


⬆ (/) / Component & Materials ([articleWebList1.asp?c\\_no1=37](articleWebList1.asp?c_no1=37))

## ISO 26262

### 해외 가이드라인 쉽게 이해하기

2015년 03월호 지면기사 / 글 | 채 승 업 \_ [sychae@ssu.ac.kr](mailto:sychae@ssu.ac.kr)

[om/sharer.php?u=http://autoelectronics.co.kr/article/articleView.asp%3Fidx=1666&t=ISO%2026262](http://om/sharer.php?u=http://autoelectronics.co.kr/article/articleView.asp%3Fidx=1666&t=ISO%2026262))  [v/s/share?url=http%3A%2F%2Fautoelectronics.co.kr%2Farticle%2FarticleView.asp%3Fidx%3D1666&text=ISO%2026262%20-%20%EC%9D%B4%ED%95%B4%ED%95%98%EA%B8%B0&kakao\\_agent=sdk%2F1.37.2%20os%2Fjavascript%20lang%2Fko-KR%20device%2FWin32%20origin%2Fhttp%253A%252F%252Fwww.autoelectronics.co.kr](http://s/share?url=http%3A%2F%2Fautoelectronics.co.kr%2Farticle%2FarticleView.asp%3Fidx%3D1666&text=ISO%2026262%20-%20%EC%9D%B4%ED%95%B4%ED%95%98%EA%B8%B0&kakao_agent=sdk%2F1.37.2%20os%2Fjavascript%20lang%2Fko-KR%20device%2FWin32%20origin%2Fhttp%253A%252F%252Fwww.autoelectronics.co.kr)



채승업 씨가 'ISO 26262 해외 가이드라인 쉽게 이해하기'를 5회에 걸쳐 연재한다. ISO 26262 대응을 위해 일본 JASPAR는 2014년 2월 24일 ISO 26262 가이드라인 영문판을 공개했다. 또 유럽은 SAFE가 ISO 26262 가이드라인을 공개하고 있다.

1. ISO 26262 해외 가이드라인 소개
2. ISO 26262 Part3 가이드라인 - Concept
3. ISO 26262 Part4 가이드라인 - System
4. ISO 26262 Part5 가이드라인 - HW
5. ISO 26262 Part6 가이드라인 - SW

### ECU 개발과 규제에 따른 SW 비용 증가

전장부품인 ECU의 개발에 있어 SW가 차지하는 비용은 매년 증가하고 있고, SW 기술 확보는 전장부품회사의 매출액에 큰 영향을 주고 있다. 따라서 관건은 SW 개발 비용을 어떻게 절감할 것인가가 되고 있다(그림 1). 세계 1위 전장부품 회사 보쉬는 2012년 ECU 개발에 있어 SW 개발 비용의 핵심요소를 다음과 같이 정의하며 SW 기술 확보 전략을 수립했다(그림 2).

① Bosch specific SW: 보쉬에 특화된 SW 기능.

② Bosch Platform SW: 보쉬의 AUTOSAR Platform (개발 비용을 60% → 40% 하향)

- 보쉬는 AUTOSAR Platform의 SW 개발 비용을 절감하기 위해 2013년 6월 AUTOSAR 4.x BSW를 Open source로 COMASSO (<http://www.comasso.org>) 단체를 만들었다. AUTOSAR 회원사이면 첫째 500유로(75만 원), 그 후 매년 2,500유로(350만 원)에 AUTOSAR BSW를 사용할 수 있다. 현재 17개 회사가 가입돼 있다.

③ SW sharing: 타 OEM/Tier-x의 SW 도입(채용 목표가 향후 20% → 40% 상향)

- AUTOSAR 기반의 SW-C 재사용 개념을 도입해 OEM/Tier-x에서 개발한 SW-C를 재사용 SW 개발 비용을 절감, Tier1이 단독으로 SW를 개발하는 비중이 감소할 것이고, ECU 통합 및 IT 기술 융합이 가속화될 것이다.

보쉬는 Tier1은 SW 기술 활용능력으로 시스템 통합, 서플라이어 관리, 기능 위주의 해당 도메인 장점 확보 강화에 AUTOSAR가 필요하다고 보고 있다. OEM은 SW 개발 비용 절감을 위한 조직적인 투자를 어떻게 할 것인지, Tier1은 여러 OEM에 대응하기 위한 다양한 차종 전개에 대응해 어떻게 조직을 키울 것인지, legacy code와 Architecture를 어떻게 조직에 반영할 것인가를 고민하고 있다.

([http://ssl.logger.co.kr/tracker\\_ad.tsp?u=37061&mode=C&adCode=57236](http://ssl.logger.co.kr/tracker_ad.tsp?u=37061&mode=C&adCode=57236))

([http://ssl.logger.co.kr/tracker\\_ad.tsp?u=37061&mode=C&adCode=77860](http://ssl.logger.co.kr/tracker_ad.tsp?u=37061&mode=C&adCode=77860))

### 과월호 e-Book 보기 (</ebook/list.asp>)



(</ebook/list.asp>)(</ebook/list.asp>)

### News & Analysis

2륜차 리어램프에 최적! 4ch 리니어 LED 드라이버  
(</article/articleView.asp?idx=3473>)

다쏘시스템, 조메트리와 파트너십 체결...  
(</article/articleView.asp?idx=3472>)

포레스터 리서치, 산업용 IoT SW 플랫폼 리더로 마인드스피어 선정  
(</article/articleView.asp?idx=3471>)

[보도자료]ST, IIoT와 자동차 애플리케이션을 위한 안전한 셀룰러 연결 제공  
(</article/articleView.asp?idx=3470>)

뤼츠 시스템 솔루션즈, 자동차 이더넷 테스트(ATE) 발표  
(</article/articleView.asp?idx=3469>)

(<https://www.drivingthenation.com>)

([http://smartn.co.kr/book/book\\_detail.asp?p\\_no=B00159](http://smartn.co.kr/book/book_detail.asp?p_no=B00159))

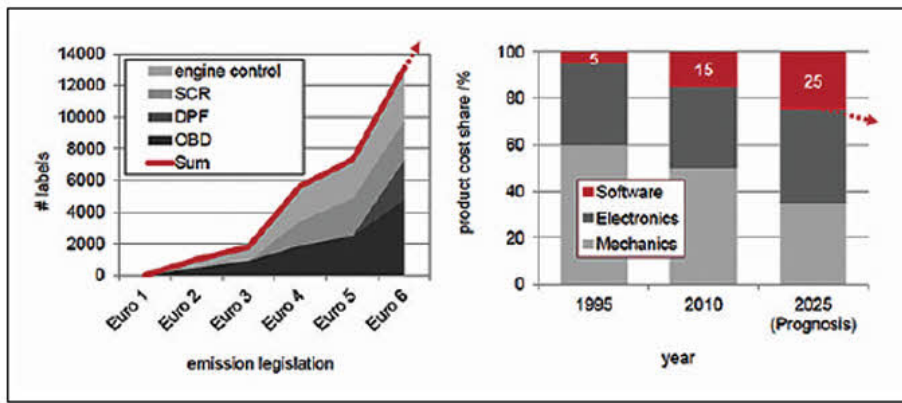


그림 1 | Euro 규제 및 SW 개발 비용 증가

출처 | An AUTOSAR-based powertrain control SW product line, FEV, 2014.10

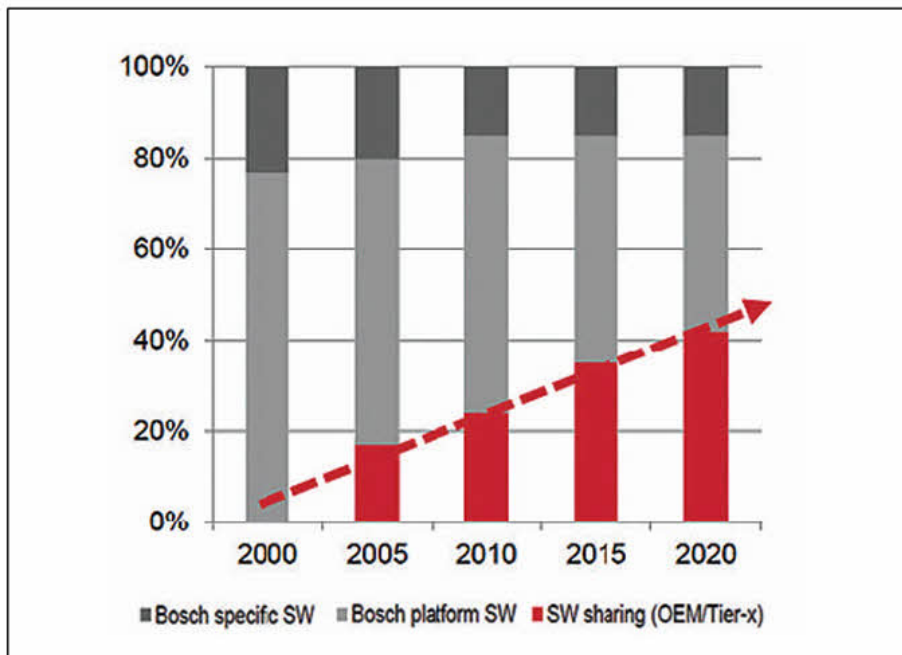


그림 2 | 보쉬의 ECU 개발과 SW 개발 비용 분석

출처 | Elektroniksysteme im Automobil, München 12.-14-02.2012, Michael Walther, Robert Bosch GmbH

#### EMC 영향에 따른 MCU 특징 고려한 SW 개발

ISO 26262 적용이 의무화되면서 차량 반도체 회사들이 EMC 영향을 고려한 Safety Mechanism 기술을 반영하기 시작했다. 이에 따라 ISO 26262 적용 전 500페이지 정도의 하드웨어 매뉴얼이 최소 5배 이상인 2,500페이지 이상으로 늘어났다.

예를 들어, 그림 3과 같이 LED를 켜기 위한 SW 개발은 ISO 26262 적용 전에는 단순한 1과 0에 대한 값에 해당하는 GPIO의 register에 1의 값을 설정함으로써 됐지만, ISO 26262 적용 후에는 EMC의 영향을 고려해 LED 켜기가 제대로 수행됐는지에 대한 모니터링을 위한 HW register 설정이 추가됐다(그림 3의 ①).

EMC 영향으로 LED 켜기가 실패했을 경우 MCU의 SMU(Safety Mechanism Unit)의 설정 정보에 따라 향후 어떤 시나리오를 적용할 것인지를 판단하는 처리 로직을 개발해야 하게 됐다(그림 3의 ②). 즉, ISO 26262가 적용되면서 EMC 영향에 의한 오동작(Malfunction)을 방지하기 위해 MCU에서 Safety Mechanism을 제공하게 됐고, SW 개발자는 MCU에서 제공하는 Safety Mechanism을 이해하고 SW를 개발해야만 하게 됐다.

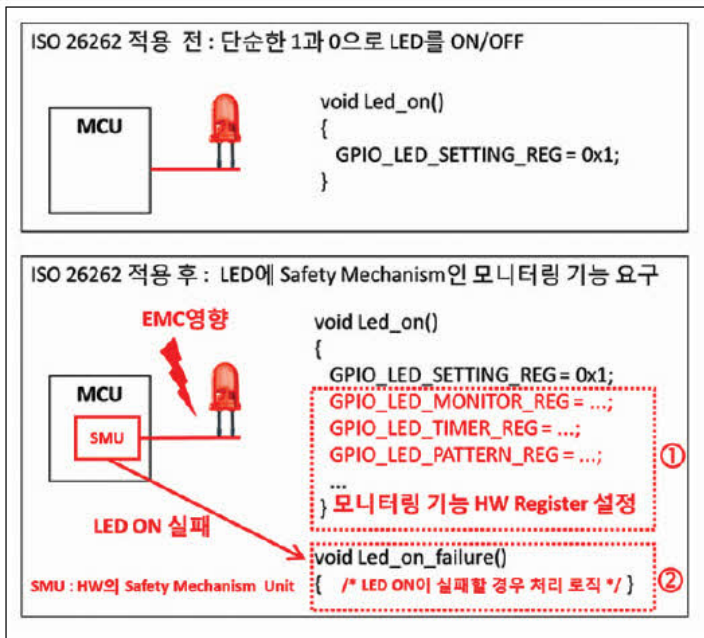


그림 3 | ISO 26262 적용 전과 후의 차이(LED ON 예제)

이런 이유로 ISO 26262를 적용한 전장부품 ECU를 개발하기 위해서는 경력사원이 필요하게 됐고, 최소 2,500페이지가 넘는 하드웨어 매뉴얼 숙지가 요구되고 있다. 한편 ISO 26262를 지원하는 MCU가 적용된 ECU가 양산되면, 하드웨어 매뉴얼의 복잡도로 인해 다른 회사의 MCU로 바꾸기가 힘들어진다.

#### Part6, Product development at the Software level

ISO 26262의 Part6은 다음과 같은 프로세스를 가지고 있다.

특이사항은 Part6-7 소프트웨어 아키텍처 설계에서 SW-C의 재사용을 고려해야하는 것이다. 여기서 SW-C의 재사용이 무엇인지 설명할 필요가 있다. ISO 26262에서 SW-C의 개념은 "1개 혹은 그 이상의 SW Units"로 표현돼 있어 개념 이해가 어렵지만, AUTOSAR에서는 1개의 SW-C는 개발자가 직접 코딩(hand Coding)하는 1개의 C 파일이라고 정의돼 있다.

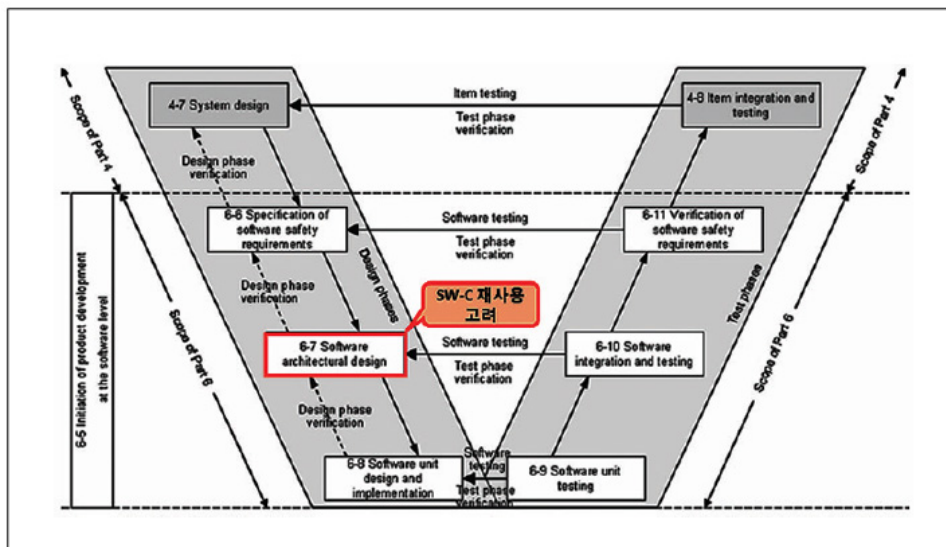


그림 4 | ISO 26262 Part6의 소프트웨어 양산 개발 절차(ISO 26262 Part6, Figure2)

SW-C의 재사용은 AUTOSAR Rte API를 사용함으로써 1개의 SW-C가 소스 코드의 수정 없이 재컴파일로 다른 ECU에서도 사용될 수 있음을 의미한다(그림 5). 즉, AUTOSAR를 적용하지 않으면 SW-C의 재사용을 보장하기 힘들다. 또 SW-C가 재사용되려면, 이전에 AUTOSAR로 개발한 SW-C가 존재해야하는 것이다.

ISO 26262를 준수한 소프트웨어 개발은 표 1과 같이 SW 개발자에게 개발 능력을 요구한다.

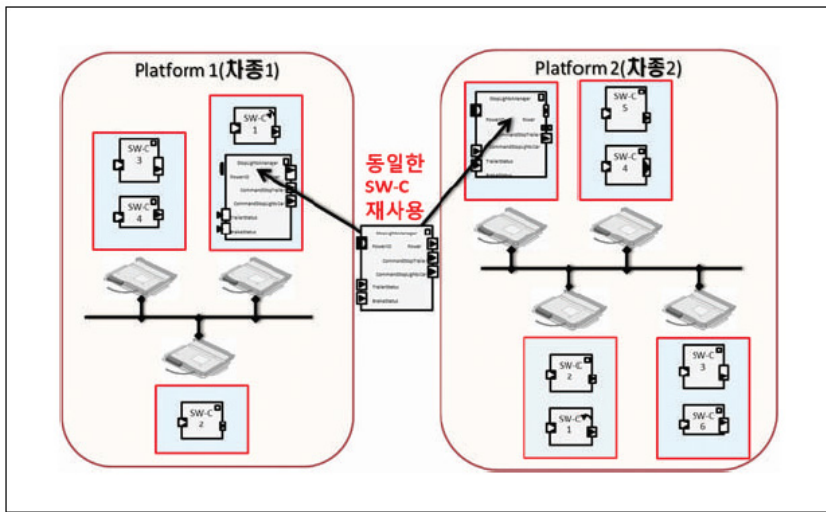


그림 5 | 서로 다른 차종에 동일한 SW-C의 재사용 개념도

① 6-5 Initiation of product development at the hardware level:

Part4-5에서 실시한 Project/Safety/ Item integration and testing plan의 업데이트된 내용을 고려해 소프트웨어 일정을 다시 고려해서 Part6-5.5 Safety plan을 업데이트한다. 또한 Part6 소프트웨어는 Part5와 다르게 6-5.5.2 Software verification plan으로 소프트웨어 검증 계획을 수행해야 한다. 특히 AUTOSAR 기반의 ECU를 개발할 경우는 "Part6-Annex C Software Configuration"을 고려해 수행해야 한다.

표 1 | 자동차의 전장부품인 ECU 개발에 있어서 요구되는 SW 개발 능력

기술 수준	SW 개발 능력 항목
Level 1	버그 없는 SW
Level 2	최적화된 SW: ROM/RAM 사용량 최소, API 실행 속도 향상
Level 3	신뢰성 높은 SW: MISRA C 준수, MC/DC 커버리지 달성
Level 4	가용성 높은 SW: HIS Source Code Metrics 준수자료: <a href="http://portal.automotive-his.de/images/pdf/SoftwareTest/his-sc-metriken.1.3.1_e.pdf">http://portal.automotive-his.de/images/pdf/SoftwareTest/his-sc-metriken.1.3.1_e.pdf</a>
Level 5	재사용 되는 AUTOSAR 기반의 SW-C
Level 6	안전 목표를 달성하기 위한 Safety Mechanism이 적용된 SW-C

이와 관련해 해외 OEM은 AUTOSAR 기반에서의 진단(Diagnostic)과 차량 네트워크의 Software verification을 위한 OEM-Tier1-Tier2 간 Software Configuration의 인하우스(사내용) 검증 도구가 구축돼 있다. OEM이 AUTOSAR 기반의 Software Configuration을 이용한 Software verification을 구축하려면 최소 3년 이상의 연구개발이 필요하다.

또 "6-5.5.3 Design and coding guidelines for modeling and programming languages(Simulink 등을 이용한 Model based Development 개발 과정, AUTOSAR BSW 적용 방법, MISRA-C 등)"와 "6-5.5.4 Tool application guidelines(Traceability 도구, 정정 분석 도구, AUTOSAR Vendor 등)"에 대해서 소프트웨어 개발에 필요한 가이드라인을 OEM과 서플라이어가 8-5.5.2 DIA(Development Interface Agreement)에서 협의한 대로 가이드라인을 채택해야 한다.

표 2 | EPB(전자식 주차 브레이크)에서 SW-C의 예

NO	이름	SW-C의 개요	분류
EPB-SWE-001	장력 센서값 검출	장력 센서 출력의 전압값을 AD 변환 센서 값을 검출	신규개발
EPB-SWE-002	차량 정보 수신	CAN 통신에 의한 차량 정보 (차속, 주차 브레이크 스위치)를 취득	재사용
EPB-SWE-003	구동 출력	제어량을 모터 제어 값으로 변환 출력	수정

일반적으로 수주할 OEM에서 정한 소프트웨어 개발에 대한 가이드라인(1,000페이지 이상)을 따라야 하기 때문에, 서플라이어는 최신 소프트웨어 개발 방법에 대한 가이드라인을 내부적으로 가지고 있어야 하며, 수주할 OEM과 기술적인 차이가 없도록 항상 내부 가이드라인의 내용을 최신 내용으로 업데이트해야 한다. 이와 함께 소프트웨어 관련 가이드라인을 이해할 수 있도록 내부 소프트웨어 개발자의 역량을 키워야 한다.

우리나라 서플라이어들이 가장 어려워하는 점은 해외 OEM의 “6-5.5.3 Design and coding guidelines for modeling and programming languages”에 대한 내용에 기술돼 있는 AUTOSAR, ISO 26262 등의 가이드라인 내용이다. 이 때문에 개발 지연이 발생하고 있다. 즉, 서플라이어가 소프트웨어 가이드라인을 이해 못하면 OEM에게 요청할 내용이 무엇인지, 서플라이어가 산출물로 OEM에게 전달해야 하는 것이 무엇인지 파악하지 못해 개발이 지연되고 있다.

#### ② 6-6 Specification of software safety requirements:

Part4-7.5.1 TSC(Technical Safety Concept), Part4-7.5.2 SDS(System Design Specification), Part4-7.5.3 HSI(Hardware software interface specification)을 고려해 Part6-6.5.1 SWSR(Software safety requirements specification)을 작성하고, Part6-6.5.2 HSI(Hardware software interface specification)을 업데이트한다. 또한 Part6-6.5.3 Software verification plan 을 업데이트하고, Part5-6.5.3 Software verification report(Part6-6.5.1 SWSR에 대해서)를 작성한다.

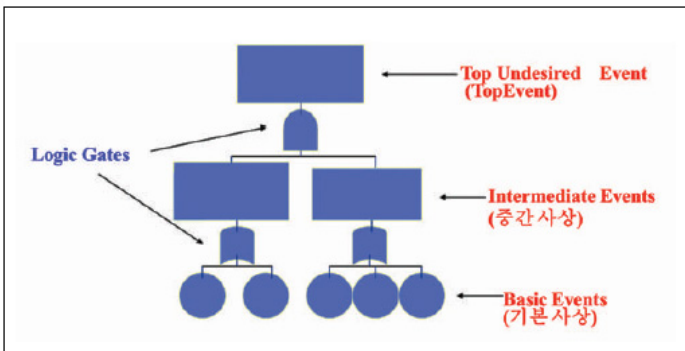


그림 6 | FTA의 구조에서 용어 정의

#### ③ 6-7 Software architectural design:

Part6-5.5.3 Design and coding guidelines for modeling and programming languages, Part6-6.5.1 SWSR(Software safety requirements), Part6-6.5.2 HSI(Hardware software interface specification)을 고려해 Part6-7.5.1 SWADS(Software architectural design specification)을 작성하는데, 수주한 OEM에서 협의한 SW 개발 절차 가이드라인 Part6-5.5.3을 준수해야 한다.



표 3 | JASPAR의 전자식 주차 브레이크 SW FMEA 작성 예

No.	Software component		Failure mode	Cause	Possibility of safety goal violation		
	Main category	Sub category			EPB-SG-001	EPB-SG-002	EPB-SG-002
EPB-SWE-003	구동 출력	모터 제어값의 계산	모터 제어값이 의도한 값보다 작음	...		1	
			모터 제어값에 Soft error 발생	모터 제어값의 RAM 고장	1		
Software			Hardware		Validation	Remarks	
SW diagnosis method	SW measure method	HW diagnosis method	HW measure method				
...	...	...	...	OK			
EPB-SSR-001(차속이상 검출)	EPB-SSR-009(Lock 해제 처리)	...	...	OK			

Part6-7.5.4 Safety analysis report(SW FMEA/FTA 실시 결과)를 작성해 해당 안전목표(Safety Goal)를 달성하기 위해 어떤 Safety Mechanism을 선정해 어떻게 적용했는지를 작성한다. Part6-7.5.5 Dependent failures analysis report는 Part9-7 Analysis of dependent failures를 준수해 Part6-7.5.4 Safety analysis report에 cascading failure(다른 Element의 failure가 전파돼 해당 Element를 failure가 발생됨)와 common cause failure(공동 원인으로 인해 서로 다른 Element의 Failure가 발생됨)을 고려해 서로 다른 SW-C 간(서로 다른 ASIL의 SW-C 간, 안전/비안전 SW-C 간)에 간섭 없이(freedom from interference), 혹은 충분히 독립(sufficient Independence)돼 SWSR을 구현했음을 증명할 수 있도록 근거를 작성한다.

또 Part6-7.5.2 Safety Plan, Part6-7.5.3 SWRS (Software safety requirements) specification, Part6-7.5.6 Software verification report(Part6-7.5.1 SWADS에 대해서)를 업데이트 한다.

Part6-7 Software architectural design에서 Part5-7 Hardware design과 비교해 큰 차이는 해외 OEM의 SW 개발 절차 가이드라인 Part6-5.5.3 Design and coding guidelines for modeling and programming languages를 준수해야 하고, Part6-7.5.5 Dependent failures analysis report를 작성하는 점이다.

#### ④ 6-8 Software unit design and implementation:

Part6-5.5.3 Design and coding guidelines for modeling and programming languages, Part6-7.5.3 SWRS(Software safety requirements) specification를 고려해 Part6-8.5.1 SWUDS(Software unit design specification)과 Part6-8.5.2 Software unit implementation을 작성한다. Software unit은 1개의 SW-C 파일에서 구현된 함수(API)로 생각하면 된다. Part6-8.5.3 Software verification report(Part6-8.5.1 SWUDS에 대해서)를 업데이트를 한다.

#### ⑤ 6-9 Software unit testing:

Part6-9.5.1 Software verification plan을 업데이트하고, Part6-6.5.2 HSI(Hardware software interface specification), Part6-8.5.1 SWUDS(Software unit design specification), Part6-8.5.2 Software unit implementation을 고려해 Part6-9.5.2 Software verification specification(Part6-9 Software unit testing에 대해서)를 작성한다. Part6-9.5.3 Software verification report(Part6-9 Software unit testing에 대해서)를 업데이트한다.

#### ⑥ 6-10 Software integration and testing:

Part6-10.5.1 Software verification plan(Part6-10 Software integration and testing에 대해서)을 업데이트하고, Part6-6.5.2 Hardware-software interface, Part6-7.5.1 Software architectural design, Part6-8.5.2 Software unit implementation을 고려해, Part6-10.5.2 Software verification specification(Part6-10 Software integration and testing에 대해서)을 업데이트한다. 또 양산 보드에 탑재할 수 있는 바이너리 형태의 통합 테스트를 할 수 있는 Part6-10.5.3 Embedded Software를 만든다.

AUTOSAR에서 통합 테스트를 하기 위해서 Part6-10.5.3 Embedded Software는 Postbuild 형태로 진단(Diagnostic)이나 차량 네트워크를 바이너리로 만들어 Target Board에 올린 상태로, 통신 정보나 진단 관련 값을 ECU configuration ARXML을 변경해 가면서 통합 테스트를 한다. Part6-10.5.4 Software verification report(Part6-10 Software integration and testing 에 대해서)를 업데이트 한다.

BMW	Fortiss, Promosoma	BMW and Fortiss	(implements)
No.			
1	Fault Avoidance	Freedom from interference	Partitioning
2		Replication	
3		Barrier	Interlock
4			Data Interlock
5	Error Detection	Stateless Error Detection	Checksum
6			Control Flow Interlock
7			Parity Checker
8			CRC
9		Comparison	Software Self Test
10		Self Test	Hardware Self Test
11			RAM Self Test
12			ROM Self Test
13		Range Check	
14		Challenge Response Check	
15		Message Readback Check	
16		Stateful Error Detection	Plausibility
17			Analytic Redundancy
18			Autosar Monitoring
19			Sensor Plausibility
20			Sensor Correlation
21			Sensor Rationality Check
22			Gradient Checker
23		Logical Monitoring	Logical Control Flow Monitoring
24			Logical Data Flow Monitoring
25			Data Sequence Monitor
26		Temporal Monitoring	Temporal Control Flow Monitoring
27			Deadline Supervision
28			Alive Supervision
29			Data Timeout Monitor
30			Maximum Age
31	Error Handling	Masking	Error Filtering
32			Default Value
33			Voting
34			1-out Voter
35			2-out Voter
36		Error Correction	Error Correction Code
37			Hamming Code
38			Reed Solomon Code
39			Convolutional Code
40		Reporting	(CHROMOSOME Health Monitor)
41			(Autosar Dem)
42		Recovery	Reset
43			Partition Reset
44			Device Reset
45		Degradation	

그림 7 | SAFE의 SAFE\_D3.6.b에서 정의한 총 36가지 Safety Mechanism

## JASPAR ISO 26262 가이드라인

### Part6-7 Software architectural design

일본 JASPAR의 ISO 26262 가이드라인에서는 Part6-7 Software architectural design에서 다음과 같은 2가지를 중요하게 강조했다.

표 4 | ISO 26262 Part6에 대한 일본과 유럽의 비교

국가	일본 JASPAR	유럽 SAFE
Part6	Part6-7.5.5 Dependent failures analysis report에 대해 SW FTA/FMEA와 병행 실시	<ul style="list-style-type: none"> <li>Part6은 SW에 의한 Safety Mechanism에 대해 총 36가지로 정형화</li> <li>1대 차량의 모든 ECU에 적용된 Safety Mechanism을 XML로 전산화함</li> </ul>
	일본과 유럽은 공통적으로 ISO 26262를 준수한 AUTOSAR 적용 방법에 대해 구체적으로 기술	

#### ① 초기 SW 아키텍처 설계 진행 시 SW-C 재사용성 고려:

ISO 26262 Part6-7.4.6 안전에 관련된 SW-C에 대해 3가지 분류(newly developed, reused with modifications, reused without modifications) 중 1가지를 선택해 표기해야 한다고 말하지만, JASPAR에서는 4가지(신규 개발, 수정, 재사용, COTS: 외부에서 구매한 SW-C)로 분류했다. 특히 SW-C가 reused without modifications(재사용)인 경우 Part8-12 Qualification of software components으로 전산화해 SW-C를 관리함으로 SW 개발 비용을 절감할 수 있다고 했다.

#### ② SW의 Safety Analysis로 SW FTA/FMEA를 실시하면서, 종속고장 분석(Dependent failures analysis)을 병행해 통합작성: [Dependent failures analysis에 대해서는 JASPAR의 Handbook for FS(Software Partitioning edition).pdf를 참고하기 바란다.]

- FTA에서 Dependent failures analysis로 cascading failure를 찾으려면, FTA의 Intermediate Events(그림 6)에서 변수값을 중심으로 찾을 수 있으며, 그것은 ASIL 등급이 높은 SW-C의 변수값이 cascading failure의 대상이 된다. FTA에서 Dependent failures analysis로 common cause failure를 찾으려면, FTA의 Basic Event(그림 6)로부터 서로 다른 SW-C가 공통으로 사용하는 인터페이스, 공유 라이브러리를 중심으로 찾을 수 있다.

common cause failure의 근본 원인(Root Cause)은 FTA의 AND gate에 failure를 발생시키 때문에 Safety Mechanism이 필요하다. 또 SW 라이브러리 등은 높은 ASIL 등급을 적용해서 개발할 필요가 있다.

- FMEA에서 Dependent failures analysis는 FTA에서 cascading failure의 분석에서 누락된 것이 없는지 확인하고, 비안전 관련 SW-C → 안전 SW-C로, 혹은 낮은 ASIL의 SW-C → 높은 ASIL의 SW-C로 간섭이 없는지(freedom from interference) 확인한다.

## SAFE ISO 26262 가이드라인

### Part6 SW의 Safety Mechanism

유럽 SAFE의 ISO 26262 가이드라인은 "SAFE\_D3.6.b.pdf"의 "Deliverable D3.6.b: Safety Code Generator Specification"에서 BMW와 독일 포티스(Fortiss) 연구소가 공동 연구해 Part6 SW에서 필요한 Safety Mechanism을 총 36가지로 정의했다(그림 7). 또 XML 기반 Configuration 방법으로 Safety Mechanism의 소스 코드를 자동 생성하는 방법을 제시했다. 자세한 내용은 "http://www.safe-project.eu/SAFE-Publications/SAFE\_D3.6.b.pdf"를 참고하면 된다.

Safety Mechanism이 총 36가지로 정형화가 되면, OEM 입장에서는 서플라이어가 어떤 종류의 SW Safety Mechanism를 적용했는지 XML로 정리해 전산화할 수 있다. 또 OEM은 서플라이어로부터 모은 정보로 새로운 차량을 개발할 때 다른 서플라이어에게 필요한 Safety Mechanism을 요구할 수 있다.

최근 OEM이 요청하는 대표적인 Safety Mechanism은 Fault Avoidance로 Freedom From Interference인 Partitioning 기능 MPU(Memory Protection Unit)가 있다. 이것은 반드시 반도체 MCU에서 HW적으로 MPU 기능을 지원하고, SW적으로 MPU를 제어할 수 있도록 개발하는 것이다. 주로 AUTOSAR OS SC3의 Memory protection 기능을 사용한다.

## 일본과 유럽의 ISO 26262의 Part6 가이드라인 비교


일본과 유럽의 ISO 26262 가이드라인은 서로 보완하는 관계임으로 비교보다는 모두를 자세히 읽어봐야 한다. 일본의 ISO 26262 Part6는 Part6-7.5.5 Dependent failures analysis report에 대한 내용을 자세히 다루고 있다. 유럽의 ISO 26262 Part6은 SW에 의한 Safety Mechanism에 대해 자세히 설명하고 있다.

국내 자동차 산업 기업들이 ISO 26262를 준수해 해외 OEM에 효과적으로 대응하는데 도움이 될 수 있도록 일본 JASPAR와 유럽 SAFE의 ISO 26262 가이드라인에서 중요한 부분을 뽑아 다뤄왔다. 총 5회에 걸쳐 연재했지만, 수천 페이지 분량의 ISO 26262 가이드라인을 효과적으로 다루는 데는 부족함이 있었다.

ISO 26262 2nd Edition에서는 전기차와 차량 보안에 대한 내용이 많이 언급되고 있다. 전기차 기반 자율주행차 개발을 위한 내용이 중점이다. 현재는 IEC61508 2nd Edition이 모든 기능안전성 분야에 기초가 되고 있지만, 향후에는 ISO 26262 2nd Edition이 모든 기능안전성 분야의 기초가 될 것이다.

왜냐하면 보안(Security)과 IoT(Internet of thing)가 포함돼 네트워크 망과 기능안전성이 요구되는 모든 제품 개발의 초석이 되기 때문이다. 특히 고령화에 따른 서비스 로봇과 의료기기 등에 적용해야할 무인 이동체의 기능안전성 요구사항이 명확해진다.

<저작권자(c)스마트엔컴퍼니. 무단전재-재배포금지>

 PDF 원문보기

본 기사의 전문은 PDF문서로 제공합니다. (로그인필요)

다운로드한 PDF문서를 웹사이트, 카페, 블로그등을 통해 재배포하는 것을 금합니다. (비상업적 용도 포함)

100자평 쓰기

로그인

로그인후 입력하세요

등록

Advertising / Media Partnership / Sponsoring

(/member/inquiry.asp?sel\_type=ad)

회사소개 (<http://www.smartn.co.kr>) 개인정보취급방침 (/member/protect.asp) 이메일주소 무단수집 거부 (/member/noemailcollect.asp)

온라인 문의 (/member/inquiry.asp) 정기구독 신청 ([http://www.smartn.co.kr/book/book\\_detail.asp?p\\_no=B00033](http://www.smartn.co.kr/book/book_detail.asp?p_no=B00033))

정기구독 주소변경 (/member/subs\_edit.asp)

스마트엔컴퍼니(주) 대표이사 : 박성규 사업자등록번호 : 108-81-64739 통신판매업신고 : 2019-서울구로-2138호

서울특별시 구로구 디지털로34길 43, 607호(구로동, 코오롱사이언스밸리1차) P: (Phone) 02-841-0017 F: (Fax) 02-841-0584 ✉ [webmaster@smartn.co.kr](mailto:webmaster@smartn.co.kr)

© Smart & Company