

SCOPE OF APPLICATION All Project/Engineering	HYUNDAI AutoEver	SHT/SHTS 1 / 81
Responsibility: Classic AUTOSAR team	AUTOSAR E2E User Manual	DOC. NO
AUTOSAR E2E User Manual		

Document Change Histroy				
Date (YYYY-MM-DD)	Ver.	Editor	Chap	내용(개정 전 -> 개정 후)
2016-04-01	1.1.0	Seongmin Kim	All	• First Release
2016-08-16	1.1.1	Seongmin Kim	3.2 5.2 5.3 6.3 8.1	• AUTOSAR E2E Stack 설명 수정 • RteEndToEndProtectionWrapperMode 추가 • VariableDataPrototypeInSystemInstanceRef 추가 • E2E Protection Wrapper 관련 API 추가 • E2E Protection Wrapper 관련 내용 추가
2016-10-20	1.1.2	Seongmin Kim	3.1 3.2 4.3 5.2 5.4 6.1 6.2 6.3 7.1 7.2 8.1 8.2	• AUTOSAR Library에 대한 설명으로 변경 • AUTOSAR E2E Stack 내용 상세화 • Deviations/Limitations 추가 • Crc 추가 • Software Component Template 내용 상세화 • 영문 내용 한글화 • 영문 내용 한글화 • 영문 내용 한글화 및 관련 내용 상세화 • Bend 옵션 추가 • Usage of E2E Library (Rte) 관련 Error Message 추가 • Application Note 상세화 • 설계시 유의사항 추가
2016-11-26	1.1.3	Seongmin Kim	4.3 8.2.2	• Change Log 형식 변경 • 일부 오류 내용 삭제 및 신규 내용 추가
2017-02-06	1.1.4	Seongmin Kim	4.2 4.3	• 신규 버전 릴리즈 관련 정보 추가
2017-07-18	1.1.5	Seongmin Kim Gongbin Lim	2 3.2.1 4.2 4.3.1 4.5.1 6.1 6.3 8.2.1	• E2E Profile 5, E2E Profile 6, E2E State Machine 관련 내용 추가 • 설계시 유의 사항 (E2E Profile 1, 2의 초기화 관련 내용, E2E State Machine 관련 내용) 추가
2018-03-22	1.2.0	Gongbin Lim	2 3.2 4 6	• E2E Profile 11 관련 내용 추가

Edition Date: 2021/12/29	File Name: E2E_UM.docx	Creation Gongbin Lim	Check JeongSu Lim	Approval Seunghoon Yoo
Document Management System		2021/12/29	2021/12/29	2021/12/29

2018-06-18	1.2.1	Gongbin Lim	3.2.2.3 4.3 8.2	• Change Log 추가, E2EXf 사용 시 유의사항 추가
2019-11-22	1.2.2.0	Gongbin Lim	4.2 4.3	• Scope of the release 수정, Change Log 추가
2020-08-05	1.2.3.0	Gongbin Lim	4.2 4.3	• Scope of the release 수정, Change Log 추가
2021-12-29	1.2.4.0	Gongbin Lim	4.2 4.3	• Scope of the release 수정, Change Log 추가
2022-08-12	1.2.5.0	Seungjin Noh	4.2 4.3	• Scope of the release 수정, Change Log 추가

Table of Contents

1. OVERVIEW	8
2. REFERENCE.....	8
3. AUTOSAR SYSTEM	9
3.1 Overview of Software Layers	9
3.2 AUTOSAR E2E Stack	9
3.2.1 E2E Library	11
3.2.1.1 E2E Profile 1.....	11
3.2.1.2 E2E Profile 2.....	12
3.2.1.3 E2E Profile 5.....	13
3.2.1.4 E2E Profile 6.....	14
3.2.1.5 E2E Profile 11	14
3.2.1.6 E2E State Machine	16
3.2.2 Usage of E2E Library	19
3.2.2.1 E2E Protection Wrapper	19
3.2.2.2 COM E2E Callouts	19
3.2.2.3 E2E Transformer.....	20
4. PRODUCT RELEASE NOTES	20
4.1 Overview.....	20
4.2 Scope of the release	21
4.3 Change Log	21
4.3.1 Version 1.2.4.0	21
4.3.2 Version 1.2.3.0	22
4.3.3 Version 1.2.2.0	22
4.3.4 Version 1.2.1	22
4.3.5 Version 1.2.0	22
4.3.6 Version 1.1.5	23
4.3.7 Version 1.1.4	23
4.3.8 Version 1.1.3	23
4.3.9 Version 1.1.2	24
4.3.10 Version 1.1.1.....	24
4.3.11 Version 1.1.0.....	24

4.3.12 Version 1.0.0.....	25
4.4 Limitations	25
4.4.1 E2E Library	25
4.4.2 Usage of E2E Library (Rte).....	26
4.5 Deviations	26
4.5.1 E2E Library	26
4.5.2 Usage of E2E Library (Rte).....	26
5. CONFIGURATION GUIDE.....	26
5.1 E2E.....	26
5.2 Crc	26
5.2.1 CrcGeneral Container	26
5.3 Rte	27
5.3.1 RteGeneration Container.....	27
5.4 Software Component Template	27
5.4.1 EndToEndProtectionSet Container	27
5.4.2 EndToEndProtection Container	27
5.4.3 EndToEndProfile Container	27
5.4.4 EndToEndProtectionVariablePrototype Container.....	28
5.4.5 VariableDataPrototypeInSystemInstanceRef Container	28
5.4.6 EndToEndProtectionISignalIPdu Container.....	28
5.4.7 NonQueuedSenderComSpec/NonQueuedReceiverComSpec	29
5.5 Com	29
5.5.1 ComIPdu Container	29
6. APPLICATION PROGRAMMING INTERFACE (API)	30
6.1 Type Definitions.....	30
6.1.1 E2E Profile 1 types	30
6.1.1.1 E2E_P01ConfigType	30
6.1.1.2 E2E_P01DataIDMode.....	30
6.1.1.3 E2E_P01SenderStateType.....	31
6.1.1.4 E2E_P01ReceiverStateType	31
6.1.1.5 E2E_P01ReceiverStatusType.....	31

6.1.2 E2E Profile 2 types	32
6.1.2.1 E2E_P02ConfigType	32
6.1.2.2 E2E_P02SenderStateType	33
6.1.2.3 E2E_P02ReceiverStateType	33
6.1.2.4 E2E_P02ReceiverStatusType	33
6.1.3 E2E Profile 5 types	34
6.1.3.1 E2E_P05ConfigType	34
6.1.3.2 E2E_P05ProtectStateType	35
6.1.3.3 E2E_P05CheckStateType	35
6.1.3.4 E2E_P05CheckStatusType	35
6.1.4 E2E Profile 6 types	36
6.1.4.1 E2E_P06ConfigType	36
6.1.4.2 E2E_P06ProtectStateType	36
6.1.4.3 E2E_P06CheckStateType	37
6.1.4.4 E2E_P06CheckStatusType	37
6.1.5 E2E Profile 11 types	37
6.1.5.1 E2E_P11DataIDMode	37
6.1.5.2 E2E_P11ConfigType	38
6.1.5.3 E2E_P11ProtectStateType	38
6.1.5.4 E2E_P11CheckStateType	38
6.1.5.5 E2E_P11CheckStatusType	38
6.1.6 E2E State Machine types	39
6.1.6.1 E2E_PCheckStatusType	39
6.1.6.2 E2E_SMConfigType	39
6.1.6.3 E2E_SMCheckStateType	40
6.1.6.4 E2E_SMStateType	40
6.2 Macro Constants	40
6.2.1 Error Flags by E2E Library	40
6.2.2 Error Flags by E2E Protection Wrapper functions on byte 1 of the return value	41
6.3 Functions	42
6.3.1 E2E Library – E2E Profile 1 routines	42
6.3.1.1 E2E_P01Protect	42
6.3.1.2 E2E_P01Check	43
6.3.2 E2E Library – E2E Profile 2 routines	43
6.3.2.1 E2E_P02Protect	43
6.3.2.2 E2E_P02Check	44
6.3.3 E2E Library – E2E Profile 5 routines	45
6.3.3.1 E2E_P05Protect	45
6.3.3.2 E2E_P05ProtectInit	45
6.3.3.3 E2E_P05Check	46
6.3.3.4 E2E_P05CheckInit	47
6.3.3.5 E2E_P05MapStatusToSM	47
6.3.4 E2E Library – E2E Profile 6 routines	48
6.3.4.1 E2E_P06Protect	48
6.3.4.2 E2E_P06ProtectInit	48
6.3.4.3 E2E_P06Check	49

6.3.4.4 E2E_P06CheckInit	50
6.3.4.5 E2E_P06MapStatusToSM	50
6.3.5 E2E Library – E2E Profile 11 routines.....	51
6.3.5.1 E2E_P11Protect	51
6.3.5.2 E2E_P11ProtectInit	52
6.3.5.3 E2E_P11Check	52
6.3.5.4 E2E_P11CheckInit	53
6.3.5.5 E2E_P11MapStatusToSM	53
6.3.6 E2E Library – E2E State Machine routines.....	54
6.3.6.1 E2E_SMCheck.....	54
6.3.6.2 E2E_SMCheckInit.....	55
6.3.7 E2E Protection Wrapper	55
6.3.7.1 Single channel wrapper routines and init routines	55
6.3.7.2 Redundant wrapper routines	59
6.3.8 COM E2E Callouts	66
6.3.8.1 IPDU_E2EProtect_<IPDU ID> ()	66
6.3.8.2 IPDU_E2ECheck_<IPDU ID> ()	66
7. GENERATOR.....	68
7.1 Generator Option.....	68
7.1.1 E2E	68
7.1.2 Usage of E2E Library (Rte).....	68
7.2 Generator Error Message.....	68
7.2.1 E2E	68
7.2.1.1 Error Messages.....	68
7.2.1.2 Warning Messages	68
7.2.1.3 Information Messages	68
7.2.2 Usage of E2E Library (Rte).....	68
7.2.2.1 Error Messages.....	68
7.2.2.2 Warning Messages	71
7.2.2.3 Information Messages	71
8. APPENDIX.....	72
8.1 Application Note (Usage of E2E Library).....	72
8.1.1 E2E Protection Wrapper	72
8.1.1.1 UsesEndToEndProtection.....	72
8.1.1.2 EndToEndProtectionSet.....	72
8.1.1.3 EndToEndProtection	73
8.1.1.4 EndToEndProfile.....	73
8.1.1.5 EndToEndProtectionVariablePrototype	74
8.1.2 Com E2E Callouts	75
8.1.2.1 EndToEndProtection!SignalIPdus	75
8.1.2.2 ComIPduCallout.....	76

8.2 설계시 유의사항.....	77
8.2.1 E2E	77
8.2.2 Usage of E2E Library (Rte).....	78
8.2.3 E2EXf 사용 시 유의사항.....	78

1. Overview

본 문서는 AUTOSAR 플랫폼에서 AUTOSAR 표준을 만족하는 End-To-End Protection 기능을 사용하는 경우 사용자가 파라미터 설정 또는 시스템 설계시 주의하거나 참고할 사항을 제공한다.

자세한 사항은 Reference 문서를 참고한다.

설정관련 Category 의 해석은 다음과 같다.

- Changeable (C) : User 에 의해서 설정 가능한 항목
- Fixed (F) : User 에 의한 변경이 불가능한 항목
- NotSupported (N) : 사용되지 않는 항목

2. Reference

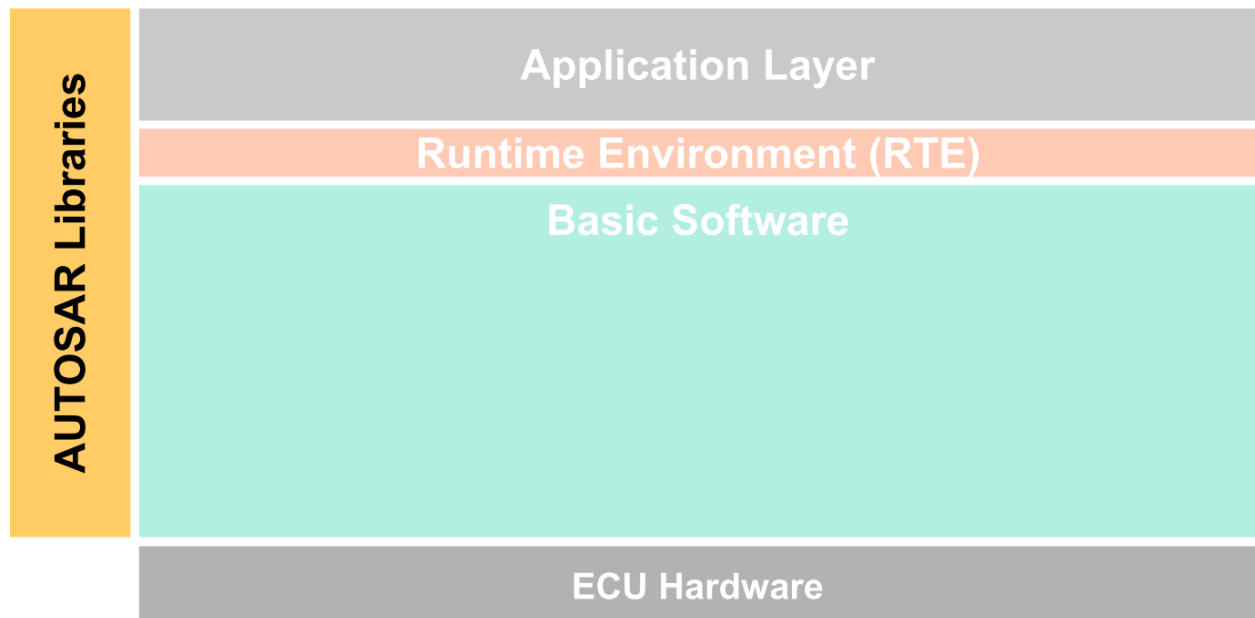
Sl. No.	Title	Version
1.	AUTOSAR_SRS_Libraries (AUTOSAR_SRS_Libraries.pdf)	2.1.0
2.	Specification of SW-C End-to-End Communication Protection Library (AUTOSAR_SWS_E2ELibrary.pdf)	2.0.0
3.	Specification of SW-C End-to-End Communication Protection Library (AUTOSAR_SWS_E2ELibrary.pdf)	4.3.0
4.	Specification of SW-C End-to-End Communication Protection Library (AUTOSAR_SWS_E2ELibrary.pdf)	4.3.1
5.	Software Component Template (AUTOSAR_TPS_SoftwareComponentTemplate.pdf)	4.2.0

3. AUTOSAR System

3.1 Overview of Software Layers

AUTOSAR 의 Software Architecture 에서 Library 는 특수한 목적을 위한 Function 을 모아둔 것으로 다음의 특징을 가진다.

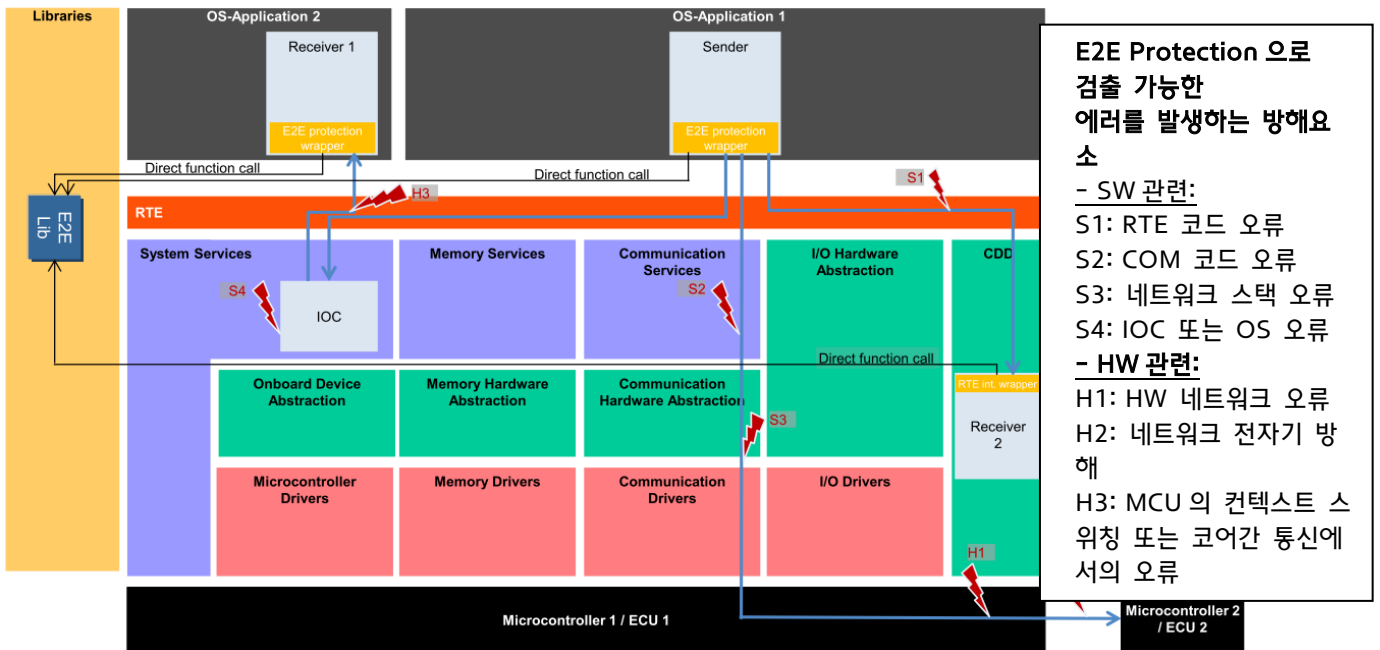
- BSW 모듈(RTE 를 포함), 소프트웨어 컴포넌트, 라이브러리, 또는 Integration 코드에서 호출될 수 있다.
- Caller 의 Context 와 같은 Protection Environment 에서 실행된다.
- Library 만 호출할 수 있다.
- 재진입이 가능하다.
- Internal State 를 갖지 않는다.
- Initialization 을 필요로 하지 않는다.
- Synchronous 하다. 즉, Wait Point 를 갖지 않는다.



3.2 AUTOSAR E2E Stack

E2E 는 제어기의 실행 중 데이터 송·수신 과정에서 다음과 같은 결함(Fault)의 발생으로 인한 영향으로부터 데이터를 보호하는 개념으로부터 출발한다. 이러한 결함의 예는 다음과 같다.

- HW Faults (예: CAN 트랜시버의 레지스터 오류)
- 전파 방해 (예: EMC)
- VFB 를 구현한 소프트웨어 내 시스템 오류 (예: RTE, IOC, COM 및 네트워크 스택)



EN50159 와 ISO/DIS26262 에서 메시지 교환 통신에서 고려되어야 하는 결함(Fault)은 다음과 같다.

1. 중복 (Repetition): 동일한 메시지를 두 번 이상 수신
2. 삭제 (Deletion): 메시지 전체 또는 일부가 삭제
3. 삽입 (Insertion): 메시지 전체 또는 일부가 삽입
4. 잘못된 순서 (Incorrect sequence): 메시지들을 잘못된 순서로 수신
5. 변형 (Corruption): 메시지 전체 또는 일부에 변형이 발생
6. 시간적 결함(지연) (Timing faults(delay)): 메시지의 시간적 제약사항 위반 (예: 메시지가 너무 늦게 수신)
7. 어드레싱 결함 (Addressing faults): 결함으로 인해 메시지를 잘못된 목적지로 전송 후 정상수신으로 처리
8. 불일치 (Inconsistency): 결함으로 인해 네트워크 상태나 전송 데이터를 통신 노드들이 서로 다르게 보유
9. 위장 (Masquerade): 위조 데이터를 가진 수신 메시지를 적절한 송신자가 보낸 진본인 것처럼 설계

위와 같은 결함 발생으로 인한 영향성에 대해 데이터를 보호하기 위해 E2E Library 는 다음의 메커니즘을 제공한다.

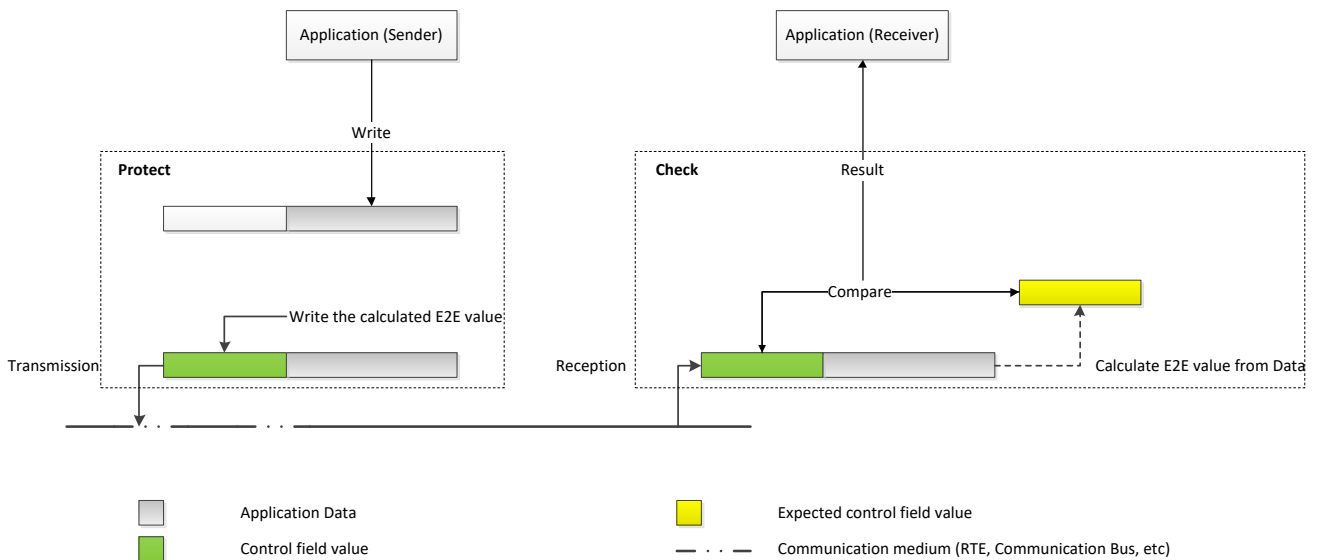
1. Safety 데이터에 컨트롤 데이터를 붙여 RTE 상 전송을 통해 데이터를 보호한다.
2. RTE로부터 수신된 Safety 데이터를 이 컨트롤 데이터를 사용하여 검증한다.
3. 수신된 Safety 데이터의 결함을 표시하고, 이는 수신자 SW-C 에 의해 처리된다.

3.2.1 E2E Library

E2E Library 는 다음의 기능을 제공한다.

- Sender: 전송 데이터에 CRC 나 Counter 같은 컨트롤 필드를 E2E Header 영역에 업데이트 (Protect)
- Receiver: 수신 데이터의 E2E Header 를 평가 (Check)

(수신 데이터에 대해 예상 컨트롤 필드 값을 계산하여 수신한 값과 비교)



AUTOSAR 는 융통성과 표준성을 고려하여 적절한 솔루션을 제공하기 위해 E2E 보호 메커니즘들을 결합하여 유연한 E2E 프로파일 세트를 정의한다. 각각의 E2E 프로파일은 고정된 동작 방식을 가지지만 Function 파라미터를 통해 일부 설정 옵션을 제공한다. (예: 보호될 데이터와 관련된 CRC 위치) 4.0.3 에서 제공하는 프로파일은 Profile 1, Profile 2 가 있다.

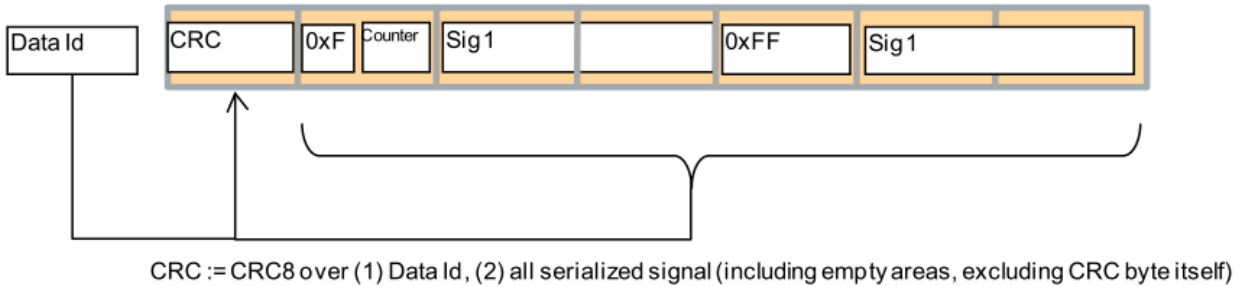
※ AUTOSAR 4.0.3 사양에서는 8 bit CRC (Profile 1: Crc_CalculateCrc8(), Profile 2: Crc_CalculateCrc8H2F())를 지원한다. AUTOSAR 4.3.1 사양에서는 8 bit CRC (Profile 11 : Crc_CalculateCrc8()) 를 지원한다. 현대오트론의 E2E Library 는 16 bit CRC (Crc_CalculateCrc16())을 지원하기 위해 4.3.0 의 Profile 5, Profile 6 를 제공한다.

3.2.1.1 E2E Profile 1

AUTOSAR 사양에서 제시하는 E2E Profile 1 의 기능은 다음과 같다.

Mechanism	Detected failure modes
Counter	Repetition, deletion, insertion, incorrect sequence
Timeout	Deletion, delay
Data ID	Insertion, addressing faults
CRC	Corruption

E2E Profile 1 의 데이터 레이아웃은 다음과 같이 표현할 수 있다. (예시)



- Counter: Counter 영역은 4 비트이며, Counter 의 위치는 사용자가 지정할 수 있다.
Counter 값은 Sender 에서 Protect 요청 시 마다 1 씩 증가하며, 범위는 0 ~ 14 이다.
- Data ID: Data ID 는 16 비트이며, Data ID 는 직접 전송하지 않고 CRC 계산에 사용된다. (그림 참조)
사용자는 Data ID Mode 를 통해 CRC 계산에서 Data ID 의 사용 방식을 지정할 수 있다.
Data ID Mode 는 다음과 같은 종류가 있다.
 1. E2E_P01_DATAID_BOTH: 2 바이트인 Data ID 을 상위/하위 바이트로 나누어 모두 CRC 계산에 사용한다.
 2. E2E_P01_DATAID_ALT: 현재 Counter 값이 홀수 이면 상위 바이트를, 짝수이면 하위 바이트를 CRC 계산에 사용한다.
 3. E2E_P01_DATAID_LOW: Data ID 의 하위 바이트만 CRC 계산에 사용한다.
- CRC: CRC 영역은 8 비트이며, CRC 위치는 사용자가 지정할 수 있다.
CRC 계산 시, Crc 모듈의 Crc_CalculateCRC8 () API 를 사용한다.
- Timeout: Receiver 에서 Check 시, 새 데이터의 사용 가능 여부 및 Counter 를 통해 데이터가 중복인지 확인하여, Timeout 발생 여부를 확인한다.

E2E Profile1 은 Variant 가 존재하며, 다음과 같다.

- Variant 1A: CRC 위치는 0 번째 비트, Counter 위치는 8 번째 비트로 위치를 고정하고, Data ID 는 상위/하위 바이트를 모두 사용한다. (E2E_P01_DATAID_BOTH)
- Variant 1B: CRC 위치는 0 번째 비트, Counter 위치는 8 번째 비트로 위치를 고정하고, Data ID 는 상위/하위 바이트를 번갈아 사용한다. (E2E_P01_DATAID_ALT)

3.2.1.2 E2E Profile 2

AUTOSAR 사양에서 제시하는 E2E Profile 2 의 기능은 다음과 같다.

Mechanism	Detected failure modes
Counter	Unintended message repetition, message loss, insertion of messages, re-sequencing
Data ID	Insertion of messages, masquerading
CRC	Message corruption, insertion of messages (masquerading)
Timeout (detection and handling implemented by SW-C)	Message loss, message delay

E2E Profile 2 의 데이터 레이아웃은 다음과 같이 표현할 수 있다. (예시)



- Counter: Counter 영역은 4 비트이며, Counter 의 위치는 8 번째 비트로 고정된다.
Counter 값은 Sender 에서 Protect 요청 시 마다 1 씩 증가하며, 범위는 0 ~ 15 이다.
- Data ID: Data ID 는 8 비트이며, 총 16 개의 Data ID 세트를 사용한다.
Data ID 는 직접 전송하지 않고 CRC 계산에 사용되며,
현재 Counter 의 값에 해당하는 인덱스의 Data ID 를 사용하여 CRC 를 계산한다.
(예를 들어, 현재 Counter 가 8 이면, 8 번째 Data ID 를 사용하여 CRC 를 계산한다.)
- CRC: CRC 영역은 8 비트이며, CRC 위치는 0 번째 비트로 고정된다.
CRC 계산 시, Crc 모듈의 Crc_CalculateCRC8H2F() API 를 사용한다.
- Timeout: Receiver 에서 Check 시, 새 데이터의 사용 가능 여부 및
Counter 를 통해 데이터가 중복인지 확인하여, Timeout 발생 여부를 확인한다.

3.2.1.3 E2E Profile 5

AUTOSAR 사양에서 제시하는 E2E Profile 5 의 기능은 다음과 같다.

Fault	Main safety mechanisms
Repetition of information	Counter
Loss of information	Counter
Delay of information	Counter
Insertion of information	Data ID
Masquerading	Data ID, CRC
Incorrect addressing	Data ID
Incorrect sequence of information	Counter
Corruption of information	CRC
Asymmetric information sent from a sender to multiple receivers	CRC (to detect corruption at any of receivers)
Information from a sender received by only a subset of the receivers	Counter (loss on specific receivers)
Blocking access to a communication channel	Counter (loss or timeout)

E2E Profile 5 의 헤더(CRC 와 Counter 의 위치)는 다음과 같이 표현할 수 있다.

Transmission order	0								1								2							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	E2E CRC																E2E Counter							

- Counter: Counter 영역은 8 비트이며, 헤더에서 CRC 다음에 위치한다.
Counter 값은 Sender 에서 Protect 요청 시 마다 1 씩 증가하며, 범위는 0 ~ 0xFF 이다.
- Data ID: Data ID 는 16 비트이며, 시스템 전체에서 유일성을 가진다.
Data ID 는 직접 전송하지 않고 CRC 계산에 사용된다.
- CRC: CRC 영역은 16 비트 (polynomial in normal form 0x1021 (Autosar notation))이며,
CRC 위치는 헤더의 첫 2 바이트이다.
CRC 계산 시, Crc 모듈의 Crc_CalculateCRC16() API 를 사용한다.
- Length: Length 를 직접 전송하지 않는다.
- Timeout: Receiver 에서 Check 시, 새 데이터의 사용 가능 여부 및
Counter 를 통해 데이터가 중복인지 확인하여, Timeout 발생 여부를 확인한다.

3.2.1.4 E2E Profile 6

AUTOSAR 사양에서 제시하는 E2E Profile 6 의 기능은 다음과 같다.

Fault	Main safety mechanisms
Repetition of information	Counter
Loss of information	Counter
Delay of information	Counter
Insertion of information	Data ID
Masquerading	Data ID, CRC
Incorrect addressing	Data ID
Incorrect sequence of information	Counter
Corruption of information	CRC
Asymmetric information sent from a sender to multiple receivers	CRC (to detect corruption at any of receivers)
Information from a sender received by only a subset of the receivers	Counter (loss on specific receivers)
Blocking access to a communication channel	Counter (loss or timeout)

E2E Profile 6 의 헤더(CRC, Length 와 Counter 의 위치)는 다음과 같이 표현할 수 있다.

Transmission order	0							1								2								3								
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	E2E CRC															E2E Length																
4	E2E Counter																															

- Counter: Counter 영역은 8 비트이며, 헤더에서 Length 다음에 위치한다.
Counter 값은 Sender 에서 Protect 요청 시 마다 1 씩 증가하며, 범위는 0 ~ 0xFF 이다.
- Data ID: Data ID 는 16 비트이며, 시스템 전체에서 유일성을 가진다.
Data ID 는 직접 전송하지 않고 CRC 계산에 사용된다.
- CRC: CRC 영역은 16 비트 (polynomial in normal form 0x1021 (Autosar notation))이며,
CRC 위치는 헤더의 첫 2 바이트이다.
CRC 계산 시, Crc 모듈의 Crc_CalculateCRC16() API 를 사용한다.
- Length: Length 영역은 16 비트이며, 헤더에서 CRC 다음에 위치한다..
전송하는 Data 의 길이를 나타낸다.
- Timeout: Receiver 에서 Check 시, 새 데이터의 사용 가능 여부 및
Counter 를 통해 데이터가 중복인지 확인하여, Timeout 발생 여부를 확인한다.

3.2.1.5 E2E Profile 11

AUTOSAR 사양에서 제시하는 E2E Profile 11 의 기능은 다음과 같다.

Fault	Main safety mechanisms
Repetition of information	Counter
Loss of information	Counter
Delay of information	Counter
Insertion of information	Data ID
Masquerading	Data ID, CRC
Incorrect addressing	Data ID
Incorrect sequence of information	Counter
Corruption of information	CRC
Asymmetric information sent from a sender to multiple receivers	CRC (to detect corruption at any of receivers)
Information from a sender received by only a subset of the receivers	Counter (loss on specific receivers)
Blocking access to a communication channel	Counter (loss or timeout)

E2E Profile 11의 헤더(CRC, Counter의 위치)는 다음과 같이 표현할 수 있다.

Byte Order	0								1							
Transmission Order	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Order	7	6	5	4	3	2	1	0	15	14	12	12	11	10	9	8
	E2E CRC								DataIDNibble				Counter			

Option

- Counter: Counter 영역은 4 비트이며, Counter의 위치는 사용자가 지정할 수 있다.
Counter 값은 Sender에서 Protect 요청 시 마다 1씩 증가하며, 범위는 0 ~ 14 이다.
- Data ID: Data ID는 16 비트이며, Data ID는 CRC 계산에 사용된다.
사용자는 Data ID Mode를 통해 CRC 계산에서 Data ID의 사용 방식을 지정할 수 있다.
Data ID Mode는 다음과 같은 종류가 있다.
 1. E2E_P11_DATAID_BOTH: 2 바이트인 Data ID를 상위/하위 바이트로 나누어 모두 CRC 계산에 사용한다.
 2. E2E_P11_DATAID_NIBBLE: Data ID의 하위 바이트를 CRC 계산에 사용하고, 상위 바이트의 하위 nibble을 헤더에 포함하여 전송한다(사용자가 위치 지정 가능).
- CRC: CRC 영역은 8 비트이며, CRC 위치는 사용자가 지정할 수 있다.
CRC 계산 시, Crc 모듈의 Crc_CalculateCRC8 () API를 사용한다.
- Timeout: Receiver에서 Check 시, 새 데이터의 사용 가능 여부 및 Counter를 통해 데이터가 중복인지 확인하여, Timeout 발생 여부를 확인한다.

E2E Profile11은 Variant가 존재하며, 다음과 같다.

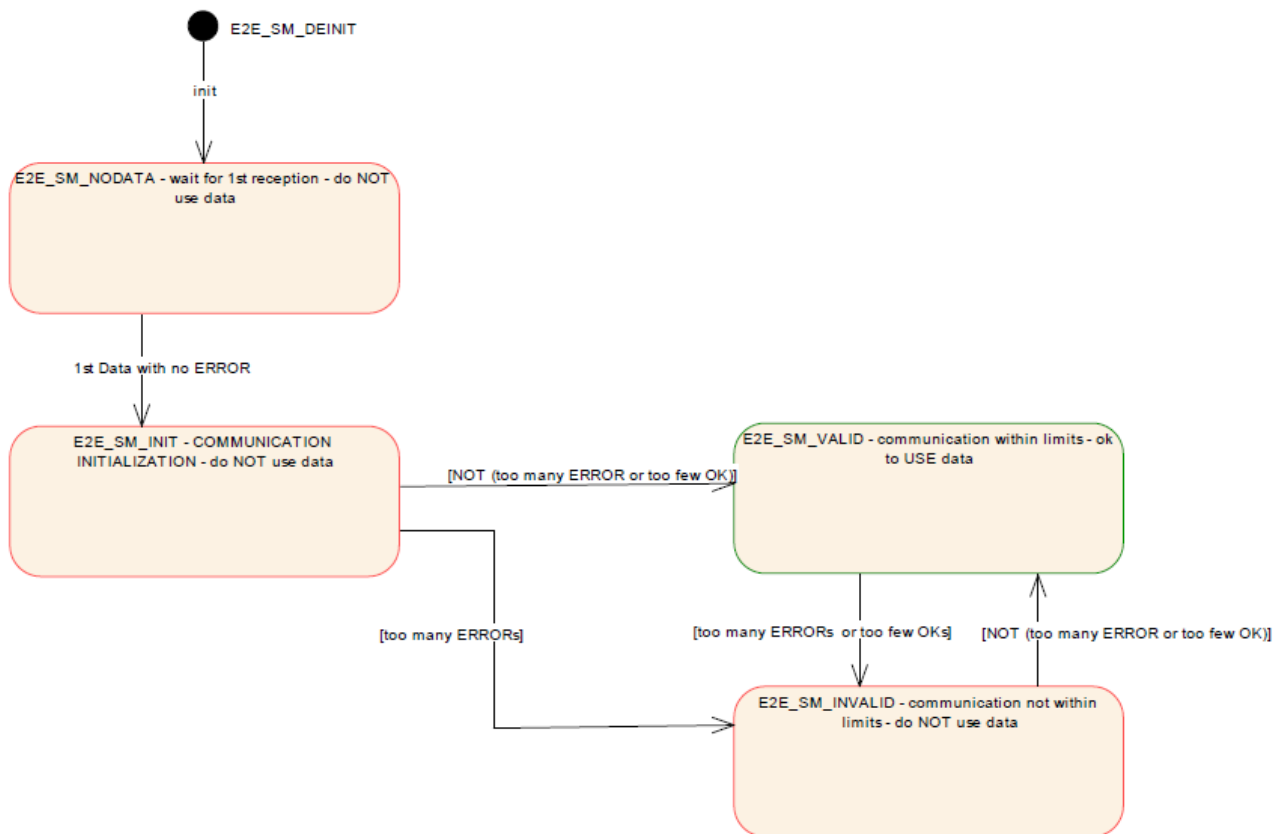
- Variant 11C: CRC 위치는 0 번째 비트, Counter 위치는 8 번째 비트, Data ID Nibble 위치는 12 번째 비트로 위치를 고정하고,
Data ID는 상위 바이트의 nibble(헤더 포함용)/하위 바이트(CRC 계산용)를 모두 사용한다.
(E2E_P11_DATAID_NIBBLE)
- Variant 11A: CRC 위치는 0 번째 비트, Counter 위치는 8 번째 비트로 위치를 고정하고,
Data ID는 상위/하위 바이트를 번갈아 사용한다. (E2E_P01_DATAID_ALT)

3.2.1.6 E2E State Machine

E2E Profile Check 함수는 각 cycle 에 한해 data 를 verify 한다. E2E State Machine 은 Check 함수의 결과 값들을 저장하여 RTE/SWC/COM 에게 제공한다.

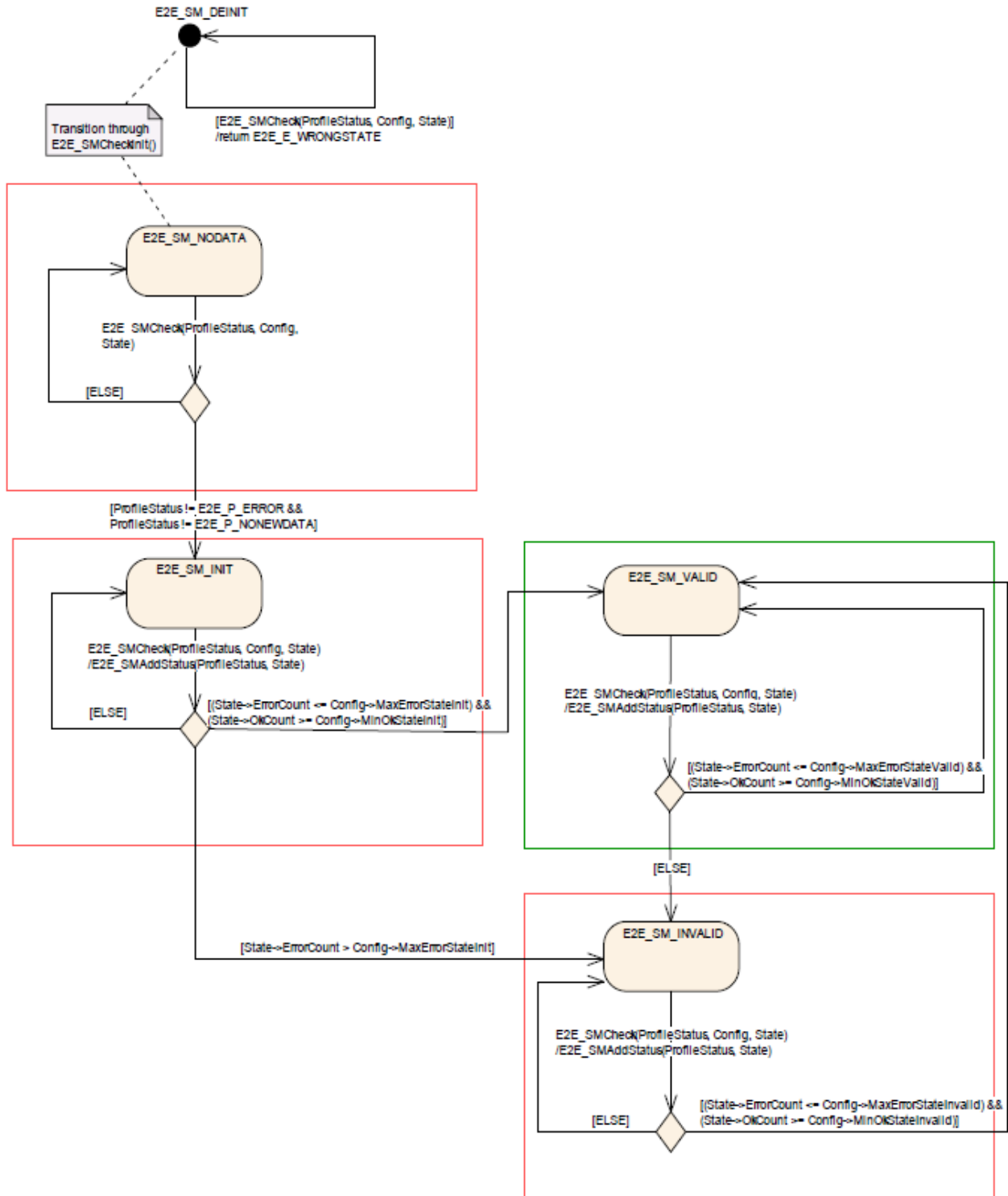
E2E State Machine 은 모든 E2E Profile 에 적용 가능하다.

E2E State Machine 의 다이어그램은 아래와 같이 요약 가능하다.



- E2E_SM_DEINIT, E2E_SM_NODATA, E2E_SM_INIT, E2E_SM_VALID, E2E_SM_INVALID 의 5 개의 State 로 구성된다.
- E2E Profile Check 함수의 결과값들에 따라, 각 state 는 상태를 변화시키거나 유지한다.

E2E State Machine 은 E2E_SMCheckInit(), E2E_SMCheck() 함수로 구성된다. E2E_SMCheck 함수의 다이어그램은 다음과 같다.



- E2E_SM_DEINIT 상태일 경우, E2E_SM_Check 함수가 호출되면 E2E_E_WRONGSTATE 의 결과값으로 종료하고, 상태를 유지한다.
- E2E_SM_CheckInit 함수가 호출되면 E2E_SM_DEINIT 상태에서 E2E_SM_NODATA 상태로 전환하고 함수를 종료한다.
- E2E_SM_NODATA 상태일 경우, E2E_SM_Check 함수가 호출되면 인자로 받은 ProfileStatus(가장 최근의 E2E Profile Check 함수 결과 상태)가 E2E_P_ERROR 와 E2E_P_NONEWDATA 가 아니면 E2E_SM_INIT 상태로 전환하고 함수를 종료한다.
- ProfileStatus 가 E2E_P_ERROR 또는 E2E_P_NONEWDATA 이면 상태를 유지하고 함수를 종료한다.
- E2E_SM_INIT 상태일 경우, E2E_SM_Check 함수가 호출되면 E2E_SMAAddStatus 함수를 수행하여 가

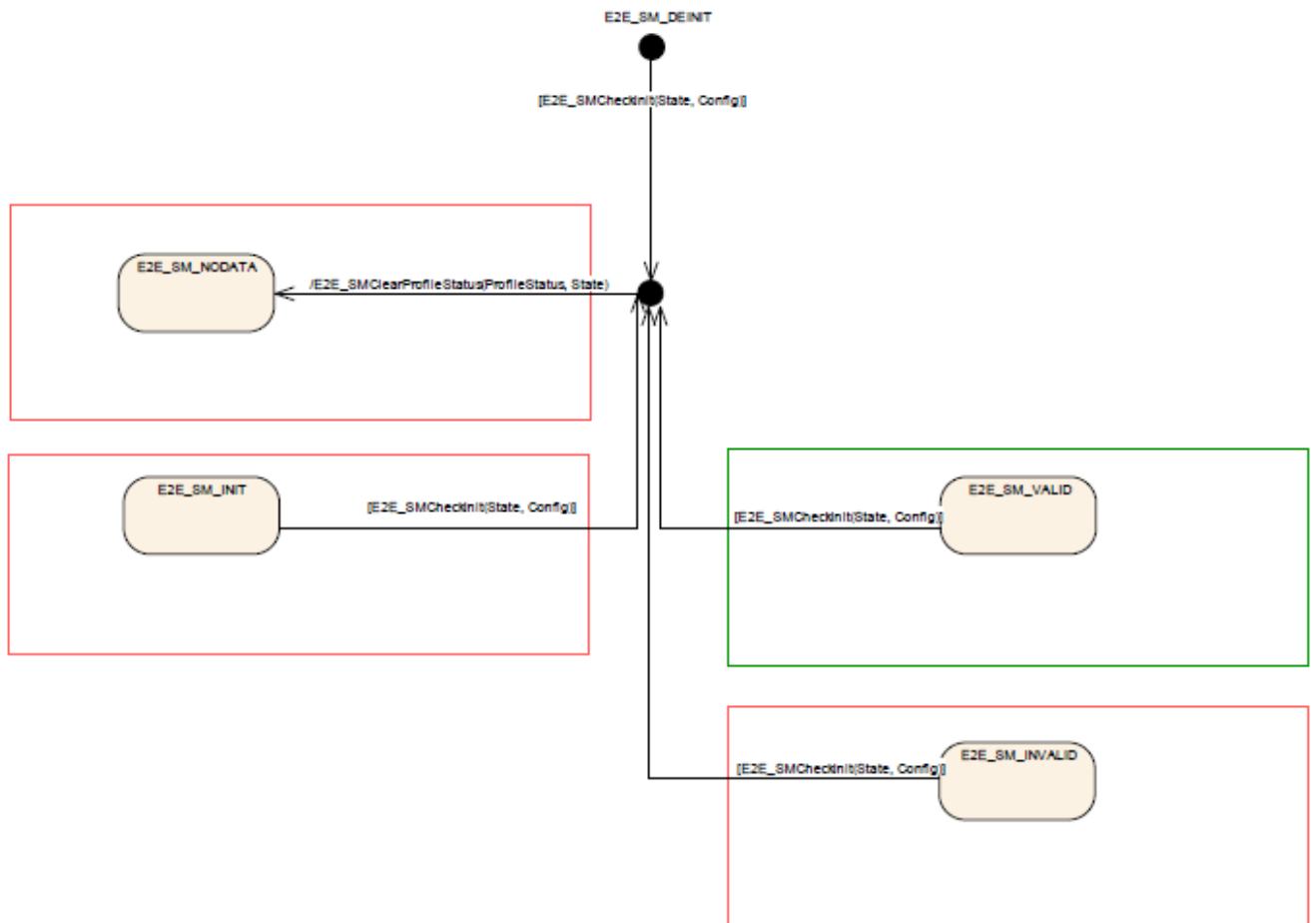
장 최근의 E2E Profile Check 함수 결과를 업데이트 한다.

- 업데이트 완료 후, 미리 설정한 ERROR 개수와 OK 개수를 비교하여 E2E_SM_VALID, E2E_SM_INVALID 로 상태를 전환하거나, 상태를 유지하고 함수를 종료한다.
- E2E_SM_VALID, E2E_SM_INVALID 상태에서 E2E_SM_Check 함수가 호출되었을 경우, 위와 유사한 방식으로 동작한다.

E2E_SMAAddStatus 함수는 다음과 같이 동작한다.

- 인자로 받은 ProfileStatus(최근 동작한 E2E Profile Check 함수의 결과 상태)를 업데이트 한다.
- 업데이트 된 ProfileStatus 저장 배열을 스캔하여 E2E_P_OK 개수를 업데이트 한다.
- 업데이트 된 ProfileStatus 저장 배열을 스캔하여 E2E_P_ERROR 개수를 업데이트 한다.
- ProfileStatus 저장 배열의 Index 를 1 증가 시킨다.

E2E_CheckInit 함수의 다이어그램은 다음과 같다.



- State Machine 의 상태와 상관없이, E2E_SMClearStatus 함수를 수행한 후, E2E_SM_NODATA 상태로 전환한다.

E2E_SMClearStatus 함수는 다음과 같이 동작한다.

- ProfileStatus 저장 배열의 원소들을 E2E_P_NOTAVAILABLE 로 초기화한다.
- E2E_P_OK 개수, E2E_P_ERROR 개수를 0 으로 초기화 한다.
- ProfileStatus 저장 배열의 Index 를 0 으로 초기화 한다.
- 따라서, ProfileStatus 저장 배열이 NULL pointer 가 되지 않도록 설정해주어야 한다(NULL pointer 일 경우, E2E_E_INPUTERR_NULL 발생)

3.2.2 Usage of E2E Library

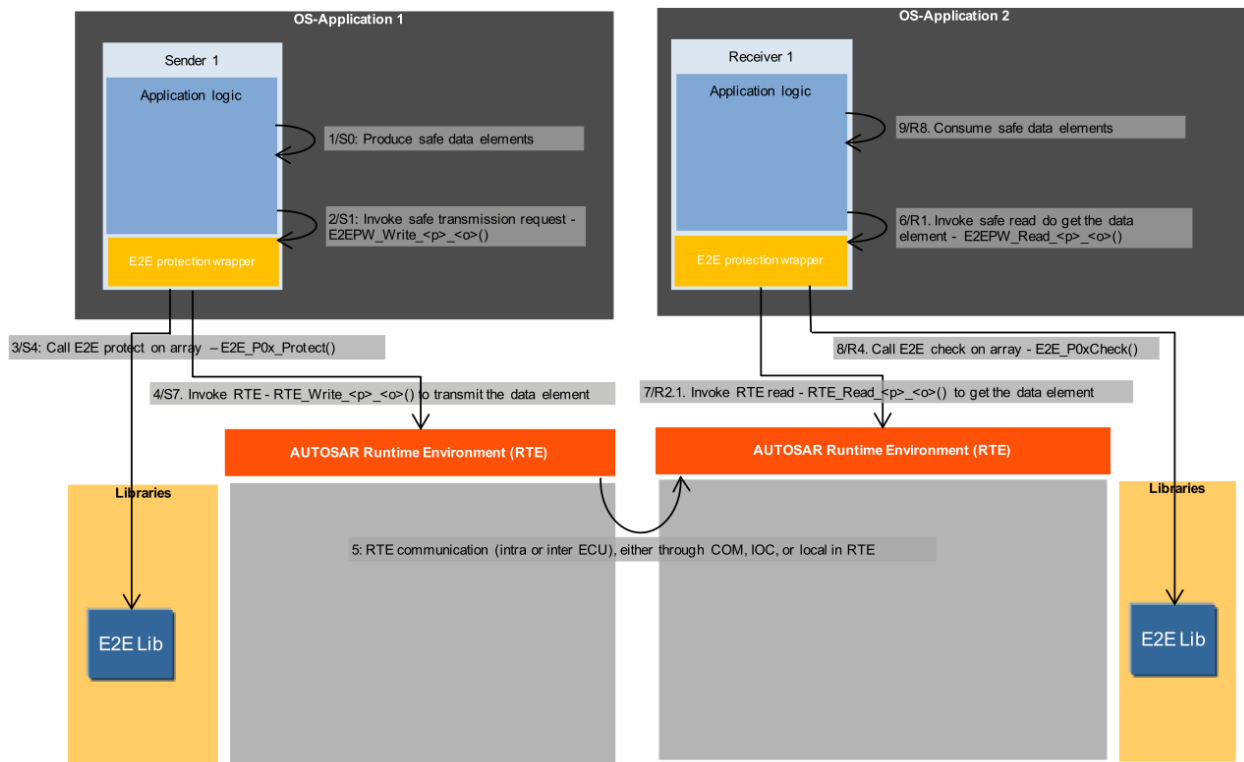
E2E Library 를 사용하기 위해 다음과 같은 솔루션을 제공한다. 관련 코드는 Rte 모듈을 통해 생성된다.

※ 아래에 기술된 E2E Protection Wrapper 와 Com E2E Callouts 는 E2E Profile 1 과 E2E Profile 2 를 지원한다. (AUTOSAR 사양)

3.2.2.1 E2E Protection Wrapper

E2E Protection Wrapper 는 Rte 의 Sender-Receiver 통신에 대응하여 Rte_Write/Rte_Read API 에 대응하는 Wrapper API 를 제공하는 기능을 한다.

E2E Protection Wrapper 는 다음과 같이 동작한다.



- Sender 의 경우 애플리케이션 데이터에 대해 E2E 라이브러리를 호출하여 E2E 보호 후, `Rte_Write` API 를 호출하여 E2E 헤더가 포함된 애플리케이션 데이터를 Rte 로 전송한다.
- Receiver 의 경우, `Rte_Read` 를 호출하여 Rte 로부터 데이터를 수신 후, E2E 라이브러리를 호출하여 E2E 헤더를 통해 데이터의 정상 여부를 판별 후, 그 결과와 수신 데이터를 리턴한다.

E2E Protection Wrapper 는

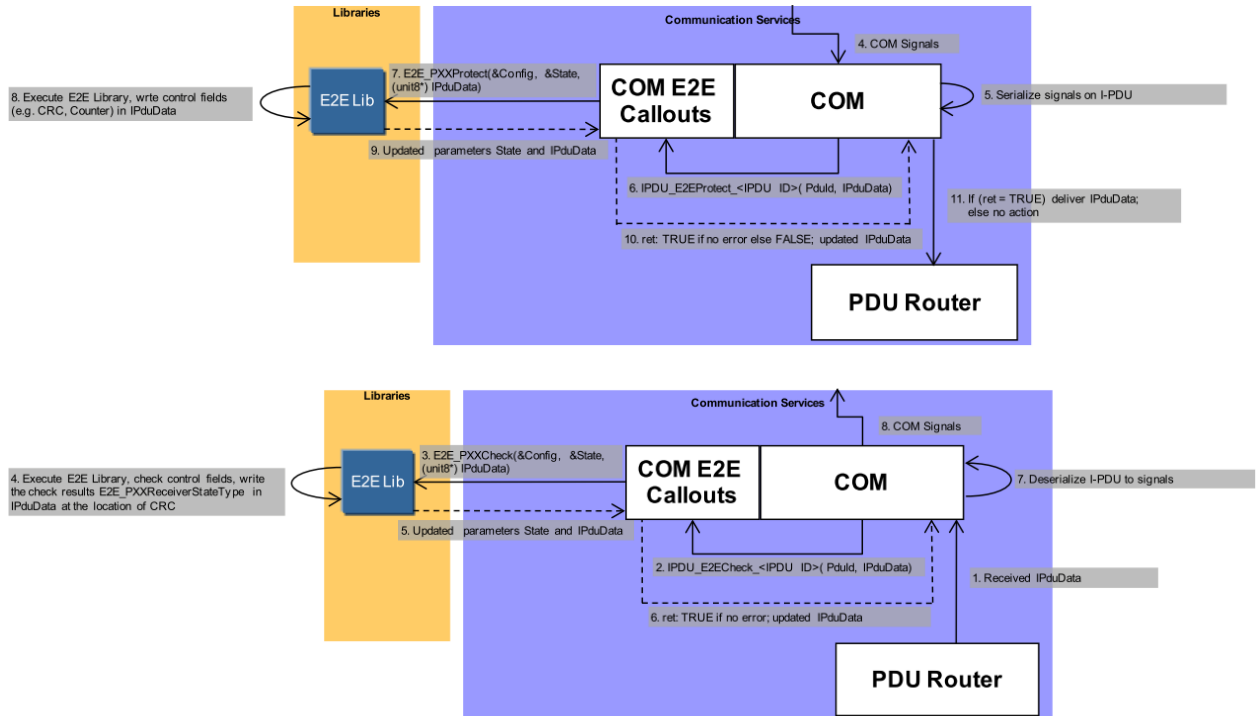
제어기 내에서 Rte 를 통한 SW-C 간 통신 및 CAN 등을 통한 제어기 간의 통신에서 모두 사용 가능하며, 제어기 간 통신의 경우, I-PDU 와 동일한 레이아웃으로 데이터를 배치하는 기능이 추가될 수 있다. (Serialization) 또한 신뢰성 확보를 위해 Protect 와 Check 를 두 번씩 수행하여 결과를 비교하는 Redundant 기능을 제공한다.

3.2.2.2 COM E2E Callouts

COM E2E Callouts 는 Com 의 I-PDU Callout 기능을 사용하여, Com 에서 I-PDU 를 PduR 로 전송하기 전 데이터를 보호하고,

PduR로부터 I-PDU를 수신한 후, 데이터를 검사하여 데이터의 사용 여부를 결정하는 기능이다.

COM E2E Callouts는 다음과 같이 동작한다.



- Sender의 경우,
Com에서 PduR로 I-PDU를 전송하기 전 I-PDU Callout에 등록된 COM E2E Callouts를 호출한다.
COM E2E Callouts는 I-PDU의 Signal Group에 대해 E2E 라이브러리를 호출하여 E2E 보호한다.
Com은 E2E 헤더가 포함된 I-PDU를 PduR로 전송한다.
- Receiver의 경우,
Com에서 PduR로부터 I-PDU를 수신하면, I-PDU Callout에 등록된 COM E2E Callouts를 호출한다.
COM E2E Callouts는 I-PDU의 Signal Group에 대해 E2E 라이브러리를 호출하여
데이터의 정상 여부를 판별하고, 그 결과를 리턴한다.
Com은 결과가 True일 경우 I-PDU를 취하고, False일 경우 I-PDU를 버린다.

COM E2E Callouts는 Com 모듈을 사용한 제어기간 통신에만 사용할 수 있으며,
Rte와 Com의 오류로 인한 결함은 탐지가 불가능하다.

3.2.2.3 E2E Transformer

오토론의 E2E Transformer는 현재 E2E Profile 5,6,11을 지원한다. 또한, E2E Transformer는 E2E State Machine을 사용한다.

E2E Transformer 관련 사항은 E2E Transformmr 유저 매뉴얼을 참고한다.

4. Product Release Notes

4.1 Overview

이 Chapter에서는, 현대오토론 E2E Stack에 대한 Release 관련 내용을 제공하는데 목적이 있으며, E2E Stack Software product release version에 대한, 제한사항 및 특이사항을 기술한다.

4.2 Scope of the release

이 문서에 대한 모든 내용은, 다음의 현대오트론 E2E stack 모듈에 한정한다.

Module	AUTOSAR version	SWS version	Module version
E2E	4.2.2	4.2.2	1.2.5.0

- ※ Module version 은 각 모듈의 BswModule Description(Bswmd)파일의 Sw version 을 의미한다.
- ※ E2E Profile 1, E2E Profile2 는 AUTOSAR version 4.0.3 (SWS version 동일)에 근거한다.
- ※ E2E Profile 5, E2E Profile6, E2E State Machine 은 AUTOSAR version 4.3.0 (SWS version 동일)에 근거한다.
- ※ E2E Profile 11 은 AUTOSAR version 4.3.1 (SWS version 동일)에 근거한다.
- ※ E2E Transformer 와 version 일치를 위해, AUTOSAR version 을 4.2.2 로 고정한다.

4.3 Change Log

4.3.1 Version 1.2.5.0

- 신규 기능
 - 없음
- 개선 사항
 - UNECE Code Security 관련 코드 업데이트

원인	- UNECE Code Security 코드 업데이트 요청
동작 영향	- 없음
설정 영향	없음
ASW 필요 조치 사항	없음

4.3.2 Version 1.2.4.0

- 신규 기능
 - 없음
- 개선 사항
 - UNECE Code Security 관련 코드 업데이트

원인	- UNECE Code Security 코드 업데이트 요청
동작 영향	- 없음
설정 영향	없음
ASW 필요 조치 사항	없음

4.3.3 Version 1.2.3.0

- 신규 기능
 - 없음
- 개선 사항
 - E2E 가이드 라인 내용 추가, MISRA / RTE 정당화

원인	- 가이드라인 내용 추가 요청, MISRA / RTE 정당화 요청
동작 영향	- 없음
설정 영향	없음
ASW 필요 조치 사항	없음

4.3.4 Version 1.2.2.0

- 신규 기능
 - 없음
- 개선 사항
 - lib 폴더에 있는 .c 파일을 delivery 폴더로 이동, MISRA-C 2012 적용

원인	- 소스코드 오픈 요청, MISRA-C 2012 적용 요청
동작 영향	- 없음
설정 영향	없음
ASW 필요 조치 사항	없음

4.3.5 Version 1.2.1

- 신규 기능
 - 없음
- 개선 사항
 - E2E_P11Check API 의 파라미터 검사 로직 수정

원인	- E2E_P11Check 시, Input 검사 로직 구현이 AUTOSAR 사양과 다름
동작 영향	- E2E_P11Check 의 인자 중에서 DataPtr, Length 를 NULL_PTR, 0 으로 받을 경우 기존에는 E2E_E_INPUTERR_WRONG 이 return 되었으나, 수정 후에는 E2E_E_OK 로 return 되고 State 가 E2E_P11STATUS_NONEWDATA 로 변함
설정 영향	없음
ASW 필요 조치 사항	없음

4.3.6 Version 1.2.0

- 신규 기능
 - 8 비트 CRC 지원을 위한 E2E Profile 11 지원

원인	- 8 비트 CRC 지원을 위해 E2E Profile 11 지원
동작 영향	없음
설정 영향	없음
ASW 필요 조치 사항	없음

➤ 개선 사항

- 없음

4.3.7 Version 1.1.5

➤ 신규 기능

- 16 비트 CRC 지원을 위한 E2E Profile 5, E2E Profile 6, E2E State Machine 지원

원인	- 16 비트 CRC 지원을 위해 E2E Profile 5, E2E Profile 6, E2E State Machine 지원
동작 영향	없음
설정 영향	없음
ASW 필요 조치 사항	없음

➤ 개선 사항

- 없음

4.3.8 Version 1.1.4

➤ 신규 기능

- 기능 안전 적용에 대응하여 관련 산출물 개발

원인	- Safety Manual (E2E_Safety_Manual.pdf) 개발
동작 영향	없음
설정 영향	없음
ASW 필요 조치 사항	없음

➤ 개선 사항

- 없음

4.3.9 Version 1.1.3

➤ 신규 기능

- 없음

➤ 개선 사항

- 사양, 코딩 룰 등에 따른 코드 정리

원인	<ul style="list-style-type: none"> - E2E 내부 함수의 Argument 이름을 코딩 룰에 맞게 변경 (Prefix 제거) - AUTOSAR 사양에 따라 E2E API의 Argument 및 지역 변수에 대해 Memory Class 를 E2E_VAR에서 AUTOMATIC으로 변경 - Compile 속도 개선을 위해 MemMap 관련 코드 사용 축소
동작 영향	없음
설정 영향	없음
ASW 필요 조치 사항	없음

4.3.10 Version 1.1.2

➤ 신규 기능

- 없음

➤ 개선 사항

- API의 Return Type 수정

원인	API의 Return Type이 열거형 타입을 자체 생성 사용 (AUTOSAR 사양에서는 Std_ReturnType을 사용)
동작 영향	없음
설정 영향	없음
ASW 필요 조치 사항	Return Type을 얻기 위한 변수의 타입을 Std_ReturnType으로 변경

4.3.11 Version 1.1.1

➤ 신규 기능

- 없음

➤ 개선 사항

- Argument에 대한 검증 기능 강화

원인	API의 Argument에 대해 사용자의 오입력 시 동작 오류 발생 가능
동작 영향	없음
설정 영향	없음
ASW 필요 조치 사항	없음

4.3.12 Version 1.1.0

최초 릴리즈

4.3.13 Version 1.0.0

초기 버전 (입수)

4.4 Limitations

4.4.1 E2E Library

➤ Data ID 설정 시 주의사항

※ 이는 “Specification of SW-C End-To-End Communication Protection Library” [1]의 4.1 에 기술된 사항으로 사용자는 해당 내용을 반드시 확인하여야 한다. 또한, 해당 사항과 관련된 자세한 내용은 같은 문서의 11.5 를 참조한다.

- E2E Profile 1 은 Implicit 방식의 2 바이트 Data ID 를 사용하며, CRC8 계산에 이용된다. 서로 다른 2 개의 2 바이트 수에 대한 CRC 는 같은 결과를 가져올 수 있으므로, 사용자의 주의가 필요하다. “Specification of SW-C End-To-End Communication Protection Library” [1]의 E2EUSE072, E2EUSE073 을 참조한다.
- E2E Profile 2 는 Counter 값에 따라 Data ID 리스트에서 선택된 Implicit 방식의 1 바이트 Data ID 를 사용하며, 이는 CRC 계산에 이용된다. E2E Profile 2 의 Data ID 리스트 사용 및 생성에 관한 자세한 내용은 “Specification of SW-C End-To-End Communication Protection Library” [1]의 Chapter 13 을 참조한다.

➤ Data Element 레벨에서의 라이브러리 호출 시 제약 사항

※ 이는 “Specification of SW-C End-To-End Communication Protection Library” [1]의 4.1.1 에 기술된 내용으로 사용자는 해당 내용을 반드시 확인하여야 한다.

- E2E Library 가 Data Element 레벨(예: SW-C 로부터 또는 E2E Protection Wrapper 로부터)에서 호출되면, 해당 통신은 1:1 또는 1:N Explicit Sender-Receiver Communication 이어야 한다. 다시 말해, E2E Library 가 Data Element 레벨에서 호출되면, N:1 방식, Implicit 통신, 그리고, 나머지 통신 모델 (특히 Client-Server 모델)은 지원하지 않는다. [E2E0224]
- E2E Library 가 Data Element 레벨 그리고 1:N 통신 모델에서 호출될 때, 해당 Data Element 가 둘 이상의 I-PDU 를 통해 전송되면, 모든 I-PDU 는 같은 레이어아웃이어야 한다. [E2E0255]
- E2E Library 가 Data Element 레벨에서 호출되면, 해당 Data Element 는 intra-ECU 통신 (COM 의 개입이 없는)이거나 Com 의 I-PDU 에 매핑되어야 한다. 하지만, 동시에 양쪽 모두에 매핑될 수 없다. [E2E0226]

➤ I-PDU 레벨에서 라이브러리 호출 시 제약 사항

- E2E Library 가 I-PDU 레벨에서 호출되면, 해당 I-PDU 에 대해 오직 하나의 세팅만 있어야 한다. 이는 I-PDU 내에서 여러 Signal Group 들에 대해 Callout 을 통한 E2E Protection 은

불가능하다는 것을 의미한다. [E2E0269]

4.4.2 Usage of E2E Library (Rte)

- COM E2E Callouts 관련 API 의 인터페이스
 - ComIPduCallout 관련 AUTOSAR 사양 4.2.2 버전 적용하여, IPDU_E2EProtect_<IPDU ID>(), IPDU_E2ECheck_<IPDU ID> () API 들의 인터페이스 변경 [E2EUSE0250], [E2EUSE0251]
- PDU 레벨에서 라이브러리 호출 시 제약 사항
 - ComIPduCallout 관련 AUTOSAR 사양 4.2.2 버전 적용하여, IPDU_E2EProtect_<IPDU ID>(), IPDU_E2ECheck_<IPDU ID> () API 들의 인터페이스 변경 [E2EUSE0250], [E2EUSE0251]

4.5 Deviations

4.5.1 E2E Library

- 16 비트 CRC 지원
 - 4.0.3 에서 제공하지 않는 16 비트 CRC 지원을 위해 E2E Profile 5, E2E Profile 6, E2E State Machine 를 지원한다.
- 관련 AUTOSAR 사양을 4.3.0 을 참조한다.

4.5.2 Usage of E2E Library (Rte)

- COM E2E Callouts 관련 API 의 인터페이스
 - ComIPduCallout 관련 AUTOSAR 사양 4.2.2 버전 적용하여, IPDU_E2EProtect_<IPDU ID>(), IPDU_E2ECheck_<IPDU ID> () API 들의 인터페이스 변경 [E2EUSE0250], [E2EUSE0251]
- E2E Protection Wrapper 의 Redundant API 의 인터페이스
 - Redundant 사용 시 Init API 관련 4.2.2 버전 적용하여, E2EPW_WriteInit1, E2EPW_WriteInit2, E2EPW_ReadInit1, E2EPW_ReadInit2 를 추가 [Chapter 12.1.7.2]

5. Configuration Guide

5.1 E2E

없음

5.2 Crc

5.2.1 CrcGeneral Container

Parameter Name

Value

Category

Parameter Name	Value	Category
Crc8Mode ¹⁾	CRC_8_RUNTIME/ CRC_8_TABLE/ CRC_8_HARDWARE	C
Crc8H2FMode ¹⁾	CRC_8H2F_RUNTIME/ CRC_8H2F_TABLE/ CRC_8H2F_HARDWARE	C

1) E2E Profile 1,2,11 을 사용 할 경우, Crc8Mode 와 Crc8H2FMode 는 반드시 설정되어야 한다.

설정 항목에 대한 자세한 정보는 Crc 모듈의 User Manual 을 참조한다.

5.3 Rte

5.3.1 RteGeneration Container

Parameter Name	Value	Category
RteEndToEndProtectionWrapperMode ¹⁾	SINGLE/REDUNDANT	C

1) E2E Protection Wrapper 기능을 사용할 경우,

Wrapper function 을 Single channel function(Write, Read)으로 출력할 지

Redundant function (Write1, Write2, Read1, Read2)으로 출력할 지 결정한다.

5.4 Software Component Template

5.4.1 EndToEndProtectionSet Container

E2E Protection Wrapper 또는 Com E2E Callouts 의 기능을 사용할 경우

ARPackage 에 EndToEndProtectionSet 컨테이너를 추가하여 관련 설정을 할 수 있다.

EndToEndProtectionSet 컨테이너는 EndToEndProtection 컨테이너들을 포함한다.

5.4.2 EndToEndProtection Container

E2E Protection Wrapper 또는 Com E2E Callouts 를 사용할 경우

EndToEndProtectionSet 컨테이너 내에 EndToEndProtection 컨테이너를 추가하여

개별적인 E2E Protection 관련 설정을 할 수 있다.

EndToEndProtection 컨테이너는 EndToEndProfile, EndToEndProtectionVariableDataPrototype,

EndToEndProtectionISignalIPdu 컨테이너들을 포함한다.

5.4.3 EndToEndProfile Container

E2E Protection Wrapper 또는 Com E2E Callouts 를 사용할 경우,

EndToEndProtection 컨테이너 내에 EndToEndProfile 을 추가하여 관련 설정을 할 수 있다.

Parameter Name	Value	Category
Category ¹⁾	PROFILE_01/PROFILE_02	C
DataIds ²⁾	User Defined (> 0)	C

Parameter Name	Value	Category
DataIDMode ³⁾	0 / 1 / 2	C
DataLength ⁴⁾	User Defined (> 0)	C
MaxDeltaCounterInit ⁵⁾	User Defined (> 0)	C
CrcOffset ⁶⁾	User Defined (> 0)	C
CounterOffset ⁷⁾	User Defined (> 0)	C

- 1) 사용할 Profile 을 선택한다.
- 2) E2E_P01ConfigType 의 DatID 또는 E2E_P02ConfigType 의 DataIDList 를 설정한다.
- 3) E2E_P01ConfigType 의 DataIDMode 를 설정한다. Value 에 대한 설정 값은 다음과 같다.
0: E2E_P01_DATAID_BOTH, 1: E2E_P01_DATAID_ALT, 2: E2E_P01_DATAID_LOW
- 4) E2E_P01ConfigType 또는 E2E_P02ConfigType 의 DataLength 를 설정한다.
- 5) E2E_P01ConfigType 또는 E2E_P02ConfigType 의 MaxDeltaCounterInit 를 설정한다.
- 6) E2E_P01ConfigType 의 CRCOffset 을 설정한다.
- 7) E2E_P01ConfigType 의 CounterOffset 을 설정한다.

5.4.4 EndToEndProtectionVariablePrototype Container

E2E Protection Wrapper 기능을 사용하는 경우, EndToEndProtectionVariablePrototype 을 설정한다.
EndToEndProtectionVariablePrototype 에는 하나의 Sender 와 여러 Receiver 컨테이너를 설정할 수 있다.
Sender 와 Receiver 는 모두 VariableDataPrototypeInSystemInstanceRef 컨테이너이다.

5.4.5 VariableDataPrototypeInSystemInstanceRef Container

Parameter Name	Value	Category
ContextCompositionRef ¹⁾	RootSwCompositionPrototype	C
ContextComponentRef ²⁾	SwComponentPrototype	C
ContextPortRef ³⁾	RPortPrototype	C
TargetDataprototypeRef ⁴⁾	VariableDataPrototype	C

- 1) E2E Protection Wrapper 를 사용할 RootSwCompositionPrototype 을 설정한다.
- 2) E2E Protection Wrapper 를 사용할 Composition 을 설정한다.
- 3) E2E Protection Wrapper 를 사용할 Port 를 설정한다.
- 4) E2E Protection Wrapper 를 사용할 Sender-Receiver Interface 의 Data Element 를 선택한다.
해당 Data Element 는 ContextPortRef 에서 설정한 Port 에서 사용하고 있어야 한다.

5.4.6 EndToEndProtectionISignalIPdu Container

Com E2E Callouts 기능을 사용 하는 경우, EndToEndProtectionISignalIPdu 컨테이너를 설정한다.

Parameter Name	Value	Category
ISignalGroupRef ¹⁾	ISignalGroup	C

Parameter Name	Value	Category
ISignalIPduRef ²⁾	ISignalIPdu	C

- 1) E2E Protection 을 사용할 Signal Group 을 설정한다.
- 2) 1)에서 설정한 Signal Group 이 포함된 I-PDU 를 설정한다.

5.4.7 NonQueuedSenderComSpec/NonQueuedReceiverComSpec

5.4.5 의 ContextPortRef 에 해당하는 Port 는 TargetDataPrototypeRef 에 해당하는 Data Element 에 대해 ProvidedComSpecs 에 NonQueuedSenderComSpec 컨테이너 또는 RequiredComSpecs 에 NonQueuedReceiverComSpec 컨테이너가 설정되어야 하며, NonQueuedSenderComSpec 또는 NonQueuedReceiverComSpec 컨테이너에서 다음의 항목을 설정한다.

Parameter Name	Value	Category
UsesEndToEndProtection ¹⁾	true/false	C

- 1) 해당 Port 에서 EndToEndProtection 의 사용 여부를 결정한다.

5.5 Com

5.5.1 ComIPdu Container

Com E2E Callouts 기능을 사용 하는 경우, 해당 ComIPdu 에 대해 다음과 같이 설정한다.

- Tx (Protect)

Parameter Name	Value	Category
ComIPduCallout	IPDU_E2EProtect_<IPDU ID> ¹⁾	C

- 1) <IPDU ID>는 해당 ComIPdu 의 ComIPduHandleId 값이다.

- Rx (Protect)

Parameter Name	Value	Category
ComIPduCallout	IPDU_E2ECheck_<IPDU ID> ¹⁾	C

- 1) <IPDU ID>는 해당 ComIPdu 의 ComIPduHandleId 값이다.

6. Application Programming Interface (API)

6.1 Type Definitions

6.1.1 E2E Profile 1 types

6.1.1.1 E2E_P01ConfigType

Type:	Structure		
Elements:	uint16	CounterOffset	Counter의 비트 오프셋 (MSB first 순서) 즉, 데이터 내에서 Counter의 시작 위치이다. Variant 1A와 1B의 CounterOffset은 8이다. 해당 값은 4의 배수이어야 한다.
	uint16	CRCOffset	CRC의 비트 오프셋 (MSB first 순서) 즉, 데이터 내에서 CRC의 시작 위치이다. Variant 1A와 1B의 CRCOffset은 0이다. 해당 값은 8의 배수이어야 한다.
	uint16	DataID	위장 (Masquerade)을 방지하기 위한 고유 식별자 ID 값의 선택 관련하여 제약사항이 있으며 해당 내용은 AUTOSAR 사양 (Specification of SW-C End-to-EndProtection Library)의 11.5를 참조한다.
	E2E_P01DataIDMode	DataIDMode	CRC 계산에 사용되는 Data ID의 모드 6.1.1.2 E2E_P01DataIDMode를 참조한다.
	uint16	DataLength	E2E 헤더 (CRC, Counter 등)를 포함한 데이터의 길이 비트 단위이며, 8의 배수이고, 240 이하만 가능하다.
	uint8	MaxDeltaCounterInit	연속으로 수신된 정상 데이터에 대해 Counter 값 차이의 최대 허용치의 초기값 예를 들어, MaxDeltaCounterInit이 1이고, Counter가 1인 카운터를 받는다면, 다음 수신되는 데이터의 카운터가 2, 3은 허용하지만 카운터가 4인 데이터를 수신은 허용하지 않는다. 만약 수신자가 다음 수신에서 새 데이터를 받지 않으면, 수신자는 허용치를 1 증가시킨다.
Description:	E2E Profile 1에 대한 전송 데이터의 설정 (Data Element 또는 I-PDU) 전송 데이터마다 이 typedef의 변수가 존재한다.		

6.1.1.2 E2E_P01DataIDMode

Type:	Enumeration		
Range:	E2E_P01_DATAID_BOTH	0x00	CRC에 두 바이트 모두 포함된다. (Double ID configuration) Variant 1A에서 사용된다.
	E2E_P01_DATAID_ALT	0x01	Counter의 Parity에 따라 상위와 하위 바이트를 번갈아 가며, 두 바이트 중 하나만 포함된다. (Alternation ID Configuration) Counter가 짝수이면 하위 바이트가 포함되고, Counter가 홀수이면, 상위 바이트가 포함된다. Variant 1B에서 사용된다.

	E2E_P01_DATAID_LOW	0x02	하위 바이트만 포함되고, 상위 바이트는 사용되지 않는다. 특정 시스템의 ID 가 8 비트 일 때 적용된다.
Description:	E2E Profile 1 의 Data ID 는 2 바이트이다. 1 바이트 CRC 에 2 바이트의 Data ID 를 어떻게 포함시킬 지 (Implicit)에 대해 세 가지 모드가 존재한다.		

6.1.1.3 E2E_P01SenderStateType

Type:	Structure		
Element:	uint8	Counter	다음 데이터 보호를 위해 사용되는 Counter. 초기값은 0 이다. 즉, 첫 데이터는 Counter 가 0 이다. Counter 로 보호한 이후 0xF 를 법(法)으로 증가한다. (modulo 0xF) 0xF 는 Error 값으로 표현되어, 0xF 는 건너 뛴다. (0xE 다음은 0x0 이다). 상위 4 바이트는 항상 0 이다.
Description:	E2E Profile 1 에 의해 보호되는 데이터에 대한 Sender 의 상태		

6.1.1.4 E2E_P01ReceiverStateType

Type:	Structure		
Elements:	uint8	LastValidCounter	가장 마지막에 받은 Counter 값. 만약 아직 아무 데이터도 받지 않았다면, 0x0 이다. 매 수신 마다 Counter 는 받은 값으로 업데이트된다.
	uint8	MaxDeltaCounter	연속적으로 받은 두 정상 메시지에 대해 Counter 값 차이에 대한 최대 허용치
	boolean	WaitForFirstData	True 일 경우, Receiver 의 초기화/재초기화 이후 (올바른 Data ID 와 CRC 를 가진)아직 정상 데이터가 도착하지 않았음을 의미한다.
	boolean	NewDataAvailable	E2E Library 에 의해 확인되도록 E2E Library 에 새 데이터가 사용 가능함을 알려준다. 이 속성은 E2E Library 가 아닌 E2E Library 의 Caller 가 설정한다.
	uint8	LostData	마지막 정상 데이터(메시지)의 수신 이후 유실된 데이터(메시지)의 수 이 속성은 Status 가 E2E_P01STATUS_OK 또는 E2E_P01STATUS_OKSOMELOST 일 때만 설정된 다. 이외의 Status 값에 대해서는 LostData 값은 Undefined 이다.
	E2E_P01ReceiverStatusType	Status	Check 함수에 의해 결정되는 Data 의 검증 결과 6.1.1.5 E2E_P01ReceiverStatusType 를 참조한다.
Description:	E2E Profile 1 에 의해 보호되는 데이터에 대한 Receiver 의 상태		

6.1.1.5 E2E_P01ReceiverStatusType

Type:	Enumeration
-------	-------------

Range:	E2E_P01STATUS_OK	0x0	OK: 통신 매체로부터 새 데이터를 수신하였고, CRC 가 정확하며, Counter 는 Status 가 _INITIAL/_OK/_OKSOMELOST 중 하나인 마지막 수신한 데이터로부터 1 만큼 증가되었다. 즉, 마지막 정상 데이터 수신 후 데이터 유실이 발생하지 않았다.
	E2E_P01STATUS_N ONEWDATA	0x1	Error: Check 함수가 호출되었지만, 마지막 호출 이후 통신 매체(예: RTE, COM)에 따르면 새 데이터를 사용할 수 없는 상태이다. 그 결과 데이터의 E2E Check 는 실행되지 않았다.
	E2E_P01STATUS_W RONGCRC	0x2	Error: 통신 매체로부터 데이터를 수신하였으나 CRC 가 부정확하다.
	E2E_P01STATUS_INI TIAL	0x4	Error: 통신 매체로부터 새 데이터를 수신하였고, CRC 가 정확하지만, Receiver 가 초기화/재초기화한 이후 첫 번째 데이터이다. 따라서 Counter 는 아직 검증되지 않았다.
	E2E_P01STATUS_RE PEATED	0x8	Error: 통신 매체로부터 새로운 데이터를 수신하였고, CRC 가 정확하지만, Counter 가 Status 가 _INITIAL/_OK/_OKSOMELOST 중 하나인 마지막 수신한 데이터와 동일하다.
	E2E_P01STATUS_OK SOMELOST	0x20	OK: 통신 매체로부터 새로운 데이터를 수신하였고, CRC 가 정확하지만, Counter 가 Status 가 _INITIAL/_OK/_OKSOMELOST 중 하나인 마지막 수신한 데이터의 Counter 보다 DeltaCounter 만큼 증가하였다. (1 < DeltaCounter ≤ MaxDeltaCounter) 즉, 마지막 정상/최초 수신 이후, 순서 상 몇몇 데이터가 유실되었을 가능성이 있다. 하지만, 이 것은 설정한 허용치 내에 있다.
	E2E_P01STATUS_W RONGSEQUENCE	0x40	Error: 통신 매체로부터 새로운 데이터를 수신하였고, CRC 가 정확하지만, Counter 가 Status 가 _INITIAL/_OK/_OKSOMELOST 중 하나인 마지막 수신한 데이터의 Counter 에 대해 Delta 값이 너무 크다. (DeltaCounter > MaxDeltaCounter) 즉, 마지막 정상/최초 수신 이후, 순서 상 너무 많은 데이터가 유실되었을 가능성이 있다.
Description:	E2E Profile 1 의 Check 함수에 의해 결정된 데이터 검증 결과		

6.1.2 E2E Profile 2 types

6.1.2.1 E2E_P02ConfigType

Type:	Structure		
Elements:	uint16	DataLength	E2E 헤더 (CRC, Counter 등)를 포함한 데이터의 길이 비트 단위이며, 8 의 배수이다.
	uint8[16]	DataIDList	위장 (Masquerade)을 방지하기 위해 적절하게 선택된 Data ID 들의 배열

	uint8	MaxDeltaCounterInit	<p>연속으로 수신된 정상 데이터에 대해 Counter 값 차이의 최대 허용치의 초기값 예를 들어, MaxDeltaCounterInit 이 1 이고, Counter 가 1 인 카운터를 받는다면, 다음 수신되는 데이터의 카운터가 2, 3 은 허용하지만 카운터가 4 인 데이터를 수신은 허용하지 않는다.</p> <p>만약 수신자가 다음 수신에서 새 데이터를 받지 않으면, 수신자는 허용치를 1 증가시킨다.</p>
Description:	E2E Profile 2 에 대해 RTE port 로 보내는 Data Element 의 수정 불가능한 설정 Profile 2 에서는 Counter 와 CRC 의 위치는 설정 불가능하다.		

6.1.2.2 E2E_P02SenderStateType

Type:	Structure		
Element:	uint8	Counter	<p>다음 데이터를 보호하기 위해 사용되는 Counter. 초기값은 0 이다. 즉, 첫 데이터는 Counter 가 0 이다. Counter 로 보호한 이후 16 을 법(法)으로 증가한다. (modulo 16)</p>
Description:	E2E Profile 2 에 의해 보호되는 데이터에 대한 Sender 의 상태		

6.1.2.3 E2E_P02ReceiverStateType

Type:	Structure		
Elements:	uint8	LastValidCounter	가장 마지막에 받은 Counter 값.
	uint8	MaxDeltaCounter	연속적으로 받은 두 정상 메시지에 대해 Counter 값 차이에 대한 최대 허용치
	boolean	WaitForFirstData	True 일 경우, Receiver 의 초기화/재초기화 이후 (올바른 Data ID 와 CRC 를 가진)아직 정상 데이터가 도착하지 않았음을 의미한다.
	boolean	NewDataAvailable	E2E Library 에 의해 확인되도록 E2E Library 에 새 데이터가 사용 가능함을 알려준다. 이 속성은 E2E Library 가 아닌 E2E Library 의 Caller 가 설정한다.
	uint8	LostData	마지막 정상 데이터(메시지)의 수신 이후 유실된 데이터(메시지)의 수
	E2E_P02ReceiverStatusType	Status	Check 함수에 의해 결정되는 Data 의 검증 결과 6.1.2.4 E2E_P02ReceiverStatusType 를 참조한다.
Description:	E2E Profile 2 에 의해 보호되는 데이터에 대한 Receiver 의 상태		

6.1.2.4 E2E_P02ReceiverStatusType

Type:	Enumeration		
Range:	E2E_P02STATUS_OK	0x0	<p>OK: 통신 매체로부터 새 데이터를 수신하였고, CRC 가 정확하며, Counter 는 Status 가 _INITIAL/_OK/_OKSOMELOST 중 하나인 마지막 수신한 데이터로부터 1 만큼 증가되었다. 즉, 마지막 정상 데이터 수신 후 데이터 유실이 발생하지 않았다.</p>

	E2E_P02STATUS_N ONEWDATA	0x1	Error: Check 함수가 호출되었지만, 마지막 호출 이후 통신 매체(예: RTE, COM)에 따르면 새 데이터를 사용할 수 없는 상태이다. 그 결과 데이터의 E2E Check 는 실행되지 않았다.
	E2E_P02STATUS_W RONGCRC	0x2	Error: 통신 매체로부터 데이터를 수신하였으나 CRC 가 부정확하다.
	E2E_P02STATUS_INI TIAL	0x4	Error: 통신 매체로부터 새 데이터를 수신하였고, CRC 가 정확하지만, Receiver 가 초기화/재초기화한 이후 첫 번째 데이터이다. 따라서 Counter 는 아직 검증되지 않았다.
	E2E_P02STATUS_RE PEATED	0x8	Error: 통신 매체로부터 새로운 데이터를 수신하였고, CRC 가 정확하지만, Counter 가 Status 가 _INITIAL/_OK/_OKSOMELOST 중 하나인 마지막 수신한 데이터와 동일하다.
	E2E_P02STATUS_OK SOMELOST	0x20	OK: 통신 매체로부터 새로운 데이터를 수신하였고, CRC 가 정확하지만, Counter 가 Status 가 _INITIAL/_OK/_OKSOMELOST 중 하나인 마지막 수신한 데이터의 Counter 보다 DeltaCounter 만큼 증가하였다. ($1 < \text{DeltaCounter} \leq \text{MaxDeltaCounter}$) 즉, 마지막 정상/최초 수신 이후, 순서 상 몇몇 데이터가 유실되었을 가능성이 있다. 하지만, 이 것은 설정한 허용치 내에 있다.
	E2E_P02STATUS_W RONGSEQUENCE	0x40	Error: 통신 매체로부터 새로운 데이터를 수신하였고, CRC 가 정확하지만, Counter 가 Status 가 _INITIAL/_OK/_OKSOMELOST 중 하나인 마지막 수신한 데이터의 Counter 에 대해 Delta 값이 너무 크다. ($\text{DeltaCounter} > \text{MaxDeltaCounter}$) 즉, 마지막 정상/최초 수신 이후, 순서 상 너무 많은 데이터가 유실되었을 가능성이 있 다.
Description:	E2E Profile 2 의 Check 함수에 의해 결정된 데이터 검증 결과		

6.1.3 E2E Profile 5 types

6.1.3.1 E2E_P05ConfigType

Type:	Structure		
Elements:	uint16	Offset	데이터의 시작으로부터 E2E header 의 첫번째 비트의 비트 오프셋. (bit numbering: bit 0 이 lsb 이다.). Offset 은 8 의 배수이며, 값의 범위는 $0 \leq \text{Offset} \leq \text{DataLength} - (3*8)$ 이다. 예: Offset 이 8 이면, CRC 의 Low Byte 는 Byte 1 에 High Byte 는 Byte 2 에 위치한다.
	uint16	DataLength	데이터의 길이 (비트) 값의 범위는 $(3*8) \leq \text{DataLength} \leq 4096*8$ (4kB)
	uint16	DataID	시스템 전체에서 유일한 식별자

	uint8	MaxDeltaCounter	연속으로 수신된 정상 데이터에 대해 Counter 값 차이의 최대 허용치 예: Counter 가 1 이고, MaxDeltaCounter 가 3 이면, 다음 수신 가능한 데이터의 카운터 값은 2, 3, 4 이다.
Description:	E2E Profile 5 에 대해 전송 데이터 (Data Element or I-PDU)의 설정. 각각의 전송 데이터에 대해 이 typedef 의 Instance 가 존재한다.		

6.1.3.2 E2E_P05ProtectStateType

Type:	Structure		
Element:	uint8	Counter	다음 데이터를 보호하기 위해 사용되는 Counter. 초기값은 0 이다. 즉, 첫 데이터는 Counter 가 0 이다. E2E_P05Protect()가 호출될 때마다 0xFF 까지 증가한다.
Description:	E2E Profile 5 에 의해 보호되는 데이터에 대한 Sender 의 상태		

6.1.3.3 E2E_P05CheckStateType

Type:	Structure		
Elements:	E2E_P02CheckStatusType	Status	Check 함수에 의해 결정되는 현 사이클에서 데이터 검증 결과
	uint8	Counter	이전 사이클의 데이터 Counter
Description:	E2E Profile 5 에 의해 보호되는 하나의 싱글 데이터에 대한 수신 상태		

6.1.3.4 E2E_P05CheckStatusType

Type:	Enumeration		
Range:	E2E_P05STATUS_OK	0x0	OK: 통신 매체로부터 새 데이터를 수신하였고, CRC 가 정확하며, Counter 는 Status 가 _INITIAL/_OK/_OKSOMELOST 중 하나인 마지막 수신한 데이터로부터 1 만큼 증가되었다. 즉, 마지막 정상 데이터 수신 후 데이터 유실이 발생하지 않았다.
	E2E_P05STATUS_NEWDATA	0x1	Error: Check 함수가 호출되었지만, 마지막 호출 이후 통신 매체(예: RTE, COM)에 따르면 새 데이터를 사용할 수 없는 상태이다. 그 결과 데이터의 E2E Check 는 실행되지 않았다.
	E2E_P05STATUS_ERROR	0x7	Error:
	E2E_P05STATUS_REPEATED	0x8	Error: 통신 매체로부터 새로운 데이터를 수신하였고, CRC 가 정확하지만, Counter 가 Status 가 _INITIAL/_OK/_OKSOMELOST 중 하나인 마지막 수신한 데이터와 동일하다.

	E2E_P05STATUS_OK SOMELOST	0x20	OK: 통신 매체로부터 새로운 데이터를 수신하였고, CRC 가 정확하지만, Counter 가 Status 가 _INITIAL/_OK/_OKSOMELOST 중 하나인 마지막 수신한 데이터의 Counter 보다 DeltaCounter 만큼 증가하였다. ($1 < \text{DeltaCounter} \leq \text{MaxDeltaCounter}$) 즉, 마지막 정상/최초 수신 이후, 순서 상 몇몇 데이터가 유실되었을 가능성이 있다. 하지만, 이 것은 설정한 허용치 내에 있다.
	E2E_P05STATUS_W RONGSEQUENCE	0x40	Error: 통신 매체로부터 새로운 데이터를 수신하였고, CRC 가 정확하지만, Counter 가 Status 가 _INITIAL/_OK/_OKSOMELOST 중 하나인 마지막 수신한 데이터의 Counter 에 대해 Delta 값이 너무 크다. ($\text{DeltaCounter} > \text{MaxDeltaCounter}$) 즉, 마지막 정상/최초 수신 이후, 순서 상 너무 많은 데이터가 유실되었을 가능성이 있 다.
Description:	E2E Profile 5 에 의해 보호되는 하나의 사이클에서 하나의 싱글 데이터에 대한 수신 상황		

6.1.4 E2E Profile 6 types

6.1.4.1 E2E_P06ConfigType

Type:	Structure		
Elements:	uint16	Offset	데이터의 시작으로부터 E2E header 의 첫번째 비트의 비트 오프셋. (bit numbering: bit 0 이 lsb 이다.). Offset 은 8 의 배수이며, 값의 범위는 $0 \leq \text{Offset} \leq \text{MaxDataLength} - (5*8)$ 이다. 예: Offset 이 8 이면, CRC 의 High Byte 는 Byte 1 에 Low Byte 는 Byte 2 에 위치한다.
	uint16	MinDataLength	데이터의 최소 길이 (비트) 값의 범위는 $\text{MinDataLength} \leq \text{Length}, 5*8 \leq \text{MinDataLength} \leq 4096*8$
	uint16	MaxDataLength	데이터의 최대 길이 (비트) 값의 범위는 $\text{DataLength} \leq \text{MaxDataLength}$ $\text{MinDataLength} \leq \text{MaxDataLength} \leq 4096*8$
	uint16	DataID	시스템 전체에서 유일한 식별자
	uint8	MaxDeltaCounter	연속으로 수신된 정상 데이터에 대해 Counter 값 차이의 최대 허용치 예: Counter 가 1 이고, MaxDeltaCounter 가 3 이 면, 다음 수신 가능한 데이터의 카운터 값은 2, 3, 4 이다.
Description:	E2E Profile 6 에 대해 전송 데이터 (Data Element or I-PDU)의 설정. 각각의 전송 데이터에 대해 이 typedef 의 Instance 가 존재한다.		

6.1.4.2 E2E_P06ProtectStateType

Type:	Structure
-------	-----------

Element:	uint8	Counter	다음 데이터를 보호하기 위해 사용되는 Counter. 초기값은 0이다. 즉, 첫 데이터는 Counter가 0이다. E2E_P06Protect()가 호출될 때마다 0xFF 까지 증가한다. 0xFF 까지 증가한 후, 다음 값은 0x00 부터 시작한다. Overflow는 caller에게 전달되지 않는다.
Description:	E2E Profile 6에 의해 보호되는 데이터에 대한 Sender의 상태		

6.1.4.3 E2E_P06CheckStateType

Type:	Structure		
Elements:	E2E_P06CheckStatusType	Status	Check 함수에 의해 결정되는 현 사이클에서 데이터 검증 결과
	uint8	Counter	이전 사이클의 데이터 Counter
Description:	E2E Profile 6에 의해 보호되는 하나의 싱글 데이터에 대한 수신 상태		

6.1.4.4 E2E_P06CheckStatusType

Type:	Enumeration		
Range:	E2E_P06STATUS_OK	0x0	OK: 해당 Cycle 내에서 Data check가 성공적이었다 (Counter가 1만큼 증가하였다).
	E2E_P06STATUS_NONEWDATA	0x1	Error: Check 함수가 호출되었지만, 마지막 호출 이후 통신 매체(예: RTE, COM)에 따르면 새 데이터를 사용할 수 없는 상태이다. 그 결과 데이터의 E2E Check는 실행되지 않았다.
	E2E_P06STATUS_ERROR	0x7	Error: counter와 관련 없는 error가 발생하였다. (예: 잘못된 CRC, 잘못된 Length)
	E2E_P06STATUS_REPEATED	0x8	Error: 해당 Cycle 내에서 Data check가 성공적이었지만, repetition 예외가 발생하였다.
	E2E_P06STATUS_OK_SOMELOST	0x20	OK: 해당 Cycle 내에서 Data check가 성공적이었다 (counter가 허용된 delta 범위 내에서 증가하였다).
	E2E_P06STATUS_WRONGSEQUENCE	0x40	Error: 해당 Cycle 내에서 Data check가 성공적이었지만, counter jump가 발생했다. Counter jump : counter 허용된 delta 범위 외로 변경된 경우
Description:	E2E Profile 6에 의해 보호되는 하나의 사이클에서 하나의 싱글 데이터에 대한 수신 상황		

6.1.5 E2E Profile 11 types

6.1.5.1 E2E_P11DataIDMode

Type:	Enumeration		
Range:	E2E_P11_DATAID_BOTH	0x00	CRC에 두 바이트 모두 포함된다. (Double ID configuration) Variant 11A에서 사용된다.
	E2E_P11_DATAID_NIBBLE	0x03	하위 바이트는 CRC 계산에 사용되고, 상위 바이트의 하위 Nibble은 헤더에 포함된다. Variant 11C에서 사용된다.
Description:	E2E Profile 11의 Data ID는 2 바이트이다. 1 바이트 CRC에 2 바이트의 Data ID를 어떻게 포함시킬 지에 대해 두 가지 모드가 존재한다.		

6.1.5.2 E2E_P11ConfigType

Type:	Structure		
Elements:	uint16	DataLength	데이터의 길이(비트). 8 의 배수여야 하며 240 이하여야 함
	uint16	DataID	시스템 전체에서 유일한 식별자
	uint8	MaxDeltaCounter	연속으로 수신된 정상 데이터에 대해 Counter 값 차이의 최대 허용치 예: Counter 가 1 이고, MaxDeltaCounter 가 3 이면, 다음 수신 가능한 데이터의 카운터 값은 2, 3, 4 이다.
	E2E_P11DataIDMode	DataIDMode	Data ID 의 모드. 6.1.5.1 의 E2E_P11DataIDMode 를 참조한다.
	uint16	CRCOffset	데이터에서 CRC 의 시작 위치
	uint16	CounterOffset	데이터에서 Counter 의 시작 위치
	uint16	DataIDNibbleOffset	데이터에서 Data ID nibble 의 시작 위치
Description:	E2E Profile 11 에 대해 전송 데이터 (Data Element or I-PDU)의 설정. 각각의 전송 데이터에 대해 이 typedef 의 Instance 가 존재한다.		

6.1.5.3 E2E_P11ProtectStateType

Type:	Structure		
Element:	uint8	Counter	다음 데이터를 보호하기 위해 사용되는 Counter. 초기값은 0 이다. 즉, 첫 데이터는 Counter 가 0 이다. E2E_P11Protect()가 호출될 때마다 0xE 까지 증가한다. 0xE 까지 증가한 후, 다음 값은 0x00 부터 시작한다.
Description:	E2E Profile 11 에 의해 보호되는 데이터에 대한 Sender 의 상태		

6.1.5.4 E2E_P11CheckStateType

Type:	Structure		
Elements:	E2E_P11CheckStatusType	Status	Check 함수에 의해 결정되는 현 사이클에서 데이터 검증 결과
	uint8	Counter	이전 사이클의 데이터 Counter
Description:	E2E Profile 11 에 의해 보호되는 하나의 싱글 데이터에 대한 수신 상태		

6.1.5.5 E2E_P11CheckStatusType

Type:	Enumeration		
Range:	E2E_P11STATUS_OK	0x0	OK: 해당 Cycle 내에서 Data check 가 성공적이었다 (Counter 가 1 만큼 증가하였다).
	E2E_P11STATUS_NEWDATA	0x1	Error: Check 함수가 호출되었지만, 마지막 호출 이후 통신 매체(예: RTE, COM)에 따르면 새 데이터를 사용할 수 없는 상태이다. 그 결과 데이터의 E2E Check 는 실행되지 않았다.
	E2E_P11STATUS_ERROR	0x7	Error: counter 와 관련 없는 error 가 발생하였다. (예: 잘못된 CRC, 잘못된 Length)

	E2E_P11STATUS_REPEATED	0x8	Error: 해당 Cycle 내에서 Data check 가 성공적이었지만, repetition 예외가 발생하였다.
	E2E_P11STATUS_OKSOMELOST	0x20	OK: 해당 Cycle 내에서 Data check 가 성공적이었다 (counter 가 허용된 delta 범위 내에서 증가하였다).
	E2E_P11STATUS_WRONGSEQUENCE	0x40	Error: 해당 Cycle 내에서 Data check 가 성공적이었지만, counter jump 가 발생했다. Counter jump : counter 허용된 delta 범위 외로 변경된 경우
Description:	E2E Profile 11 에 의해 보호되는 하나의 사이클에서 하나의 싱글 데이터에 대한 수신 상황		

6.1.6 E2E State Machine types

6.1.6.1 E2E_PCheckStatusType

Type:	Structure		
Range:	E2E_P_OK	0x00	OK: 해당 Cycle 내에서 Data check 가 성공적이었다
	E2E_P_REPEATED	0x01	Counter 값이 중복되었다
	E2E_P_WRONGSEQUENCE	0x02	해당 Cycle 내에서 Data check 가 성공적이었지만, counter jump 가 발생했다. Counter jump : counter 허용된 delta 범위 외로 변경된 경우
	E2E_P_ERROR	0x03	Error: counter 와 관련 없는 error 가 발생하였다. (예: 잘못된 CRC, 잘못된 Length)
	E2E_P_NOTAVAILABLE	0x04	어떠한 값도 아직 전달되지 않았다(초기화 할 동안). 버퍼를 초기화 하기 위해 사용되는 값이고, E2E profile 에서 오는 값은 아니다.
	E2E_P_NONEWDATA	0x05	새로운 데이터에 접근할 수 없다
Description:	Profile 버전에 상관없이, 한 cycle 에 한하여 data 송/수신 결과 상태를 나타냄		

6.1.6.2 E2E_SMConfigType

Type:	Structure		
Elements:	uint8	WindowSize	State machine 의 monitoring window 크기
	uint8	MinOkStateInit	가장 최근의 WindowSize checks 에서 E2E_SM_VALID 로의 상태변화를 위해 요구되는 E2E_P_OK 최소 개수(E2E_SM_INIT 상태의 한함)
	uint8	MaxErrorStateInit	가장 최근의 WindowSize checks 에서 E2E_SM_VALID 로의 상태변화를 위해 요구되는 E2E_P_ERROR 의 최대허용 개수(E2E_SM_INIT 상태의 한함)
	uint8	MinOkStateValid	가장 최근의 WindowSize checks 에서 E2E_SM_VALID 상태 유지를 위해 요구되는 E2E_P_OK 최소 개수(E2E_SM_VALID 상태의 한함)
	uint8	MaxErrorStateValid	가장 최근의 WindowSize checks 에서 E2E_SM_VALID 상태 유지를 위해 요구되는 E2E_P_ERROR 의 최대허용 개수(E2E_SM_VALID 상태의 한함)

	uint8	MinOkStateInvalid	가장 최근의 WindowSize checks 에서 E2E_SM_VALID 로의 상태변화를 위해 요구되는 E2E_P_OK 최소 개수(E2E_SM_INVALID 상태의 한함)
	uint8	MaxErrorStateInvalid	가장 최근의 WindowSize checks 에서 E2E_SM_VALID 로의 상태변화를 위해 요구되는 E2E_P_ERROR 의 최대허용 개수(E2E_SM_INVALID 상태의 한함)
Description:	Data exchange 를 위한 Channel 에 대한 configuration 정보		

6.1.6.3 E2E_SMCheckStateType

Type:	Structure		
Elements:	uint8*	ProfileStatusWindow	수행된 E2E Check 의 결과 상태 값들이 저장된 배열에 대한 포인터. 배열의 크기는 WindowSize
	uint8	WindowTopIndex	다음 ProfileStatus 가 쓰여질 배열 index
	uint8	OkCount	가장 최근의 WindowSize check 를 통해 얻어진 E2E_P_OK 개수
	uint8	ErrorCount	가장 최근의 WindowSize check 를 통해 얻어진 E2E_P_ERROR 개수
	E2E_SMStateType	SMState	State machine 의 현재 상태.
Description:	Communication channel 의 protection 상태		

6.1.6.4 E2E_SMStateType

Type:	Structure		
Range:	E2E_SM_VALID	0x00	E2E 에 따라 communication 이 잘 동작하고, 데이터를 사용해도 된다
	E2E_SM_DEINIT	0x01	E2E_SMCheckInit()이 호출되기 전 상태로, 데이터를 사용할 수 없다
	E2E_SM_NODATA	0x02	초기화 때문에 송신자의 데이터를 아무것도 사용할 수 없다
	E2E_SM_INIT	0x03	Start-up 이 끝난 후 데이터가 도착했으나, 아직은 사용할 수 없다
	E2E_SM_INVALID	0x04	Communication 이 잘못 작동하므로, 데이터를 사용할 수 없다
Description:	데이터를 주고받는 channel 의 상태		

6.2 Macro Constants

6.2.1 Error Flags by E2E Library

Type or error status	Relevance	Related code	Value [hex]
최소 하나의 Null 포인터 파라미터	Production	E2E_E_INPUTERR_NULL	0x13

최소 하나의 입력 파라미터 오류 (예: Out of range)	Production	E2E_E_INPUTERR_WRONG	0x17
라이브러리 내부 오류 발생 (예: Error detected by program flow monitoring, Violated invariant, or postcondition)	Production	E2E_E_INTERR	0x19
성공적으로 실행 완료	N/A	E2E_E_OK	0x00
Invalid Value	Production	E2E_E_INVALID	0xFF

6.2.2 Error Flags by E2E Protection Wrapper functions on byte 1 of the return value

Type or error status	Relevance	Related code	Value [hex]
OK: 통신 매체로부터 새 데이터를 수신하였고, CRC 가 정확하며, Counter 는 Status 가 _INITIAL/_OK/_OKSOMELOST 중 하나인 마지막 수신한 데이터로부터 1 만큼 증가되었 다. 즉, 마지막 정상 데이터 수신 후 데이터 유실이 발생하지 않았다.	Production	E2EPW_STATUS_OK	0x0
Error: Check 함수가 호출되었지만, 마지막 호출 이후 통신 매체(예: RTE, COM)에 따르면 새 데이터를 사용할 수 없는 상태이다. 그 결과 데이터의 E2E Check 는 실행되지 않았다.	Production	E2EPW_STATUS_NONEWDATA	0x1
Error: 통신 매체로부터 데이터를 수신하였으나 CRC 가 부정확하다.	Production	E2EPW_STATUS_WRONGCRC	0x2
Error: 통신매체로부터 새 데이터를 수신하였 고, CRC 가 정확하지만, Receiver 가 초기화/재초기화한 이후 첫 번째 데이터이다. 따라서 Counter 는 아직 검증되지 않았다.	Production	E2EPW_STATUS_INITIAL	0x4
Error: 통신 매체로부터 새로운 데이터를 수신하였고, CRC 가 정확하지만, Counter 가 Status 가 _INITIAL/_OK/_OKSOMELOST 중 하나인 마지막 수신한 데이터와 동일하다.	Production	E2EPW_STATUS_REPEATED	0x8
OK: 통신 매체로부터 새 데이터를 수신하였고, CRC 가 정확하지만, Counter 가 Status 가 _INITIAL/_OK/_OKSOMELOST 중 하나인 마지막 수신한 데이터의 Counter 보다 DeltaCounter 만큼 증가하였다. (1 < DeltaCounter ≤ MaxDeltaCounter) 즉, 마지막 정상/최초 수신 이후, 순서상 일부 데이터의 유실 가능성이 있다. 하지만, 이 것은 설정한 허용치 내에 있다.	Production	E2EPW_STATUS_OKSOMELOST	0x20
Error: 통신매체로부터 새 데이터를 수신하였고, CRC 가 정확하지만, Counter 가 Status 가 _INITIAL/_OK/_OKSOMELOST 중	Production	E2EPW_STATUS_WRONGSEQUENC E	0x40

하나인 마지막 수신한 데이터의 Counter 에
대해 Delta 값이 너무 크다.
(DeltaCounter > MaxDeltaCounter)
즉, 마지막 정상/최초 수신 이후,
순서 상 너무 많은 데이터 유실 가능성이 있다.

Type or error status	Relevance	Related code	Value [hex]
확장/확대 (Extension/expansion) 에러 발 생. Bit extension 이 제대로 수행되었는지에 대한 Status 예: I-PDU 내 12 비트가 16 비트 Unit 으로 확장 시, 최상위 4 비트 값들은 0 이어야 한다.	Production	E2EPW_E_DESERIALIZATION	0x3
Write1 과 Write2 로 계산된 컨트롤 필드 (E2E 헤더: Crc/Counter 등)의 값의 불일치 즉, Write1 과 Write2 간의 Voting 실패	Production	E2E_E_REDUNDANCY	0x5

6.3 Functions

6.3.1 E2E Library – E2E Profile 1 routines

6.3.1.1 E2E_P01Protect

Function Name	E2E_P01Protect
Syntax:	Std_ReturnType E2E_P01Protect(CONST E2E_P01ConfigType* Config, E2E_P01SenderStateType* State, uint8* Data)
Service ID	0x01
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	Config: 설정(Static Configuration)에 대한 포인터
Parameters (Inout)	State: 포트/데이터 통신 상태에 대한 포인터 Data: 전송할 데이터에 대한 포인터
Parameters (Out)	None
Return Value	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK
Description	전송할 배열/버퍼를 E2E Profile 1 을 통해 보호한다. Checksum 계산, Counter 와 Data ID 에 대한 처리가 포함된다.

<i>Preconditions</i>	NA
<i>Configuration Dependency</i>	None

6.3.1.2 E2E_P01Check

<i>Function Name</i>	E2E_P01Check
<i>Syntax:</i>	Std_ReturnType E2E_P01Check(E2E_P01ConfigType* Config, E2E_P01ReceiverStateType* State, uint8* Data)
<i>Service ID</i>	0x02
<i>Sync/Async</i>	Synchronous
<i>Reentrancy</i>	Reentrant
<i>Parameters (In)</i>	Config: 설정(Static Configuration)에 대한 포인터 Data: 수신한 데이터에 대한 포인터
<i>Parameters (Inout)</i>	State: 포트/데이터 통신 상태에 대한 포인터
<i>Parameters (Out)</i>	None
<i>Return Value</i>	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK
<i>Description</i>	E2E Profile 1 을 사용하여 수신한 데이터를 검사한다. CRC 계산, Counter 와 Data ID 에 대한 처리가 포함된다.
<i>Preconditions</i>	NA
<i>Configuration Dependency</i>	None

6.3.2 E2E Library – E2E Profile 2 routines

6.3.2.1 E2E_P02Protect

<i>Function Name</i>	E2E_P02Protect
<i>Syntax:</i>	Std_ReturnType E2E_P02Protect(E2E_P02ConfigType* Config, E2E_P02SenderStateType* State, uint8* Data)
<i>Service ID</i>	0x03

<i>Sync/Async</i>	Synchronous
<i>Reentrancy</i>	Reentrant
<i>Parameters (In)</i>	Config: 설정(Static Configuration)에 대한 포인터
<i>Parameters (Inout)</i>	State: 포트/데이터 통신 상태에 대한 포인터 Data: 전송할 데이터에 대한 포인터
<i>Parameters (Out)</i>	None
<i>Return Value</i>	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK
<i>Description</i>	전송할 배열/버퍼를 E2E Profile 2 을 통해 보호한다. Checksum 계산, Counter 와 Data ID 에 대한 처리가 포함된다.
<i>Preconditions</i>	NA
<i>Configuration Dependency</i>	None

6.3.2.2 E2E_P02Check

<i>Function Name</i>	E2E_P02Check
<i>Syntax:</i>	Std_ReturnType E2E_P02Check(E2E_P02ConfigType* Config, E2E_P02ReceiverStateType* State, uint8* Data)
<i>Service ID</i>	0x04
<i>Sync/Async</i>	Synchronous
<i>Reentrancy</i>	Reentrant
<i>Parameters (In)</i>	Config: 설정(Static Configuration)에 대한 포인터 Data: 수신한 데이터에 대한 포인터
<i>Parameters (Inout)</i>	State: 포트/데이터 통신 상태에 대한 포인터
<i>Parameters (Out)</i>	None
<i>Return Value</i>	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK
<i>Description</i>	E2E Profile 2 을 사용하여 수신한 데이터를 검사한다. CRC 계산, Counter 와 Data ID 에 대한 처리가 포함된다.
<i>Preconditions</i>	NA

Configuration Dependency

None

6.3.3 E2E Library – E2E Profile 5 routines

6.3.3.1 E2E_P05Protect

<i>Function Name</i>	E2E_P05Protect
<i>Syntax:</i>	Std_ReturnType E2E_P05Protect(E2E_P05ConfigType* ConfigPtr, E2E_P05ProtectStateType* StatePtr, uint8* DataPtr, uint16 Length)
<i>Service ID</i>	0x26
<i>Sync/Async</i>	Synchronous
<i>Reentrancy</i>	Reentrant
<i>Parameters (In)</i>	ConfigPtr: 설정(Static Configuration)에 대한 포인터 Length: 데이터 길이 (바이트)
<i>Parameters (Inout)</i>	State: 포트/데이터 통신 상태에 대한 포인터 Data: 전송할 데이터에 대한 포인터
<i>Parameters (Out)</i>	None
<i>Return Value</i>	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK
<i>Description</i>	전송할 배열/버퍼를 E2E Profile 5 을 통해 보호한다. Checksum 계산, Counter 에 대한 처리가 포함된다.
<i>Preconditions</i>	E2E_P05ProtectInit()
<i>Configuration Dependency</i>	None

6.3.3.2 E2E_P05ProtectInit

<i>Function Name</i>	E2E_P05ProtectInit
<i>Syntax:</i>	Std_ReturnType E2E_P05ProtectInit(E2E_P05ProtectStateType* StatePtr)
<i>Service ID</i>	0x27
<i>Sync/Async</i>	Synchronous

Reentrancy	Reentrant
Parameters (In)	None
Parameters (Inout)	State: 포트/데이터 통신 상태에 대한 포인터
Parameters (Out)	None
Return Value	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_OK
Description	Protect 상태를 초기화한다.
Preconditions	NA
Configuration Dependency	None

6.3.3.3 E2E_P05Check

Function Name	E2E_P05Check
Syntax:	Std_ReturnType E2E_P05Check(E2E_P05ConfigType* ConfigPtr, E2E_P05CheckStateType* StatePtr, uint8* DataPtr, uint16 Length)
Service ID	0x28
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	ConfigPtr: 설정(Static Configuration)에 대한 포인터 DataPtr: 수신한 데이터에 대한 포인터 Length: 데이터 길이 (바이트)
Parameters (Inout)	State: 포트/데이터 통신 상태에 대한 포인터
Parameters (Out)	None
Return Value	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK
Description	E2E Profile 5 을 사용하여 수신한 데이터를 검사한다. CRC 계산, Counter 에 대한 처리가 포함된다. 한 사이클에 하나의 싱글 데이터만 체크한다. 통신 링크의 축적된 상태를 결정하거나 계산하지 않는다.

<i>Preconditions</i>	E2E_P05CheckInit()
<i>Configuration Dependency</i>	None

6.3.3.4 E2E_P05CheckInit

<i>Function Name</i>	E2E_P05CheckInit
<i>Syntax:</i>	Std_ReturnType E2E_P05CheckInit(E2E_P05CheckStateType* StatePtr)
<i>Service ID</i>	0x29
<i>Sync/Async</i>	Synchronous
<i>Reentrancy</i>	Reentrant
<i>Parameters (In)</i>	None
<i>Parameters (Inout)</i>	State: 포트/데이터 통신 상태에 대한 포인터
<i>Parameters (Out)</i>	None
<i>Return Value</i>	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_OK
<i>Description</i>	Check 상태를 초기화한다.
<i>Preconditions</i>	NA
<i>Configuration Dependency</i>	None

6.3.3.5 E2E_P05MapStatusToSM

<i>Function Name</i>	E2E_P05MapStatusToSM
<i>Syntax:</i>	E2E_PCheckStatusType E2E_P05MapStatusToSM (Std_ReturnType CheckReturn, E2E_P05CheckStatusType Status)
<i>Service ID</i>	0x2a
<i>Sync/Async</i>	Synchronous
<i>Reentrancy</i>	Reentrant
<i>Parameters (In)</i>	CheckReturn: E2E_P05Check 함수의 Return value Status: E2E_P05Check 에 의해 결정된 Status
<i>Parameters (Inout)</i>	None

<i>Parameters (Out)</i>	None
<i>Return Value</i>	E2E_PCheckStatusType: Profile 에 독립적인 Status
<i>Description</i>	Profile 5 의 Check status 를 E2E Status Machine 에 사용될 수 있도록 Generic Check Status 에 매핑한다.
<i>Preconditions</i>	NA
<i>Configuration Dependency</i>	None

6.3.4 E2E Library – E2E Profile 6 routines

6.3.4.1 E2E_P06Protect

<i>Function Name</i>	E2E_P06Protect
<i>Syntax:</i>	Std_ReturnType E2E_P06Protect(const E2E_P06ConfigType* ConfigPtr, E2E_P06ProtectStateType* StatePtr, uint8* DataPtr, uint16 Length)
<i>Service ID</i>	0x2b
<i>Sync/Async</i>	Synchronous
<i>Reentrancy</i>	Reentrant
<i>Parameters (In)</i>	ConfigPtr: 설정(Static Configuration)에 대한 포인터 Length: 데이터 길이 (바이트)
<i>Parameters (Inout)</i>	StatePtr: 포트/데이터 통신 상태에 대한 포인터 DataPtr: 전송할 데이터에 대한 포인터
<i>Parameters (Out)</i>	None
<i>Return Value</i>	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK
<i>Description</i>	전송할 배열/버퍼를 E2E Profile 6 을 통해 보호한다. Checksum 계산, Counter 에 대한 처리가 포함된다.
<i>Preconditions</i>	E2E_P06ProtectInit()
<i>Configuration Dependency</i>	None

6.3.4.2 E2E_P06ProtectInit

<i>Function Name</i>	E2E_P06ProtectInit
----------------------	--------------------

Syntax:	Std_ReturnType E2E_P06ProtectInit(E2E_P06ProtectStateType* StatePtr)
Service ID	0x2c
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	None
Parameters (Inout)	StatePtr: 포트/데이터 통신 상태에 대한 포인터
Parameters (Out)	None
Return Value	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_OK
Description	Protect 상태를 초기화한다.
Preconditions	NA
Configuration Dependency	None

6.3.4.3 E2E_P06Check

Function Name	E2E_P06Check
Syntax:	Std_ReturnType E2E_P06Check(const E2E_P06ConfigType* ConfigPtr, E2E_P06CheckStateType* StatePtr, const uint8* DataPtr, uint16 Length)
Service ID	0x2d
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	ConfigPtr: 설정(Static Configuration)에 대한 포인터 DataPtr: 수신한 데이터에 대한 포인터 Length: 데이터 길이 (바이트)
Parameters (Inout)	StatePtr: 포트/데이터 통신 상태에 대한 포인터
Parameters (Out)	None
Return Value	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR

	E2E_E_OK
<i>Description</i>	E2E Profile 6 을 사용하여 수신한 데이터를 검사한다. CRC 계산, Counter 에 대한 처리가 포함된다. 한 사이클에 하나의 싱글 데이터만 체크한다. 통신 링크의 축적된 상태를 결정하거나 계산하지 않는다.
<i>Preconditions</i>	E2E_P06CheckInit()
<i>Configuration Dependency</i>	None

6.3.4.4 E2E_P06CheckInit

<i>Function Name</i>	E2E_P06CheckInit
<i>Syntax:</i>	Std_ReturnType E2E_P06CheckInit(E2E_P06CheckStateType* StatePtr)
<i>Service ID</i>	0x2e
<i>Sync/Async</i>	Synchronous
<i>Reentrancy</i>	Reentrant
<i>Parameters (In)</i>	None
<i>Parameters (Inout)</i>	StatePtr: 포트/데이터 통신 상태에 대한 포인터
<i>Parameters (Out)</i>	None
<i>Return Value</i>	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_OK
<i>Description</i>	Check 상태를 초기화한다.
<i>Preconditions</i>	NA
<i>Configuration Dependency</i>	None

6.3.4.5 E2E_P06MapStatusToSM

<i>Function Name</i>	E2E_P06MapStatusToSM
<i>Syntax:</i>	E2E_PCheckStatusType E2E_P06MapStatusToSM (Std_ReturnType CheckReturn, E2E_P06CheckStatusType Status)
<i>Service ID</i>	0x2f
<i>Sync/Async</i>	Synchronous

Reentrancy	Reentrant
Parameters (In)	CheckReturn: E2E_P06Check 함수의 Return value Status: E2E_P06Check 에 의해 결정된 Status
Parameters (Inout)	None
Parameters (Out)	None
Return Value	E2E_PCheckStatusType: Profile 에 독립적인 Status
Description	Profile 6 의 Check status 를 E2E Status Machine 에 사용될 수 있도록 Generic Check Status 에 매핑한다. E2E Profile 6 는 더 정교한 status 를 전달하지만, E2E state machine 과는 관련이 없다.
Preconditions	NA
Configuration Dependency	None

6.3.5 E2E Library – E2E Profile 11 routines

6.3.5.1 E2E_P11Protect

Function Name	E2E_P11Protect
Syntax:	Std_ReturnType E2E_P11Protect(const E2E_P11ConfigType* ConfigPtr, E2E_P11ProtectStateType* StatePtr, uint8* DataPtr, uint16 Length)
Service ID	0x3b
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	ConfigPtr: 설정(Static Configuration)에 대한 포인터 Length: 데이터 길이 (바이트)
Parameters (Inout)	StatePtr: 포트/데이터 통신 상태에 대한 포인터 DataPtr: 전송할 데이터에 대한 포인터
Parameters (Out)	None
Return Value	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK
Description	전송할 배열/버퍼를 E2E Profile 11 을 통해 보호한다. Checksum 계산, Counter 에 대한 처리가 포함된다.
Preconditions	E2E_P11ProtectInit()

Configuration Dependency

None

6.3.5.2 E2E_P11ProtectInit

Function Name	E2E_P11ProtectInit
Syntax:	Std_ReturnType E2E_P11ProtectInit(E2E_P11ProtectStateType* StatePtr)
Service ID	0x3c
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	None
Parameters (Inout)	StatePtr: 포트/데이터 통신 상태에 대한 포인터
Parameters (Out)	None
Return Value	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_OK
Description	Protect 상태를 초기화한다.
Preconditions	NA
Configuration Dependency	None

6.3.5.3 E2E_P11Check

Function Name	E2E_P11Check
Syntax:	Std_ReturnType E2E_P11Check(const E2E_P11ConfigType* ConfigPtr, E2E_P11CheckStateType* StatePtr, const uint8* DataPtr, uint16 Length)
Service ID	0x38
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	ConfigPtr: 설정(Static Configuration)에 대한 포인터 DataPtr: 수신한 데이터에 대한 포인터 Length: 데이터 길이 (바이트)

Parameters (Inout)	StatePtr: 포트/데이터 통신 상태에 대한 포인터
Parameters (Out)	None
Return Value	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK
Description	E2E Profile 11 을 사용하여 수신한 데이터를 검사한다. CRC 계산, Counter 에 대한 처리가 포함된다. 한 사이클에 하나의 싱글 데이터만 체크한다. 통신 링크의 축적된 상태를 결정하거나 계산하지 않는다.
Preconditions	E2E_P11CheckInit()
Configuration Dependency	None

6.3.5.4 E2E_P11CheckInit

Function Name	E2E_P11CheckInit
Syntax:	Std_ReturnType E2E_P11CheckInit(E2E_P11CheckStateType* StatePtr)
Service ID	0x39
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	None
Parameters (Inout)	StatePtr: 포트/데이터 통신 상태에 대한 포인터
Parameters (Out)	None
Return Value	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_OK
Description	Check 상태를 초기화한다.
Preconditions	NA
Configuration Dependency	None

6.3.5.5 E2E_P11MapStatusToSM

Function Name	E2E_P11MapStatusToSM
----------------------	----------------------

Syntax:	E2E_PCheckStatusType E2E_P11MapStatusToSM (Std_ReturnType CheckReturn, E2E_P11CheckStatusType Status)
Service ID	0x3a
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	CheckReturn: E2E_P11Check 함수의 Return value Status: E2E_P11Check 에 의해 결정된 Status
Parameters (Inout)	None
Parameters (Out)	None
Return Value	E2E_PCheckStatusType: Profile 에 독립적인 Status
Description	Profile 11 의 Check status 를 E2E Status Machine 에 사용될 수 있도록 Generic Check Status 에 매핑한다. E2E Profile 11 은 더 정교한 status 를 전 달하지만, E2E state machine 과는 관련이 없다.
Preconditions	NA
Configuration Dependency	None

6.3.6 E2E Library – E2E State Machine routines

6.3.6.1 E2E_SMCheck

Function Name	E2E_SMCheck
Syntax:	Std_ReturnType E2E_SMCheck(E2E_PCheckStateType ProfileStatus, const E2E_SMConfigType* ConfigPtr, E2E_SMCheckStateType* StatePtr)
Service ID	0x30
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	ProfileStatus: Profile 과 독립적인 상태(한 cycle 에서 데이터 수신 상태) ConfigPtr: static 설정에 대한 포인터
Parameters (Inout)	StatePtr: port/data communication 상태에 관한 포인터
Parameters (Out)	None

Return Value	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_INPUTERR_WRONG E2E_E_INTERR E2E_E_OK E2E_E_WRONGSTATE
Description	Communication 채널을 체크한다. E2E_POXCheck 함수의 실행 결과 히스토리에 기반하여, 데이터가 안전과 관련된 어플리케이션에서 사용 가능한지 결정한다.
Preconditions	NA
Configuration Dependency	None

6.3.6.2 E2E_SMCheckInit

Function Name	E2E_SMCheckInit
Syntax:	Std_ReturnType E2E_SMCheckInit(E2E_SMCheckStateType* StatePtr, const E2E_SMConfigType* ConfigPtr)
Service ID	0x31
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	ConfigPtr: state machine 설정에 대한 포인터
Parameters (Inout)	StatePtr: port/data communication 상태에 관한 포인터
Parameters (Out)	None
Return Value	Std_ReturnType: E2E_E_INPUTERR_NULL E2E_E_OK
Description	State machine 을 초기화
Preconditions	NA
Configuration Dependency	None

6.3.7 E2E Protection Wrapper

6.3.7.1 Single channel wrapper routines and init routines

- 1) E2EPW_Write_<p>_<o>

Function Name	E2EPW_Write_<p>_<o>
Syntax:	uint32 E2EPW_Write_<p>_<o>(-- <data>)
Service ID	0x00
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	<data>: 보호하고 전송할 Data element 해당 함수가 컨트롤 필드를 업데이트하는 E2E_PXXProtect 를 호출하기 때문에 이 파라미터는 Inout 이다. 이름과 데이터타입은 Rte_Write 함수와 동일하다.
Parameters (Out)	None
Return Value	uint32: byte 0 (lowest byte) – Rte_Write 함수의 Status · Rte 모듈 User Manual 을 참조 byte 1 – E2E Protection Wrapper 함수의 런타임 체크 Status · E2E_E_INPUTERR_NULL: 최소 하나의 포인터 파라미터가 NULL 포인터이다. · E2E_E_INPUTERR_WRONG: 최소하나의 입력 파라미터에 오류가 있다. 예: Out of range · E2E_E_INTERR: 내부 에러가 발생하였다. 예: program flow monitoring, violated invariant or postcondition 에 의해 발견된 에러 · E2E_E_OK: 성공적으로 수행이 완료되었다. · E2E_E_INVALID: invalid value byte 2 – E2E_PXXProtect 함수의 Return 값 · 6.3.1.1 E2E_P01Protect 또는 6.3.2.1 E2E_P02Protect 참조 byte 3 – 추후 사용을 위한 Placeholder · E2E_E_OK: 일반적인 경우 · E2E_E_INVALID: invalid value
Description	Data Semantic (Non-Queued) 방식으로 Safety 관련 Data element 의 Safe explicit sender-receiver 전송을 개시한다. 이는 데이터를 E2E 라이브러리 함수 E2E_PXXProtect 로 보호하고 관련된 Rte_Write 함수를 호출한다.
Preconditions	None
Configuration Dependency	None

2) E2EPW_Writelnit_<p>_<o>

Function Name	E2EPW_Writelnit_<p>_<o>
----------------------	-------------------------

Syntax:	uint8 E2EPW_WriteInit_<p>_<o>()
Service ID	0x15
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	None
Return Value	uint8: byte 0 – 런타임 체크 상태 · E2E_E_INTERR: 내부 에러가 발생하였다. 예: program flow monitoring, violated invariant or postcondition 에 의해 발견된 에러 · E2E_E_OK: 함수가 성공적으로 수행 완료되었다.
Description	에러 발생 또는 Startup 후 해당 Data structure 에 대한 재초기화
Preconditions	None
Configuration Dependency	None

3) E2EPW_Read_<p>_<o>

Function Name	E2EPW_Read_<p>_<o>
Syntax:	uint32 E2EPW_Read_<p>_<o>(-- <data>)
Service ID	0x00
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	<data>: 수신 데이터 전달을 위한 파라미터 Out 파라미터이다. 파라미터 <data>는 함수 호출이 Return 할 때까지 Valid 해야 한다.
Return Value	uint32: byte 0 (lowest byte) – Rte_Read 함수의 Status · Rte 모듈 User Manual 을 참조 byte 1 – Bit extension 검사를 포함한

	<p>E2E Protection Wrapper 함수의 런타임 체크 Status</p> <ul style="list-style-type: none"> · E2E_E_INPUTERR_NULL: 최소 하나의 포인터 파라미터가 NULL 포인터이다. · E2E_E_INPUTERR_WRONG: 최소하나의 입력 파라미터에 오류가 있다. 예: Out of range · E2E_E_INTERR: 내부 에러가 발생하였다. 예: program flow monitoring, violated invariant or postcondition 에 의해 발견된 에러 · E2EPW_E_DESERIALIZATION: 확장/확대 (Extension/expansion) 에러 발생. Bit extension 이 제대로 수행되었는지에 대한 Status 예: I-PDU 로 12 비트가 16 비트 Unit 으로 확장 시, 최상위 4 비트 값들은 0 이어야 한다. · E2E_E_OK: 성공적으로 수행이 완료되었다. <ul style="list-style-type: none"> · E2E_E_INVALID: invalid value <p>byte 2 – E2E_PXXCheck 함수의 Return 값</p> <ul style="list-style-type: none"> · 6.3.1.2 E2E_P01Check 또는 6.3.2.2 E2E_P02Check 참조 <p>byte 3 – E2E_PXXReceiverStatusType 의 Enum 값으로 Check 함수에 의해 결정되는 E2E Profile XX 내에 데이터 검증 결과를 나타낸다. (다음 에러 값에 대한 자세한 내용은 6.2.2 를 참조)</p> <ul style="list-style-type: none"> · E2EPW_STATUS_NONEWDATA: Error <ul style="list-style-type: none"> · E2EPW_STATUS_WRONGCRC: Error · E2EPW_STATUS_INITIAL: Error · E2EPW_STATUS_OK: OK · E2EPW_STATUS_OKSOMELOST: OK · E2EPW_STATUS_WRONGSEQUENCE: Error · E2E_E_INVALID: Invalid value
Description	<p>Data Semantic (Non-Queued) 방식으로 Safety 관련 Data element 의 Safe explicit sender-receiver 수신을 개시한다. 옵션으로 Rte_IsUpdated 함수를 호출한다. 그리고 Rte_Read 함수를 호출하고, 수신 데이터에 대해 E2E_PXXCheck 함수를 통해 검사한다.</p>
Preconditions	None
Configuration Dependency	None

4) E2EPW_ReadInit_<p>_<o>

Function Name	E2EPW_ReadInit_<p>_<o>
Syntax:	uint8 E2EPW_ReadInit_<p>_<o>()
Service ID	0x16
Sync/Async	Synchronous

Function Name	E2EPW_Write1_<p>_<o>
Syntax:	uint32 E2EPW1_Write_<p>_<o>(-- <data>)
Service ID	0x00
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	<data>: 보호하고 전송할 Data element 해당 함수가 컨트롤 필드를 업데이트하는 E2E_PXXProtect 를 호출하기 때문에 이 파라미터는 Inout 이다. 이름과 데이터타입은 Rte_Write 함수와 동일하다.
Parameters (Out)	None
Return Value	uint32: byte 0 (lowest byte) – E2E_E_OK (Rte_Write 이 실행되지 않기 때문) byte 1 – E2E Protection Wrapper 함수의 런타임 체크 Status · E2E_E_INPUTERR_NULL: 최소 하나의 포인터 파라미터가 NULL 포인터이다.

	<ul style="list-style-type: none"> · E2E_E_INPUTERR_WRONG: 최소하나의 입력 파라미터에 오류가 있다. 예: Out of range · E2E_E_INTERR: 내부 에러가 발생하였다. 예: program flow monitoring, violated invariant or postcondition 에 의해 발견된 에러 · E2E_E_OK: 성공적으로 수행이 완료되었다. · E2E_E_INVALID: invalid value byte 2 – E2E_PXXProtect 함수의 Return 값 · 6.3.1.1 E2E_P01Protect 또는 6.3.2.1 E2E_P02Protect 참조 byte 3 – 추후 사용을 위한 Placeholder · E2E_E_OK: 일반적인 경우 · E2E_E_INVALID: invalid value
Description	E2E Library 함수 E2E_PXXProtect 를 통해 데이터를 보호한다. Rte_Write 함수는 호출하지 않는다.
Preconditions	None
Configuration Dependency	None

2) E2EPW_Write2_<p>_<o>

Function Name	E2EPW_Write2_<p>_<o>
Syntax:	uint32 E2EPW2_Write_<p>_<o>(-- <data>)
Service ID	0x00
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	<data>: 보호하고 전송할 Data element 해당 함수가 컨트롤 필드를 업데이트하는 E2E_PXXProtect 를 호출하기 때문에 이 파라미터는 Inout 이다. 이름과 데이터타입은 Rte_Write 함수와 동일하다.
Parameters (Out)	None
Return Value	uint32: byte 0 (lowest byte) – Rte_Write 함수의 Status <ul style="list-style-type: none"> · Rte 모듈 User Manual 을 참조 byte 1 – E2E Protection Wrapper 함수의 런타임 체크 Status <ul style="list-style-type: none"> · E2E_E_INPUTERR_NULL: 최소 하나의 포인터 파라미터가 NULL 포인터이다. · E2E_E_INPUTERR_WRONG: 최소하나의 입력 파라미터에 오류가 있다. 예: Out of range · E2E_E_INTERR: 내부 에러가 발생하였다. 예: program flow monitoring, violated invariant or postcondition 에 의해 발견된 에러

	<ul style="list-style-type: none"> · E2E_E_REDUNDANCY: Write1 과 Write2 로 계산된 컨트롤 필드 (E2E 헤더: Crc/Counter 등)의 값의 불일치 즉, Write1 과 Write2 간의 Voting 실패 · E2E_E_OK: 성공적으로 수행이 완료되었다. · E2E_E_INVALID: invalid value byte 2 – E2E_PXXProtect 함수의 Return 값 · 6.3.1.1 E2E_P01Protect 또는 6.3.2.1 E2E_P02Protect 참조 byte 3 – 추후 사용을 위한 Placeholder · E2E_E_OK: 일반적인 경우 · E2E_E_INVALID: invalid value
Description	<p>Data Semantic (Non-Queued) 방식으로 Safety 관련 Data element 의 Safe explicit sender-receiver 전송을 개시한다.</p> <p>이는 데이터를 E2E 라이브러리 함수 E2E_PXXProtect 로 보호하고 컨트롤 필드를 Write1 에 의해 계산된 것과 비교한 후 관련된 Rte_Write 함수를 호출한다.</p>
Preconditions	None
Configuration Dependency	None

3) E2EPW_Writelnit1_<p>_<o>

Function Name	E2EPW_Writelnit1_<p>_<o>
Syntax:	uint8 E2EPW_Writelnit1_<p>_<o>()
Service ID	0x17
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	None
Return Value	<p>uint8:</p> <p>byte 0 – 런타임 체크 상태</p> <ul style="list-style-type: none"> · E2E_E_INTERR: 내부 에러가 발생하였다. 예: program flow monitoring, violated invariant or postcondition 에 의해 발견된 에러 · E2E_E_OK: 함수가 성공적으로 수행 완료되었다.
Description	에러 발생 또는 Startup 후 해당 Data structure 에 대한 재초기화
Preconditions	None
Configuration Dependency	None

4) E2EPW_WriteInit2_<p>_<o>

Function Name	E2EPW_WriteInit2_<p>_<o>
Syntax:	uint8 E2EPW_WriteInit2_<p>_<o>()
Service ID	0x18
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	None
Return Value	uint8: byte 0 – 런타임 체크 상태 · E2E_E_INTERR: 내부 에러가 발생하였다. 예: program flow monitoring, violated invariant or postcondition 에 의해 발견된 에러 · E2E_E_OK: 함수가 성공적으로 수행 완료되었다.
Description	에러 발생 또는 Startup 후 해당 Data sturcture 에 대한 재초기화
Preconditions	None
Configuration Dependency	None

5) E2EPW_Read1_<p>_<o>

Function Name	E2EPW_Read1_<p>_<o>
Syntax:	uint32 E2EPW_Read1_<p>_<o>(-- <data>)
Service ID	0x00
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	<data>: 수신 데이터 전달을 위한 파라미터 Out 파라미터이다. 파라미터 <data>는 함수 호출이 Return 할 때까지 Valid 해야 한다.

Return Value	<p>uint32:</p> <p>byte 0 (lowest byte) – Rte_Read 함수의 Status</p> <ul style="list-style-type: none"> · Rte 모듈 User Manual 을 참조 <p>byte 1 – Bit extension 검사를 포함한 E2E Protection Wrapper 함수의 런타임 체크 Status</p> <ul style="list-style-type: none"> · E2E_E_INPUTERR_NULL: 최소 하나의 포인터 파라미터가 NULL 포인터이다. · E2E_E_INPUTERR_WRONG: 최소하나의 입력 파라미터에 오류가 있다. 예: Out of range · E2E_E_INTERR: 내부 에러가 발생하였다. 예: program flow monitoring, violated invariant or postcondition 에 의해 발견된 에러 · E2EPW_E_DESERIALIZATION: 확장/확대 (Extension/expansion) 에러 발생. Bit extension 이 제대로 수행되었는지에 대한 Status 예: I-PDU 로 12 비트가 16 비트 Unit 으로 확장 시, 최상위 4 비트 값들은 0 이어야 한다. · E2E_E_OK: 성공적으로 수행이 완료되었다. · E2E_E_INVALID: invalid value <p>byte 2 – E2E_PXXCheck 함수의 Return 값</p> <ul style="list-style-type: none"> · 6.3.1.2 E2E_P01Check 또는 6.3.2.2 E2E_P02Check 참조 <p>byte 3 – E2E_PXXReceiverStatusType 의 Enum 값으로 Check 함수에 의해 결정되는 E2E Profile XX 내에 데이터 검증 결과를 나타낸다. (다음 에러 값에 대한 자세한 내용은 6.2.2 를 참조)</p> <ul style="list-style-type: none"> · E2EPW_STATUS_NONEWDATA: Error · E2EPW_STATUS_WRONGCRC: Error · E2EPW_STATUS_INITIAL: Error · E2EPW_STATUS_OK: OK · E2EPW_STATUS_OKSOMELOST: OK · E2EPW_STATUS_WRONGSEQUENCE: Error · E2E_E_INVALID: Invalid value
Description	<p>Data Semantic (Non-Queued) 방식으로 Safety 관련 Data element 의 Safe explicit sender-receiver 수신을 개시한다. 옵션으로 Rte_IsUpdated 함수를 호출한다. 그리고 Rte_Read 함수를 호출하고, 수신 데이터에 대해 E2E_PXXCheck 함수를 통해 검사한다.</p>
Preconditions	None
Configuration Dependency	None

6) E2EPW_Read2_<p>_<o>

Function Name	E2EPW_Read2_<p>_<o>
Syntax:	<pre>uint32 E2EPW_Read2_<p>_<o>(-- <data>)</pre>

Service ID	0x00
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	<p><data>: 수신 데이터 전달을 위한 파라미터 Out 파라미터이다. 파라미터 <data>는 함수 호출이 Return 할 때까지 Valid 해야 한다.</p>
Return Value	<p>uint32:</p> <p>byte 0 (lowest byte) – RTE_E_OK (Rte_Read 는 호출되지 않기 때문)</p> <p>byte 1 – Bit extension 검사를 포함한 E2E Protection Wrapper 함수의 런타임 체크 Status</p> <ul style="list-style-type: none"> · E2E_E_INPUTERR_NULL: 최소 하나의 포인터 파라미터가 NULL 포인터이다. · E2E_E_INPUTERR_WRONG: 최소하나의 입력 파라미터에 오류가 있다. 예: Out of range · E2E_E_INTERR: 내부 에러가 발생하였다. 예: program flow monitoring, violated invariant or postcondition 에 의해 발견된 에러 · E2EPW_E_DESERIALIZATION: 확장/확대 (Extension/expansion) 에러 발생. Bit extension 이 제대로 수행되었는지에 대한 Status 예: I-PDU 로 12 비트가 16 비트 Unit 으로 확장 시, 최상위 4 비트 값들은 0 이어야 한다. · E2E_E_OK: 성공적으로 수행이 완료되었다. · E2E_E_INVALID: invalid value <p>byte 2 – E2E_PXXCheck 함수의 Return 값</p> <ul style="list-style-type: none"> · 6.3.1.2 E2E_P01Check 또는 6.3.2.2 E2E_P02Check 참조 <p>byte 3 – E2E_PXXReceiverStatusType 의 Enum 값으로 Check 함수에 의해 결정되는 E2E Profile XX 내에 데이터 검증 결과를 나타낸다. (다음 에러 값에 대한 자세한 내용은 6.2.2 를 참조)</p> <ul style="list-style-type: none"> · E2EPW_STATUS_NONEWDATA: Error · E2EPW_STATUS_WRONGCRC: Error · E2EPW_STATUS_INITIAL: Error · E2EPW_STATUS_OK: OK · E2EPW_STATUS_OKSOMELOST: OK · E2EPW_STATUS_WRONGSEQUENCE: Error · E2E_E_INVALID: Invalid value
Description	해당 Read1 에서 수신한 데이터를 E2E_PXXCheck 를 이용하여 다시 검사한다.
Preconditions	None

Configuration Dependency

None

7) E2EPW_ReadInit1_<p>_<o>

Function Name	E2EPW_ReadInit1_<p>_<o>
Syntax:	uint8 E2EPW_ReadInit1_<p>_<o>()
Service ID	0x19
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	None
Return Value	uint8: byte 0 – 런타임 체크 상태 · E2E_E_INTERR: 내부 에러가 발생하였다. 예: program flow monitoring, violated invariant or postcondition 에 의해 발견된 에러 · E2E_E_OK: 함수가 성공적으로 수행 완료되었다.
Description	에러 발생 또는 Startup 후 해당 Data sturcture 에 대한 재초기화
Preconditions	None
Configuration Dependency	None

8) E2EPW_ReadInit2_<p>_<o>

Function Name	E2EPW_ReadInit2_<p>_<o>
Syntax:	uint8 E2EPW_ReadInit2_<p>_<o>()
Service ID	0x1a
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	None

Return Value	uint8: byte 0 – 런타임 체크 상태 · E2E_E_INTERR: 내부 에러가 발생하였다. 예: program flow monitoring, violated invariant or postcondition 에 의해 발견된 에러 · E2E_E_OK: 함수가 성공적으로 수행 완료되었다.
Description	에러 발생 또는 Startup 후 해당 Data structure 에 대한 재초기화
Preconditions	None
Configuration Dependency	None

6.3.8 COM E2E Callouts

6.3.8.1 IPDU_E2EProtect_<IPDU ID> ()

Function Name	IPDU_E2EProtect_<IPDU ID>
Syntax:	boolean IPDU_E2EProtect_<IPDU ID>(PduldType Pduld, PdulInfoType* PdulInfoPtr)
Service ID	N/A
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	Pduld: I-PDU ID
Parameters (Inout)	PdulInfoPtr: I-PDU 데이터의 포인터
Parameters (Out)	None
Return Value	boolean: true – E2E_PXXProtect 함수를 통한 데이터 보호 완료 false – E2E_PXXProtect 실행 중 오류 발생
Description	ComIPduCallout 을 통해 E2E 를 사용하여 전송될 배열/버퍼에 대해 보호를 수행한다.
Preconditions	None
Configuration Dependency	None

6.3.8.2 IPDU_E2ECheck_<IPDU ID> ()

Function Name	IPDU_E2ECheck_<IPDU ID>
----------------------	-------------------------

Syntax:	boolean IPDU_E2ECheck_<IPDU ID>(PduldType Pduld, PduInfoType* PduInfoPtr)
Service ID	N/A
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	Pduld: I-PDU ID PduInfoPtr: I-PDU 데이터의 포인터
Parameters (Inout)	None
Parameters (Out)	None
Return Value	boolean
Description	ComIPduCallout 을 통해 E2E 를 사용하여 수신한 데이터에 대해 검사를 수행한다.
Preconditions	None
Configuration Dependency	None

7. Generator

7.1 Generator Option

7.1.1 E2E

해당 사항 없음

7.1.2 Usage of E2E Library (Rte)

Option	Description
Bend	CPU 의 Byte Order 가 Big Endian (High Byte First)일 때, 이 옵션을 추가한다.

7.2 Generator Error Message

7.2.1 E2E

7.2.1.1 Error Messages

해당 사항 없음

7.2.1.2 Warning Messages

해당 사항 없음

7.2.1.3 Information Messages

해당 사항 없음

7.2.2 Usage of E2E Library (Rte)

7.2.2.1 Error Messages

- 1) ERR002000: The attribute category of EndToEndDescription can have the following values:
NONE, PROFILE_01, PROFILE_02.
 - 원인: EndToEndProfile (EndToEndDescription) 컨테이너의 Category 가 NONE, PROFILE_01, PROFILE_02 이외의 값으로 설정되어 있다.
 - 조치사항: 카테고리를 NONE, PROFILE_01, PROFILE_02 중에 하나로 선택한다.
- 2) ERR002001: In PROFILE_01, the applicable range of values for counterOffset is [0 .. 65535].
For the value of this attribute the constraint value mod 4 = 0 applies.
 - 원인: EndToEndProfile (EndToEndDescription)의 Category 가 PROFILE_01 일 때, CounterOffset 의 값이 [0..65535]를 벗어나거나 4 의 배수가 아니다.
 - 조치사항: CounterOffset 의 값을 [0..65535] 범위내의 4 의 배수로 입력한다.
- 3) ERR002002: In PROFILE_01, the applicable range of values for crcOffset is [0 .. 65535].

For the value of this attribute the constraint value mod 8 = 0 applies.

- 원인: EndToEndProfile (EndToEndDescription)의 Category 가 PROFILE_01 일 때, CrcOffset 의 값이 [0..65535]를 벗어나거나 8 의 배수가 아니다.

- 조치사항: CrcOffset 의 값을 [0..65535] 범위내의 8 의 배수로 입력한다.

4) ERR002003: In PROFILE_01, the applicable range of values for dataIdMode is [0 .. 2].

- 원인: EndToEndProfile (EndToEndDescription)의 Category 가 PROFILE_01 일 때, DataIdMode 의 값이 0, 1, 2 이외의 값이다.

- 조치사항: EndToEndProfile (EndToEndDescription)의 DataIdMode 의 값을 0~2 중 입력한다.

5) ERR002004: In PROFILE_01, there shall be only one element in the set of dataIds and in PROFILE_02, there shall be exactly ordered 16 elements in the set of dataIds.

- 원인: EndToEndProfile (EndToEndDescription)의 Category 가 PROFILE_01 이면서 DataID 의 개수가 1 개가 아니거나 Category 가 PROFILE_02 이면서 DataID 의 개수가 16 개가 아니다.

- 조치사항: EndToEndProfile (EndToEndDescription)의 DataId 의 개수를 Category 가 PROFILE_01 이면 1 개, Category 가 PROFILE_02 이면 16 개를 입력한다.

6) ERR002005: In PROFILE_01, the applicable range of values for the element of dataIds is [0 .. 65535] and in PROFILE_02, the applicable range of values for each element of dataIds is [0 .. 255].

- 원인: EndToEndProfile (EndToEndDescription)의 Category 가 PROFILE_01 이면서 DataId 의 값이 [0..65535]를 벗어나거나 Category 가 PROFILE_02 이면서 DataId 의 값이 [0..255]를 벗어난다.

- 조치사항: EndToEndProfile (EndToEndDescription)의 DataId 의 값을 Category 가 PROFILE_01 일 때는 [0..65535] 범위 내의 값으로, PROFILE_02 일 때는 [0..255] 범위 내의 값으로 입력한다.

7) ERR002006: In PROFILE_01, the applicable range of values for dataLength is [0 .. 240] and in PROFILE_02, the applicable range of values for dataLength is [0 .. 65535]. For the value of this attribute the constraint value mod 8 = 0 applies.

- 원인: EndToEndProfile (EndToEndDescription)의 Category 가 PROFILE_01 이면서 DataLength 의 값이 [0..240]의 범위를 벗어났거나 Category 가 PROFILE_02 이면서 DataLength 의 값이 [0..65535]를 벗어났다. 또는 Category 에 관계 없이 DataLength 의 값이 8 의 배수가 아니다.

- 조치사항: EndToEndProfile (EndToEndDescription)의 DataLength 의 값을 Category 가 PROFILE_01 이면 [0..240] 범위 내의 값으로, Category 가 PROFILE_02 이면 [0..65535] 범위 내의 값으로 입력한다. 그리고 Category 에 관계없이 DataLength 의 값은 8 의 배수로만 입력한다.

8) ERR002007: In PROFILE_01, the applicable range of values for maxDeltaCounterInit is [0 .. 14] and in PROFILE_02, the applicable range of values for maxDeltaCounterInit is [0 .. 15].

- 원인: EndToEndProfile (EndToEndDescription)의 Category 가 PROFILE_01 이면서

MaxDeltaCounterInit 의 값이 [0..14]를 벗어나거나 Category 가 PROFILE_02 이면서 MaxDeltaCounterInit 의 값이 [0..15]를 벗어난다.

- 조치사항: EndToEndProfile (EndToEndDescription)의 MaxDeltaCounterInit 의 값을 Category 가 PROFILE_01 일 때는 [0..14] 범위 내의 값으로, PROFILE_02 일 때는 [0..15] 범위 내의 값으로 입력한다.

9) ERR002008: The ISignalGroupRef of EndToEndProtectionISignalIPdu is not configured or invalid.

- 원인: EndToEndProtectionISignalIPdu 의 ISignalGroupRef 에 값이 없거나 참조하는 ISignalGroup 이 존재하지 않는다.

- 조치사항: EndToEndProtectionISignalIPdu 의 ISignalGroupRef 를 실제 존재하는 ISignalGroup 을 참조하는 값으로 입력한다.

10) ERR002009: The ISignalIPduRef of EndToEndProtectionISignalIPdu is not configured or invalid.

- 원인: EndToEndProtectionISignalIPdu 의 ISignalIPduRef 에 값이 없거나 참조하는 ISignalGroup 이 존재하지 않는다.

- 조치사항: EndToEndProtectionISignalIPdu 의 ISignalIPduRef 를 실제 존재하는 ISignalGroup 을 참조하는 값으로 입력한다.

11) ERR002011: If the E2E Library is invoked at the level of Data Elements, then a Data Element shall either map to a local intra-ECU communication (without COM involvement) or shall map to a COM I-PDU, but shall not map to both at the same time.

- 원인: E2E Protection Wrapper 를 사용하는 Port 에 대해 Intra-ECU 연결 (Assembly Connector 를 통한)과 Inter-ECU (SenderReceiverToSignalGroupMapping)이 모두 존재한다.

- 조치사항: 해당 Port 의 연결을 Intra-ECU 또는 Inter-ECU 중 한 종류가 되도록 구성한다.

12) ERR002012: If the E2E Library is invoked at the level of Data Elements (e.g. from SW-Cs or from E2E Protection Wrapper), then the communication shall be an explicit sender-receiver communication, in 1:1 and 1:N multiplicities.

- 원인: E2E Protection Wrapper 를 사용하는 Port 에 대해 N:1 연결이 존재한다.

- 조치사항: 해당 Port 의 연결을 1:1 또는 1:N 으로만 구성한다.

13) ERR002013: A given I-PDU <ComIPdu Path>, shall not be at the same protected by means of COM E2E callouts (through association with ISignalIPdu) and by means of E2E Protection Wrapper (through association with E2E Protection Wrapper).

- 원인: 해당 I-PDU 에 대해 E2E Protection Wrapper 와 Com E2E Callouts 가 동시에 설정되었다.

- 조치사항: I-PDU 와 관련한 SW-C 의 Port 에 대한 E2E Protection Wrapper 설정을 제거하거나 I-PDU 의 ComIPduCallout 을 통해 연결된 Com E2E Callouts 설정을 제거한다.

14)ERR002014: The ContextComponentRef in VariableDataPrototypeInSystemInstanceRef is not configured or invalid.

- 원인: E2E Protection Wrapper 를 위해 설정한 EndToEndProtectionVariablePrototype 의 Sender 또는 Receiver 들의 VariableDataPrototypeInSystemInstanceRef 컨테이너의 ContextComponentRef 가 설정되지 않았거나 설정 값이 실제 존재하지 않는다.
- 조치사항: ContextComponentRef 를 실제 존재하는 대상을 참조하도록 설정한다.

15)ERR002015: The ContextPortRef in VariableDataPrototypeInSystemInstanceRef is not configured or invalid.

- 원인: E2E Protection Wrapper 를 위해 설정한 EndToEndProtectionVariablePrototype 의 Sender 또는 Receiver 들의 VariableDataPrototypeInSystemInstanceRef 컨테이너의 ContextPortRef 가 설정되지 않았거나 설정 값이 실제 존재하지 않는다.
- 조치사항: ContextPortRef 를 실제 존재하는 대상을 참조하도록 설정한다.

16)ERR002016: The TargetDataPrototypeRef in VariableDataPrototypeInSystemInstanceRef is not configured or invalid.

- 원인: E2E Protection Wrapper 를 위해 설정한 EndToEndProtectionVariablePrototype 의 Sender 또는 Receiver 들의 VariableDataPrototypeInSystemInstanceRef 컨테이너의 TargetDataPrototypeRef 가 설정되지 않았거나 설정 값이 실제 존재하지 않는다.
- 조치사항: TargetDataPrototypeRef 를 실제 존재하는 대상을 참조하도록 설정한다.

7.2.2.2 Warning Messages

없음

7.2.2.3 Information Messages

없음

8. Appendix

8.1 Application Note (Usage of E2E Library)

E2E Protection Wrapper 와 Com E2E Callouts 를 사용하기 위한 방법을 안내한다.

8.1.1 E2E Protection Wrapper

8.1.1.1 UsesEndToEndProtection

E2E Protection Wrapper 는 Sender-Receiver 통신에 대응하며, 특히, 1:1 또는 1:N Explicit Non-Queued Sender-Receiver 통신에서 사용 가능하다.

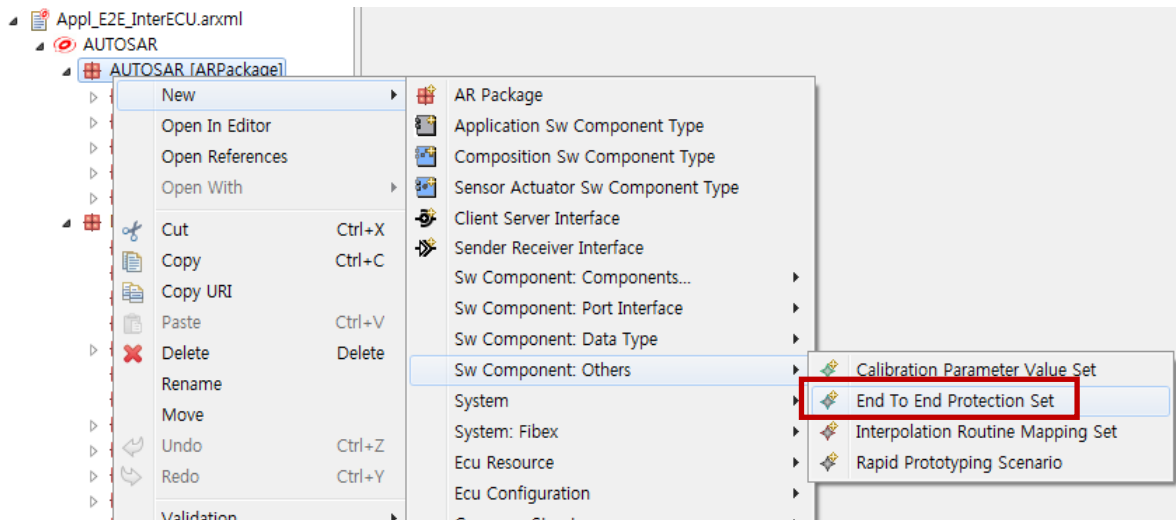
- 1:1 또는 1:N Explicit Non-Queued Sender-Receiver 통신을 구성하면, Sender 측 컴포넌트의 PPortPrototype 에는 NonqueuedSenderComSpec 가 Receiver 측 컴포넌트의 RPortPrototype 에는 NonqueuedReceiverComSpec 가 추가된다. NonqueuedSenderComSpec 및 NonqueuedReceiverComSpecs 에는 각각 UsesEndToEndProtection 항목이 존재하는데, 이를 “true” 로 변경한다. (아래 그림은 Sender 측 컴포넌트의 PPortPrototype 내에 NonqueuedSenderComSpec 이다.)

The screenshot shows the 'Communication Spec' configuration window. The 'Data Elements' dropdown is set to 'DataElement1' with a '(Total Count: 1)' label. Under 'Provided Com Specs', the 'Enable Provided Com Specs' checkbox is checked, and a 'Nonqueued Sender Com Spec' is listed with an 'Edit' button. The 'Init Value' section has 'None' selected. The 'Transmission Acknowledge' section has 'Enable Transmission Acknowledge' checked and a 'Timeout' of 100 msec. The 'Uses End To End Protection' section is highlighted with a red box, showing 'Enable Uses End To End Protection' checked.

8.1.1.2 EndToEndProtectionSet

EndToEndProtectionSet 컨테이너에서 End-To-End Protection 관련 설정을 할 수 있다.

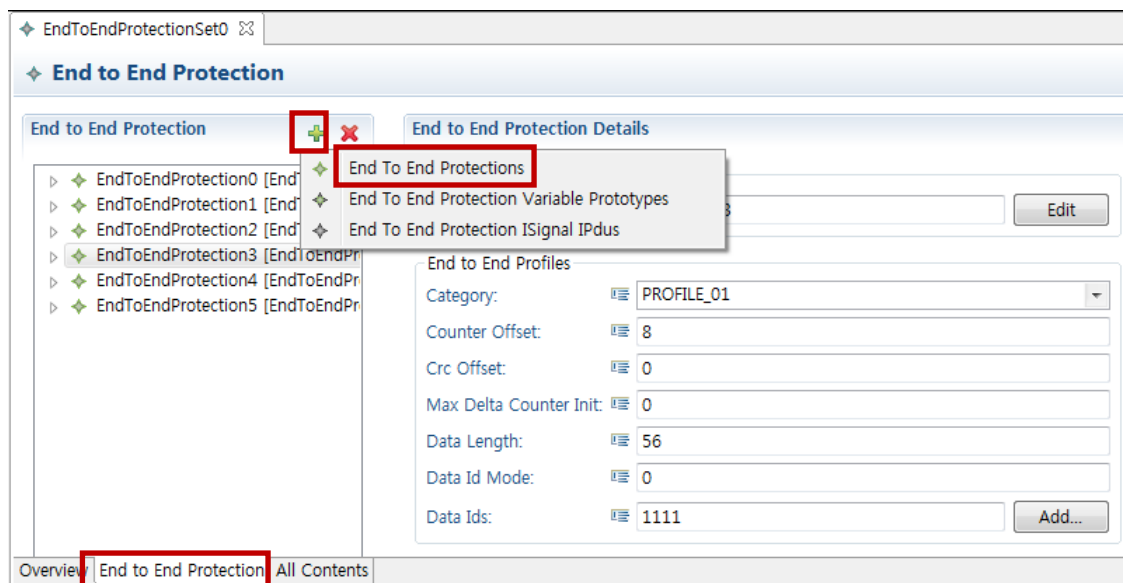
1. Arxml 파일의 ArPackage 에 EndToEndProtectionSet 을 추가한다.



8.1.1.3 EndToEndProtection

EndToEndProtectionSet 에는 다수의 EndToEndProtection 을 추가할 수 있다.

1. EndToEndProtectionSet 화면을 열고, End to End Protection 탭으로 이동한 후, 왼쪽의 (+) 버튼을 눌러 EndToEndProtection 을 선택하여 추가한다.



8.1.1.4 EndToEndProfile

EndToEndProtection 에는 Configuration 정보를 설정하는 EndToEndProfile 을 설정할 수 있다.

각 항목들은 E2E_PXXConfigType 에 매칭된다. (6.1.1.1, 6.1.2.1 참조)

1. 추가된 EndToEndProtection 의 오른쪽 End to End Profiles 의 각 항목을 입력한다.

8.1.1.5 EndToEndProtectionVariablePrototype

EndToEndProtection 내의 EndToEndProfile 과 Sender-Receiver 통신을 연결하는 설정이다.

1:1 또는 1:N 만 가능하므로 하나의 Sender 와 여러 Receiver 들을 설정할 수 있다.

1. EndToEndProtection 을 선택 후, (+) 버튼을 눌러 End To End Protection Variable Prototype 를 선택하여 추가한다.

2. EndToEndProtectionVariablePrototype 이 추가되면, 오른쪽에 Sender 와 Receivers 를 설정할 수 있는 화면이 나타난다. Sender 는 각 항목을 직접 입력할 수 있으며, Receivers 는 각각의 Receiver 들을 New 버튼을 통해 추가한 후, 테이블 내에서 각 항목을 편집한다.

8.1.2 Com E2E Callouts

Com E2E Callouts 는 Com 의 I-PDU Callout 을 통해 E2E Protection 을 수행하는 기능이다.

8.1.2.1 EndToEndProtectionISignalIPdus

EndToEndProtection 내의 EndToEndProfile 과 ISignal/IPdu 를 연결하는 설정이다.

1. 8.1.1.2~8.1.1.4 에 따라 설정한 뒤에, EndToEndProtection 를 선택 후, (+) 버튼을 눌러 End To End Protection ISignal IPdus 를 선택하여 추가한다.

2. EndToEndProtectionISignalIPdu 가 추가되면, 오른쪽에 ISignalGroupRef 와 ISignalIPduRef 를 선택하는 화면이 나타난다. 이 때 E2E 를 사용하려는 ISignalGroup 과 ISignalGroup 이 포함된 ISignalIPdu 를 선택한다.

End to End Protection ISignal IPdu Details

End To End Protection ISignal IPdu

ISignal Group: ComSignalGroup_Tx [/DBCImport/ISIGNALGROUPS/ComSignalGroup_Tx] ... +

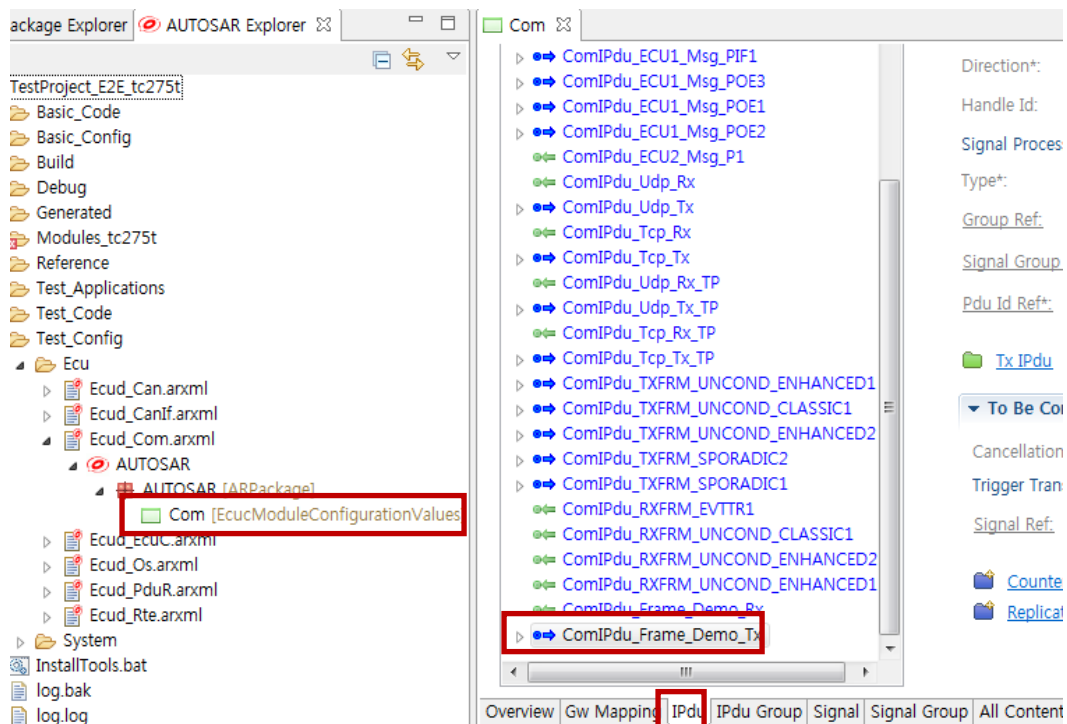
ISignal IPdu: Frame_Demo_Tx [/DBCImport/PDUS/Frame_Demo_Tx] ... +

Data Offset:

8.1.2.2 ComIPduCallout

Com E2E Callouts 는 I-PDU Callout 을 이용하기 때문에 Com E2E Callouts 에서 생성한 API 를 Com 에 등록하는 과정이 필요하다.

1. 프로젝트에서 Com 의 Ecu Configurion 파일(예: Ecud_Com.arxml)을 열고, IPdu 탭으로 이동하여, 8.1.2.1 의 2 에서 설정한 ISignalIPdu 에 해당하는 ComIPdu 을 선택한다.



2. ComIPduDirection 과 ComIPduHandleId 를 확인하여, ComIPduDirection 이 SEND 일 경우에는 IPDU_E2EProtect_<ComIPduHandleId>, ComIPduDirection 이 RECEIVE 일 경우에는 IPDU_E2ECheck_<ComIPduHandleId>를 ComIPduCallout 에 입력한다.

Container Details - ComIPdu

Short Name*:	ComIPdu_Frame_Demo_Tx
Callout:	IPDU_E2EProtect_22
Direction*:	SEND
Handle Id:	22
Signal Processing*:	DEFERRED
Type*:	NORMAL
Group Ref:	ComIPduGroupTx_ECU1_Project [/AUTOSAR/Com/ComConfig/ComIPduGroupTx_ECU1_Project] (/TestProject_E2E_tc275t
Signal Group Ref:	ComISignalGroup_ComSignalGroup_Tx [/AUTOSAR/Com/ComConfig/ComISignalGroup_ComSignalGroup_Tx] (/TestProje
Pdu Id Ref*:	Pdu_Frame_Demo_Tx [/AUTOSAR/EcuC/EcucPduCollection/Pdu_Frame_Demo_Tx] (/TestProject_E2E_tc275t/Test_Config/E

8.2 설계시 유의사항

8.2.1 E2E

➤ Profile 1 과 Profile 2 의 상태 초기화

- Profile 1 과 Profile 2 는 Protect 및 Check 에 제공할 State 변수를 사용자가 직접 초기화 해야 한다. 또한 E2E_P01ReceiverState/E2E_P02ReceiverState 의 NewDataAvailable 값은 사용자가 직접 처리해야 한다. (예: Rte_IsUpdate 를 사용하거나 Rte_Read API 호출 시 마다, Return Value 에 따라 데이터의 사용 가능성 여부를 체크하여 State 변수에 업데이트)

➤ Inter-ECU 통신에서 E2E 사용 시 데이터의 길이

※ 이는 “Specification of SW-C End-To-End Communication Protection Library” [1]의 11.3 에 기술된 사항으로 사용자는 해당 내용을 반드시 확인하여야 한다.

- 주어진 CRC 에 대해 메시지의 길이와 해밍 거리(Hamming Distance)가 연관되어 있다. 요구되는 진단 커버리지를 충족하기 위해 CRC 로 보호되는 Data Element 의 최대 길이를 적절하게 선택해야 한다. Inter-ECU 통신을 보호하기 위한 E2E 프로파일 별 길이는 다음 표와 같다.

E2E Profile	Inter-ECU 통신에 적용 가능한 최대 길이 (컨트롤 필드 포함)
E2E Profile 1	30
E2E Profile 2	42

- FlexRay 를 통한 Inter-ECU 통신의 경우 E2E Profile 1 에 의해 보호되는 데이터의 길이는 30 바이트를 넘을 수 없다. [E2EUSE051]
- CAN 또는 LIN 을 통한 Inter-ECU 통신의 경우 E2E Profile 1 에 의해 보호되는 데이터의 길이는 8 바이트를 넘을 수 없다. [E2EUSE061]
- E2E Library 를 사용할 때 E2E Library 를 사용하는 특정 시스템의 Functional/Technical safety

concept 의 설계자는 적절한 에러 감지 능력을 보장하기 위해 시스템 내에 보호되는 데이터의 최대 허용 길이를 평가해야 한다. [E2EUSE236]

- 특정 시스템의 Functional/Technical safety concept 을 설계할 때, E2E Library 의 사용자(애플리케이션)가 Sender 와 Receiver 간 Data Element 시퀀스 내에 에러가 있는 Data Element 를 탐지 하지 못한 채로 전송하는 것이 이 시스템의 Safety goal 을 위반하지 않도록 보장해야 한다. [E2EUSE170]
- SW-C 설계할 때, E2E Profile 1, 2 의 User 는 Multibyte 데이터에 대해 Little Endian 순서로 Data Element 에 배치해야 한다. [E2EUSE055]

➤ E2E State Machine 사용 시 주의 사항

- ProfileStatusWindow 는 반드시 배열로 선언 후, SMCheckStateType 으로 만든 구조체의 ProfileStatusWindow 변수에 배열 포인터로 들어가야 한다. 또한, 앞서 선언한 배열의 크기는 E2E_SMConfigType 으로 만든 구조체의 WindowSize 변수와 동일해야 한다.

8.2.2 Usage of E2E Library (Rte)

➤ Inter-ECU 통신에서 I-PDU 레이아웃

※ 이는 “Specification of SW-C End-To-End Communication Protection Library” [1]의 11.7 에 기술된 사항으로 사용자는 해당 내용을 반드시 확인하여야 한다.

- 길이가 8 비트 보다 작은 Signal 을 I-PDU 의 한 바이트 내에 배치해야 한다. 즉, 2 바이트에 걸쳐도록 배치할 수 없다. [E2EUSE062]
- 길이가 8 비트 이상인 Signal 은 I-PDU 의 바이트 경계에서 시작하거나 끝나야 한다. [E2EUSE063]
- Unused Bit 는 1 로 채운다. 따라서 E2E 를 사용하는 Com I-PDU 의 경우, ComTxIPduUnusedAreasDefault 를 반드시 0xFF 로 설정한다. [E2EUSE173]

➤ Signal/Signal Group 이외의 I-PDU 내 포함되는 정보

- I-PDU 내 Signal/Signal Group 이외에 Update bit, I-PDU counter 등이 포함될 경우, 이는 E2E 를 사용하려는 데이터 사이에 위치할 수 없다. E2E 는 데이터를 배열화하여 시작 위치부터 데이터의 길이까지의 모든 데이터에 대해 CRC 계산을 수행한다. 따라서 Update bit, I-PDU Counter 등이 CRC 계산에 포함될 경우 오류가 발생할 가능성이 있다. I-PDU 설계 시 해당 데이터 는 E2E 보호 데이터 영역에 제외되도록 하여야 한다.

8.2.3 E2EXf 사용 시 유의사항

➤ Data Length 설정

- DB 내 지정 된 E2E 메시지에서 가장 마지막 Signal 이 속하는 byte 까지의 길이를 bit size 로

입력한다(단, 8 로 나누어 떨어지도록 입력)

예)

	7	6	5	4	3	2	1	0
0	DAU_Crc2Val							
1	DAU_AinCrc2Val							
2								
3								
4								
5								
6								

만약 E2E 메시지의 Signal 배치가 위와 같다면, 가장 마지막 Signal 이 속하는 byte 까지의 길이를 bit size 로 입력하면 $8 \times 7 = 56$ 이 된다.

➤ Message 종류 별 E2E 사용 가이드 라인(HMC 의 E2E 사양에 따라 설계할 경우)

- 이벤트 속성이 없는 주기 메시지

1. 메시지의 주기와 동일하게 ASW 에서 Write API 를 호출하도록 설계해야 한다.

이유 : 주기 메시지는 Com 모듈 내에서 자체적으로 전송한다. 현재 HMC 표준플랫폼 상에서는 Write API 가 수행되어야만 E2E Header 가 업데이트 되고, 따라서 Com 모듈 내에서 자체적으로 전송하는 값은 E2E 검사가 올바르게 진행될 수 없다.(초기에 CRC 값이 업데이트 되지 않고 전송되어 CRC 오류 발생, Counter 가 업데이트 되지 않고 전송되어 카운터(중복) 오류 발생

- Change on write 이벤트 메시지

1. 보내려는 Data 값에 변화가 없어도, ASW 에서 Write API 를 호출하면 Data 가 변경되지 않을 경우에도 Data 가 전송된다. 이를 유의하여 설계한다.

이유 : ASW Write API 를 호출하면 E2E Header 가 업데이트 되면서 Counter 값이 변한다. 따라서 데이터가 전송된다.

- Repetition 이벤트 메시지

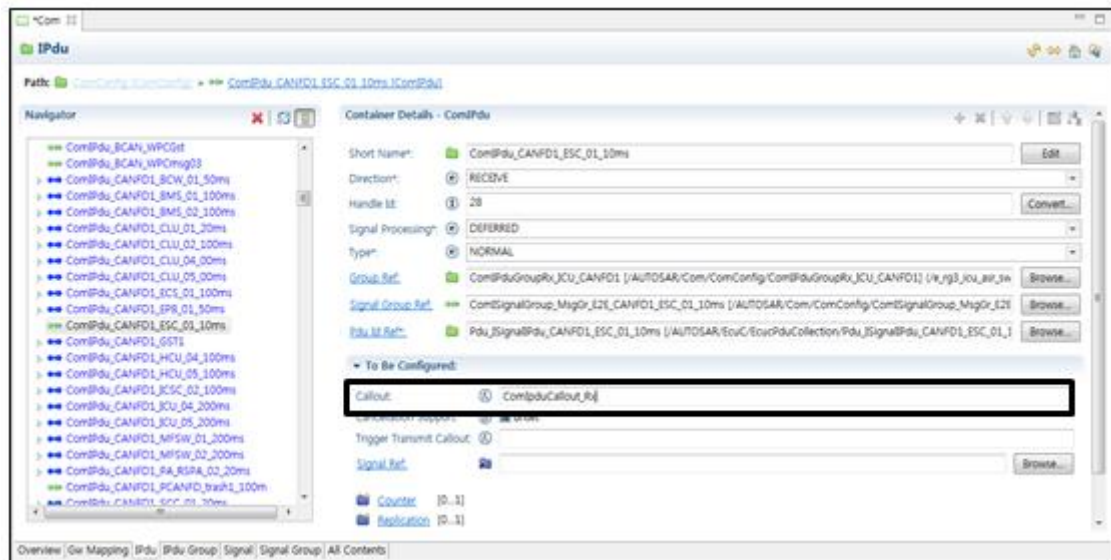
1. 카운터 검사 시 중복 에러가 발생한다. 이를 유의하여 설계한다.

이유 : 같은 데이터(같은 카운터 값)가 중복하여 전송되므로, 카운터 중복 에러가 발생한다.

- 이벤트 속성이 있는 주기 메시지

1. Event 속성이 있으므로 Alive Counter 에러 처리를 하지 않는다.
 2. 이벤트 속성이 없는 주기 메시지와는 다르게 매 주기마다 Write 처리를 하지 않는다(Event 메시지를 보낼 필요가 있는 경우에만 Write 를 수행한다).
 3. 최초 CRC 값 업데이트를 위해, 초기 한번의 Write 는 무조건 수행한다.
- E2E 를 포함하는 메시지에 대한 분할 라우팅 처리(Ethernet to CAN)
1. Com 의 Signal Gateway 혹은 Application 에서 수신 받아 Signal 단위로 gateway 를 처리한다.
 2. CRC 와 Counter 값은 Application 에 전달이 되지 않기 때문에 아래의 방법으로 처리한다.

2-1) Com 모듈의 IPdu 설정에서 Callout 등록



2-2) Callout 함수를 통한 값 수신

boolean ComIPduCallout_Rx (PduIdType PduId, const PduInfoType* PduInfoPtr);

- A. Return 값 true 로 처리 (E2E 처리 및 수신을 진행할 것 이므로)
- B. PduInfoPtr 를 통해 수신 된 CRC 와 Counter 값 저장

예) profile 5 의 경우 아래와 같이 접근해서 Byte 단위로 접근할 수 있다.

(값을 변수에 저장하고자 하는 경우 Endian 변환 처리를 해주어야 한다)

CRC : PduInfoPtr->SduDataPtr[0] ~ PduInfoPtr->SduDataPtr[1]

Counter : PduInfoPtr->SduDataPtr[2]

PduInfoPtr는 아래 PduInfoType의 포인터값

```
typedef struct
{
    P2VAR(uint8, AUTOMATIC, AUTOSAR_COMSTACKDATA) SduDataPtr;
    PduLengthType SduLength;
} PduInfoType;
```

2-3) Gateway 처리 (아래 두 가지 중 택일)

- A. Callout 함수에서 Endian 처리를 통해 값을 변수에 저장한 후 Rte_Write 를 통해서 송신
- B. Tx Pdu 에 Callout 함수를 등록하여, Rx Pdu 의 Callout 함수에서 저장한 값을

Byte 단위로 Copy 한 후 송신

3. CAN-FD to CAN 의 경우는 CDD Router 모듈에서 처리한다.