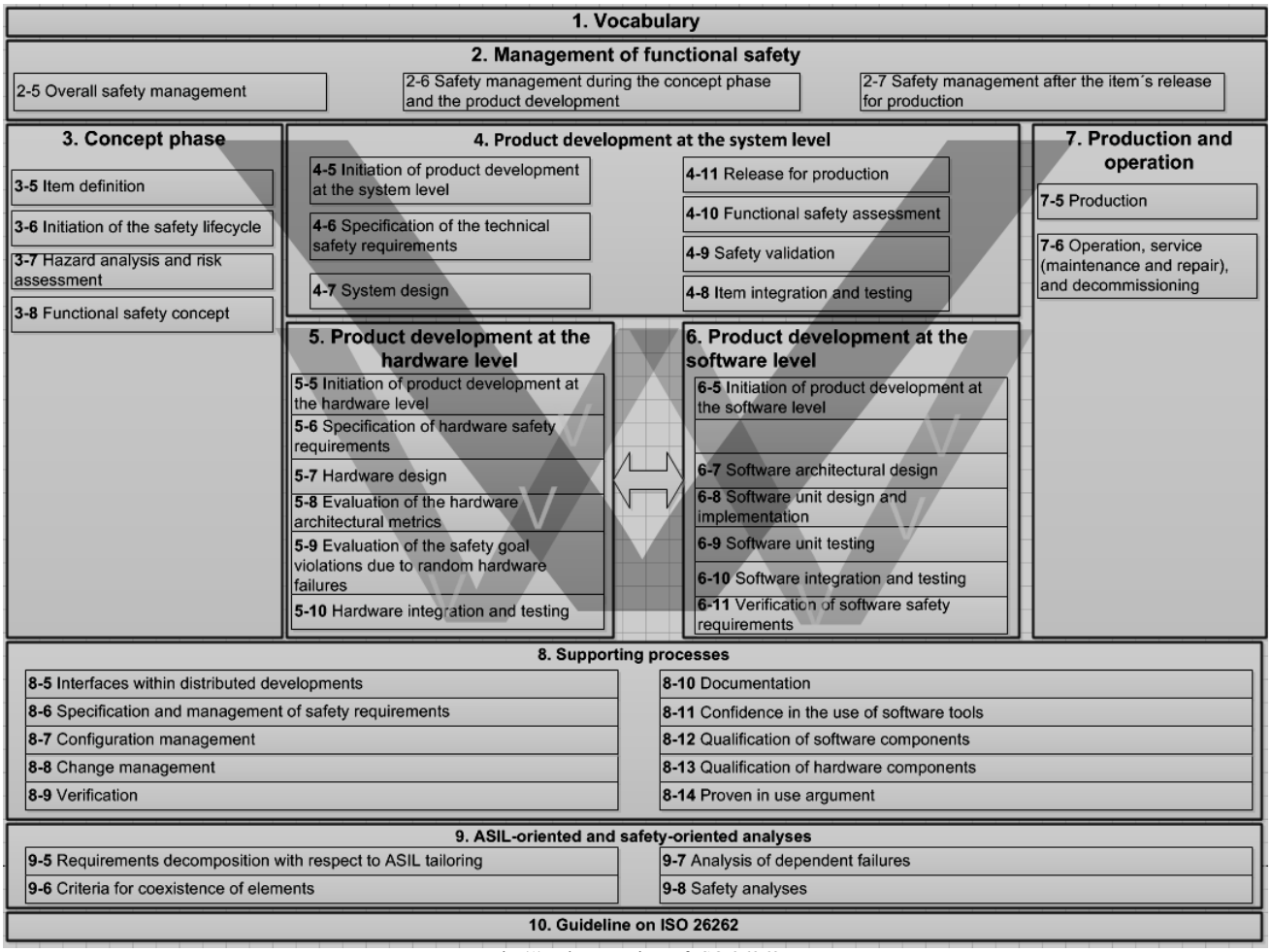

Fig.(1). The overview of ISO 26262

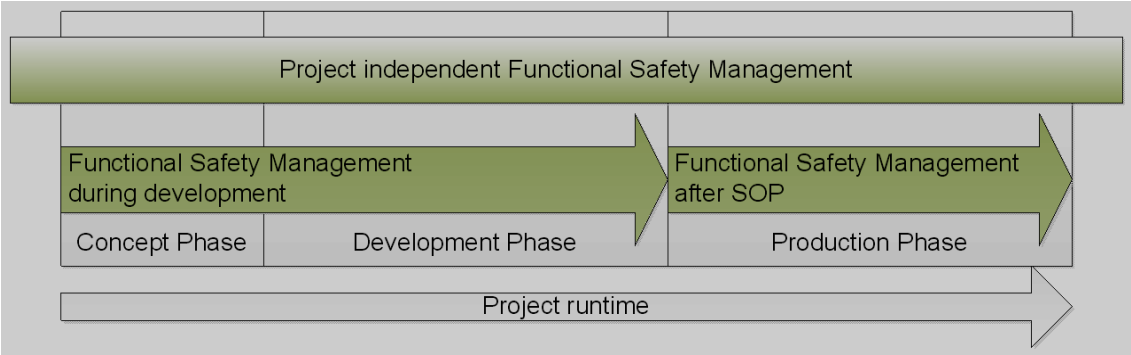

Fig.(2). Management of functional safety

Quality management system is implemented in the whole company such as CMMI, ISO TS16949, etc. Development process according to ISO 26262 is built. Employees must be qualified to perform functional safety; it can reach trough training and competence building about functional safety topic. Build safety organization and safety team to continuously improve functional safety activities.

C. Project Safety Management

During development phase, firstly we need a safety plan to describe all tasks which are needed to reach the functional safety targets and requirements. Meanwhile it plans safety development activities and responsibilities, also measures to evaluate and confirm functional safety – reviews, audits, assessments. Secondly safety cases

including the requirements, arguments and evidences must be built. Thirdly the release for production confirms that the item is ready for series-production and operation, release decision is based on safety case and safety assessment.

After product SOP, firstly still execute measures for maintaining functional safety throughout vehicle life (maintenance, inspection, repair etc.). Secondly the management of functional safety is continued (possibly by different persons). Thirdly monitor safety relevant failures on field. For each project safety manager must be nominated to plan, coordinate and track all safety activities.

III. CONCEPT AND SYSTEM DESIGN

A. The steps of concept and system design

Fig.3 shows the overview of the steps of concept and system design of functional safety.

To the step of item definition, the first objective is to define and describe the item, its dependencies on, and interaction with the environment and other items. The second objective is to support an adequate understanding of the item so that the activities in subsequent phases can be performed.

To the step of hazard analysis and risk assessment, the objective is to identify and to categorize the hazards that malfunctions in the item can trigger and to formulate the safety goals related to the prevention or mitigation of the hazardous events, in order to avoid unreasonable risk.

To the step of functional safety concept, the objective is to derive the functional safety requirements, from the safety goals, and to allocate them to the preliminary architectural elements of the item, or to external measures.

B. Item definition

To do item definition [4] [5], following questions need answer: What is my system (Extent, borders, interfaces)? What feature does my system have? Which elements / components does my system have? What are the interfaces between elements (and what is exchanged by these)?

An Item is a system which is able to completely execute one or more function(s). The Item contains all the elements that are necessary for the execution. The function is implemented within the Item through at least one of the following: element sensor, controller, or actuator (i.e. system definition of ISO26262). Usually, one Item encompasses one function group (e.g. engine control, lighting, restraint system etc.).

One vehicle has multiple Items. The environment, delimitation and content of the Item are documented in the Item definition. The Item definition defines the scope for that project which delivers the content of the Item.

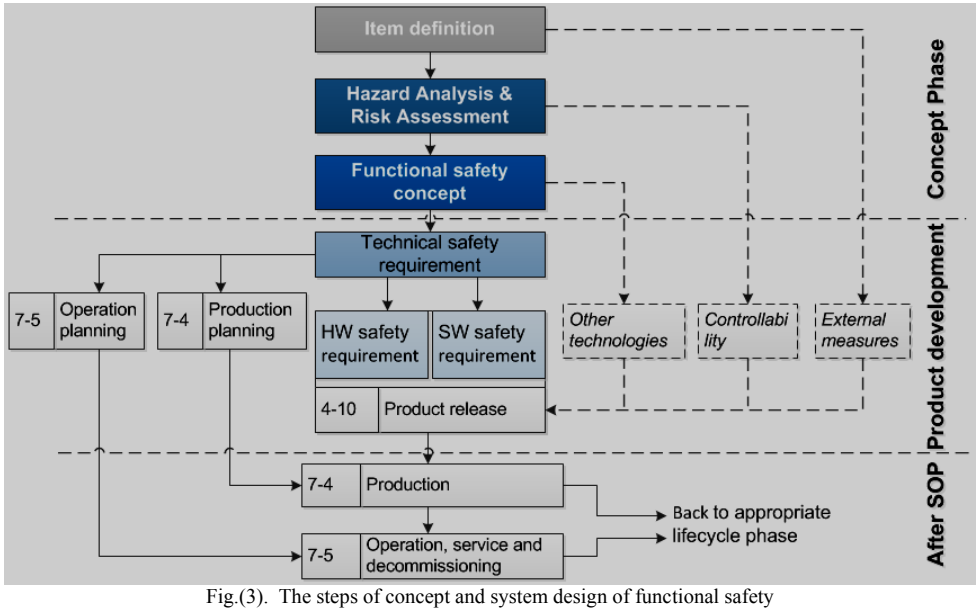

Fig.(3). The steps of concept and system design of functional safety

C. Hazard Analysis and Risk Assessment

Hazard analysis, risk assessment [6] and ASIL (Automotive Safety Integrity Level) [7] determination are used to determine the safety goals for the item such that an unreasonable risk is avoided. ASIL is a degree about the risk potential of a function in case of mal function. ASIL is the grade for implementation of processes and metrics. There are existing 4 ASIL classifications, A, B, C and D, ASIL A is the lowest ranking, ASIL D is the highest ranking classification.

For the further implementation of ISO 26262 the ASIL classification must be known. Following classifications are possible: QM (quality management) means no safety relevance according ISO26262 processes, according usual QM system (ISO/TS 16949 or ISO 9001 certification) are sufficient. ASIL A to ASILD means processes according to ISO26262.

ASIL classification need consider following issues: severity of hazard S, probability of exposure E, and controllability by driver or other road users C. Fig.4 shows the definition of these three elements.

After the estimation of severity, exposure and controllability, the ASIL classification can be calculated by following rule as figure 5.

D. Safety concept development

To comply with the safety goals, the functional safety concept contains safety measures, including the safety mechanisms, need to be implemented in the item's architectural elements and specified in the functional safety requirements [8].

1 — Class of severity categorisation

Class	S0	S1	S2	S3
Description	No injuries	light and moderate injuries	Severe injuries, possibly life-threatening, survival probable	Life-threatening injuries (survival uncertain) or fatal injuries
Reference for single injuries	AIS 0 and less than 10% probability of AIS 1-6 Damage that cannot be classified safety related	more than 10% probability of AIS 1-6 (and not S2 or S3)	more than 10% probability of AIS 3-6 (and not S3)	more than 10% probability of AIS 5 and 6

2 — Class of probability of exposure regarding duration/probability of exposure in driving situations

Class	E0	E1	E2	E3	E4
Description	Infeasible, Incredible	Very low probability	Low probability	Medium probability	High probability
Definition of duration/ probability of exposure		Not specified	< 1% of average operating time	1% - 10% of average operating time	> 10% of average operating time

3 — Classes of probability of exposure regarding frequency in driving situations

Class	E0	E1	E2	E3	E4
Description	Infeasible, Incredible	Very low probability	Low probability	Medium probability	High probability
Definition of frequency		Situations that occur **less often than once a year** for the great majority of drivers	Situations that occur **a few times a year** for the great majority of drivers	Situations that occur **once a month or more often** for an average driver	All situations that occur during **almost every drive on average**

4 — Examples of possibly controllable hazards by the driver or by the endangered individuals

Class	C0	C1	C2	C3
Description	Controllable in general	Simply controllable	Normally controllable	Difficult to control or uncontrollable
Definition	Distracting	99% or more of all drivers or other traffic participants are usually able to avoid harm.	90% or more of all drivers or other traffic participants are usually able to avoid harm.	Less than 90% of all drivers or other traffic participants are usually able, or barely able, to avoid harm.

Fig.(4). Definition of severity, exposure and controllability

ASIL determination

		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

Fig.(5). ASIL classification calculation

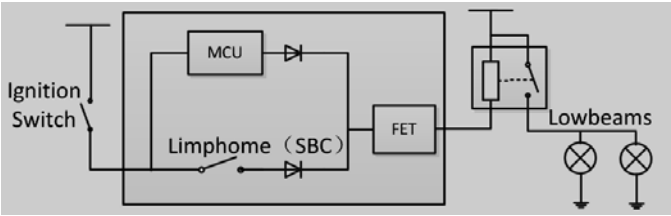

Fig.(6). Safety concept of low beam headlamp control

Take body controller computer (BCM) as an example, consider the low beam function of head lamp, the following functional safety concept should be designed. Headlamps shall be asserted when ignition switch is on and failsafe is asserted. There is no dependency on the low beam switch state in failsafe. MCU and failsafe control to low beam relay FET is wired-ORed. Further functional safety can be provided by checking low beam switch plausibility by checking for valid switch input pattern from master light switch. This requires an Off-position input. Logic is powered from ignition switch input. According to the functional safety concept, BCM block diagram for the low beam is designed as fig. 6.

IV. SOFTWARE AND HARDWARE DESIGN

A. Functional safety for hardware design

The safety requirements of the hardware considering the ASIL classification can be derived for example from the following sources:

- Technical safety concept
- System Design (Specification)
- Hardware development and verification plan
- SW Safety requirements.

For hardware development to fulfill functional safety requirement, a key point is that hardware failure rates must meet the target [9] [10]. The calculation method is:

Absolute measure: λ

Failure rate class 1 = Target ASIL D / 100 = 10-10

Failure rate class 2 = Failure rate class 1 * 10 = 10-9

Failure rate class 3 = Failure rate class 1 *100 = 10-8

Failure rate class i (i > 3) = failure rate class 1* 10(i-1)

Depending on the ASIL goal ① and on "how good" the detection mechanism is ②, the maximum possible failure rate ③ class is given by.

Fig.7 shows an example: ASIL B + Diagnostic Coverage of 90% => Maximum failure rate allowed is λ = Failure rate class 3 = 10-8.

B. Functional safety for software design

Each software safety requirement shall be assigned an ASIL. Software safety requirements shall be specified by: Natural language, semi-formal notations such as system analysis and design techniques (SADT) and unified modeling language (UML), or Formal notations e.g. prototype verification system (PVS). For software functional safety goal, ASIL decomposition can be applied at any design level of product development. ASIL decomposition helps in unnecessary rising of ASILs, higher safety goal can be implemented by two lower safety level modules [11]. Fig.8 shows the basic principle.

②

① ASIL of the Safety Goal	Diagnositic Coverage with respect to residual faults			
	>=99.9%	>= 99%	>= 90%	< 90%
D	Failure rate class 4	Failure rate class 3	Failure rate class 2	Failure rate class 1 + dedicated
C	Failure rate class 5	Failure rate class 4	Failure rate class 3	Failure rate class 2 + dedicated
B	Failure rate class 5	Failure rate class 4	Failure rate class 3	Failure rate class 2

③

Fig.(7). Effect of ASIL on the Failure Rate Class

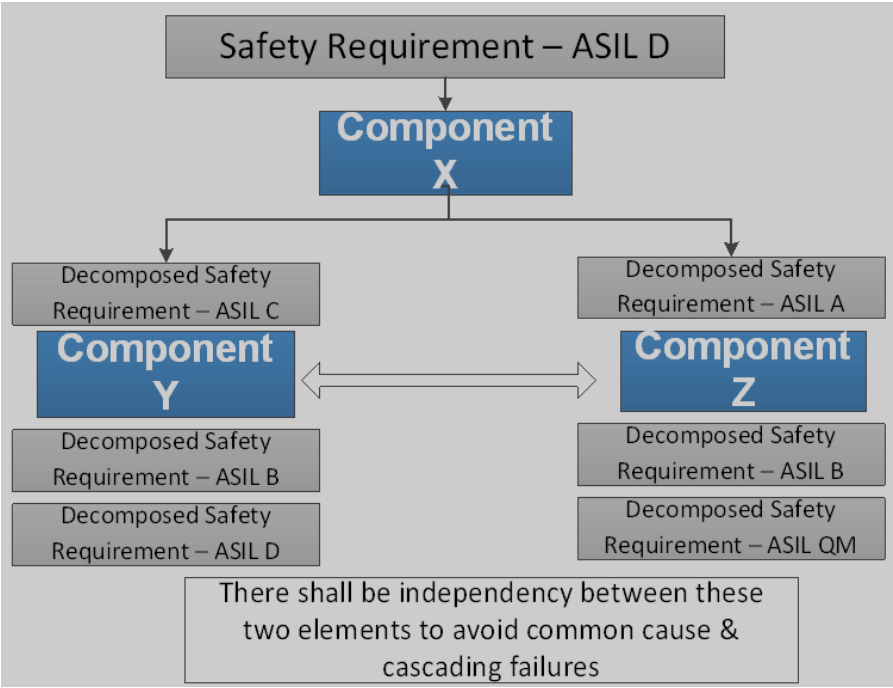

Fig.(8). Basic principle for ASIL decomposition

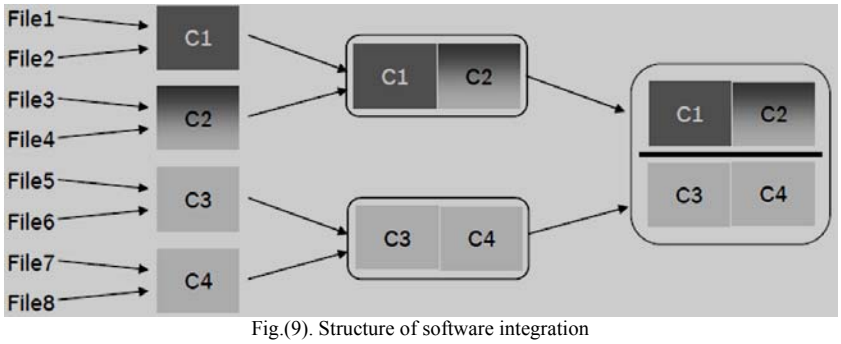

Fig.(9). Structure of software integration

Software unit design shall exhibit properties such as:

-Correct order of execution of programs

-Consistency of interfaces between software units

-Correctness of data flow between and within software units

-Correctness of control flow between and within software units

-Simplicity

-Readability and comprehensibility

-Robustness

-Suitability for modification

-Testability

Software integration steps shall be as per the hierarchical structure of the software architecture, Fig.9 shows the structure.

V. VERIFICATION AND VALIDATION

A. Design review for functional safety

To verify the design complain with functional safety target, the designed must be reviewed according to ISO26262 rule [12][13].

Verification has following eight methods. Walk-through is a less stringent peer review. Inspection is stringent peer review, a formal procedure including a previously defined procedure, checklist, moderator and review of the results. Semi-formal is based on a description given in semi-formal notation. Use of test vectors generated from a semi-formal model to test that the system behavior matches the model. Formal verification is a method used to prove the correctness of a system against the specification in formal notation of its required behavior, model checking, theorem proving etc. Control flow analysis can find possible errors, redundancy in programs and dead code without executing the program. Data flow analysis is to check that the information exchanged between SW units is applied semantically and syntactically correctly. Static code

analysis statically analyzes the code for errors. Semantic code analysis analyzes the semantics of the code like type checking. Fig.10 shows the rule of how to verify a software module.

Methods for the verification of software unit design & implementation - recommendations

Topics		ASIL			
		A	B	C	D
1a	Walk-through	++	+	o	o
1b	Inspection	+	++	++	++
1c	Semi-formal verification	+	+	++	++
1d	Formal Verification	o	o	+	+
1e	Control Flow Analysis	+	+	++	++
1f	Data Flow Analysis	+	+	++	++
1g	Static Code Analysis	+	++	++	++
1h	Semantic code analysis	+	+	+	+
++: stongly recommend, +:recommend, o: not necessary					

Fig.(10). Verification rule of software module

B. Testing for functional safety

Functional safety test includes unit test and integration test. Unit test has following methods. Requirements based tests should have at least one test case per one requirement. Interface tests check interfaces between units within a component. Fault injection tests inject a known fault during compilation or run time to increase coverage of test cases. Resource usage tests check amount of ROM, RAM Consumption, runtime etc. In back to back comparison tests between model and code, the model and code are stimulated in the same way and results compared with each other. Fig.11 shows the test concept of unit test.

Integration levels and interfaces between the elements shall be tested to verify whether the software architecture is properly realized by elements. Integration test has similar methods like unit test but the test contents are different. Requirements based tests should design at least one test case per one requirement at the architecture level. Interface tests take care of interfaces between components. Fault injection tests inject arbitrary faults in order to test safety mechanisms (e.g. by corrupting software or hardware components). Resource usage tests check the properties such as average and maximum processor performance, minimum or maximum execution times, storage usage (e.g. RAM for stack and heap, ROM for program and data) and the bandwidth of communication links (e.g. data buses) have to be determined. In back to back comparison tests between model and code, the model which simulates components and code are stimulated in the same way and results compared with each other. Fig.12 shows integration test.

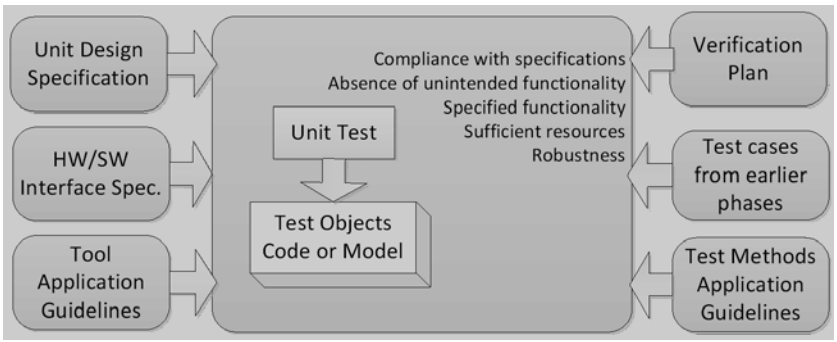

Fig.(11). Unit test of functional safety

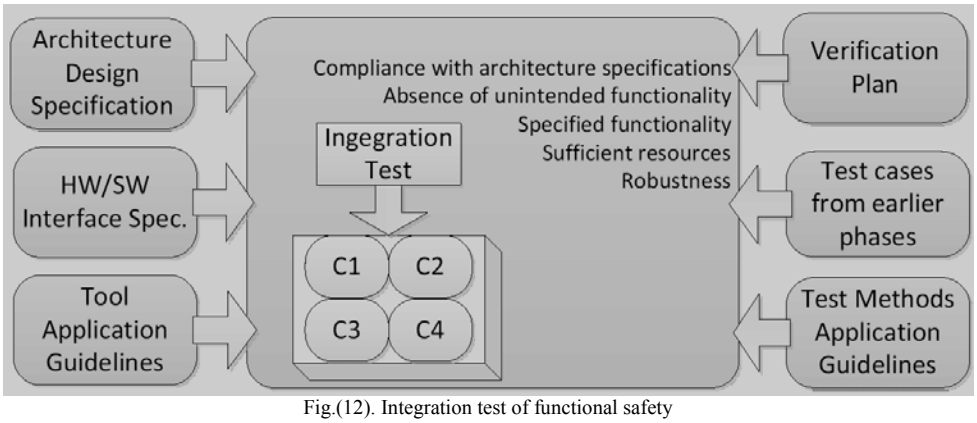

Fig.(12). Integration test of functional safety

VI. CONCLUSION

To implement functional safety according to ISO 26262, first of all a team should be built and particular functional safety target and requirement should be defined. In development stage, special methods for requirement management, concept design, system design, hardware and software design. The verification and validation for functional safety design should also use sufficient methods. With the methods introduced in this paper, functional safety goal in automotive electronics field can be reached successfully.

REFERENCES

[1] INTERNATIONALSTANDARD. ISO 26262. 2011.
[2] Graphically Notated Fault Modeling and Safety Analysis in the Context of Electric and Electronic Architecture Development and Functional Safety. Nico Adler. 978-1-4673-2789-3/12. IEEE. Pages 36-42. 2012.
[3] Functional Safety and System Security in Automation Systems – A Life Cycle Model. Thomas Novak. 1-4244-1506-3/08 IEEE. Pages 311-318. 2008
[4]Influence between Functional Safety and Security. Dong-bo Pan. Second IEEE Conference on Industrial Electronics and Applications. Pages 1323-1325. 2007.
[5] The TUV Approach To Functional Safety Assessment and Certification. Heinz Gall. The Institution of Electrical Engineers. Pages 29-44. 2004.
[6] Functional Safety Problems in the Ubiquitous Environment. Lei Jing. 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07). IEEE. 2007.
[7] A Functional Safety Analysis Approach for Analyzing CBTC System. Xianqiong Zhao, International Conference on Measuring Technology and Mechatronics Automation. Pages 737-741. 2009.
[8] Dependability and Functional Safety (Embedded Real-time Control Operating System). Giuseppe Buja, IEEE industrial electronics magazine Sep. 2012.
[9] Failure Rate Calculation with Priority FTA Method for Functional Safety of Complex Automotive Subsystems. Masahiko Takeichi. 978-1-4577-1232-6/11 IEEE. 2011.
[10] Functional Safety Aspects of Pattern Detection Algorithms. Joachim Iden. 8th IEEE International Conference on Automation Science and Engineering. Pages 747-752. Aug. 2012.
[11] Common Approach to Functional Safety and System Security in Building Automation and Control Systems. Thomas Novak. 1-4244-0826-1/07. IEEE. Pages 1141-1148. 2007.
[12] FUNCTIONAL SAFETY IN APPLICATION OF PROGRAMMABLE DEVICES IN POWER SYSTEM PROTECTION AND AUTOMATION. S. Purewal. Health & Safety Executive, UK. Pages 295-298.2004.
[13] Validation, Verification and Immunity Testing Techniques for EMC for Functional Safety. EurIng Keith Armstrong. 1-4244-1350-8/07/. IEEE. 2007.